People Issues Relating to Software Engineering Education and Training in India

by

Kirti Garg, Vasudeva Varma

in


Report No: IIIT/TR/2008/29

Centre for Software Engineering Research Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2008
People Issues Relating to Software Engineering Education and Training in India

Kirti Garg
International Institute of Information Technology, Hyderabad, India
kirti@research.iiit.ac.in

Vasudeva Varma
International Institute of Information Technology, Hyderabad, India
vv@iiit.ac.in

ABSTRACT
Software Engineering (SE) and Information Technology (IT) jobs are the most sought after career options for Indian youth in the recent times. Indian Software industry is expected to grow at a very healthy rate and each major software company has ambitious plans and growth targets for future. However, lack of proper Software Engineering (SE) education may have severe consequences and may negatively affect these growth targets.

In this paper we discuss challenges and issues related to software engineering education and training in Indian academia and industry. These are based on our interaction with industry and through our experience as Software engineering educators. These challenges arise from deep rooted issues in Software Engineering educational goals, pedagogy and instruction as well as the infrastructure. We will discuss their long term effects on various aspects of software development. We put forth our suggestions that may handle these challenges to an extent.

We also discuss the essential and minimal set of software engineering knowledge, skills and dispositions that the industry requires from young engineers willing to join the industry.

This paper provides course designing guidelines for the academia and training centers of the industry by focusing on important SE education issues, their causes and possible solutions. This, in turn would help to make SE Education more effective and inline with requirements of the Indian Software industry.

Categories and Subject Descriptors
K.3.2 [Computer and Information Science Education]

General Terms
Human Factors

Keywords
Software Engineering Education, problem solving, People Issues, India

1. INTRODUCTION
Indian IT and ITeS (IT enabled Services) industry is enjoying the leadership position in the world by accounting for 65% of the global industry in IT and off-shoring services. According to McKinsey 2005 report on “Extending India's Leadership of the Global IT and BPO Industries”, one of the major reasons for this success is the availability of abundant talent and operational excellence [3]. This report also states that IT and BPO industry account for 700,000 direct and 2.5 million indirect jobs.

Indian IT industry exhibited growth rates of about 25% since 2000. However, sustaining this growth rate is a challenge being faced now and the hurdle is the potential shortage of skilled workers in the next decade. Currently only 25% of the technical graduates are employable and the hit rate (ratio between the number of actual recruitments to number of job applications received by the Personnel department) is around 5-10% in major software services companies.

Insufficient numbers as well as lack of appropriate training is predicted to curtail the growth of IT industry in India in future. This is one major people related issue in IT sector which is directly resulting from poor software engineering education and training programs and infrastructure in the country.

We would like to distinguish between problems of Computer Science education and Software Engineering education, as these two disciplines are different in nature and hence their issues also differ to a certain extent. SE education problems lead to more people issues as SE is largely a techno-managerial subject and people form an important and integral part of the discipline.

To understand the education and training issues and the gap between the demand and supply, we have conducted a survey involving a questionnaire and a few interviews with senior executives of many major software services companies in India (including TCS, Infosys, Satyam, and Accenture). Most of the findings presented in this paper are based on this survey and our observations of the academic scenario for a couple of years. Survey also helped us to gauge Indian Software industry’s perspective of SEE challenges and problems. Respondents were asked about two aspects; a) the SE knowledge and skills expected from an entry level Software Engineer. B) Problems faced by them due to ineffective SE education.

The problems of SE education in India are mostly inter-linked. Some of these issues have been discussed very briefly by Mahanti et al in [2].

This paper is organized around current problematic features (called as issues), their consequences/impact and guidelines of
2.2 Industrial Requirements for Qualified Personnel

Indian Software industry has always felt that there are not enough well qualified software engineers that can fulfill the industry needs. It is important to identify the true nature of discipline of software engineering and accordingly decide what should be treated as a ‘qualified training program’. This would derive the educational goals or curriculum of any SE education endeavor. We discussed the notion of ‘qualified training program’ with our industry counterparts and they have all agreed that any SE courses should not be restricted to the knowledge and understanding component of learning, rather it should include higher order cognitive skills such as application, analysis, synthesis and evaluation as well. The survey asked the experts from industry to rate the importance of various other skills (problem solving, communication, self-learning, research aptitude etc.) on a scale of 0 to 5, where, 0 is not required at all and 5 is most required. All the respondents gave a minimum rating of 4 to the problem solving skills. This was followed by self-learning which was given a constant rating of 4 by all.

Industry also expected that students are at least aware of various process models and standards such as ISO, SEI-CMM, Six Sigma etc. They also emphasized on familiarity of students with various tools and techniques for version control and bug trackers etc.

We have observed that many companies tend to focus on contemporary skill requirements such as vendor certifications in a particular tool or technique.

But the academic curriculum is heavily governed by knowledge components. Some curricula provide for a project component for inculcating application skills. But rarely do a curriculum included problem solving skills, or best practices. Thus the curricula are mostly theory based.

Some academic programs include soft skills (communication, team-building) as a separate course offering. But communication specific to SE industrial practice is again rare.

2.3 Lack of Due Emphasis on SE in CS courses

Very few programs in Indian universities offer a Software Engineering major. Some IITs, IIITs (Indian Institutes of Information Technology) and private vocational institutes such as NIIT, APTech offer specialization in software engineering, but not at the under-graduate level. The IIITs are only a decade old, and produce about 100-250 graduates with SE specialization or with SE electives per year, which is a minuscule percentage of man-power requirement of industry.

Most universities offer SE as just one of the courses along with other CS courses. Graduates from such universities get to study SE for at most a semester. For some CS students this is the only opportunity to study Software Engineering before starting their career as Software Engineers.

2.4 Missing Emphasis on SE Research

Most of the Indian universities that offer CS courses are teaching universities. IITs, IIITs, and IISc conduct SE research but not on a large scale. There are a few prominent research centers focusing on software engineering research such as SETLabs at Infosys,
TRDDC of TCS, IBM India Research Lab and Microsoft Research.

Possible reasons for lack of research are as follows; barring a few premier institutions, Indian academic community is not research oriented as most of the universities are teaching universities. Hence, there is a dearth of advisors in the field of SE research. Also, owing to the social fabric, most students prefer to join high paying software development jobs, instead of getting involved with SE research.

Hence the country doesn’t have a strong SE research culture and hence misses on the benefits that research can bring to the industry.

3. PEDAGOGY ISSUES

3.1 Non-effective Classroom Teaching

Most of SE courses in India use the traditional lecture based teaching pedagogy. This kind of pedagogy is suitable for imparting theoretical knowledge. But SE is an engineering discipline and should focus on application of knowledge and skills and other practical aspects of software development. This is difficult in the traditional lecture based learning environments because the traditional methods do not provide opportunities for hands-on experience or practical training. Thus such environments are unsuitable for SE education. Other engineering disciplines include a lot of labs, which is missing in current SE education practice in India, especially in university environments.

3.2 Not So-successful Project Component

Professional courses in Computer Science i.e. B.Tech (CS), MCA and, BCA, include a mandatory project in their curriculum. The purpose is to bring forward the practical aspect of software development. But our observation is that this approach faces three major problems.

First, there is usually a one to two semester gap between attending the SE course and applying the learning from the course in the project. For example, the course is usually offered in the 3rd or 4th semester, but the project is offered in the 6th semester in a BCA or MCA course. Similarly, SE course is offered in 5th or 6th semester in a B.Tech, but project is usually a 7th or 8th semester activity. This time gap is sufficient for students to forget the basic principles by the time they need it.

Second, it is not mandatory that students opt for a software development project. Sometimes, even thought the project is a software development activity, students focus only on the end-product or the deliverable i.e. the code. They avoid learning of SE practices, tools and techniques, since this there is no proper assessment if the students explicitly used SE practices, tools or techniques for the same.

Third, it has been observed that most of the projects done at under-graduate level are academic in nature. Hence these projects do not represent the issues of scale and complexity of a real-world software development. But, most of the SE concepts and practices learned by students in the course are applicable for large scale complex projects. Thus, students find it difficult and irrelevant to apply these SE knowledge and skills for the academic scale projects, in turn get frustrated and develop the notion that Software Engineering is mere theoretical and not a useful subject [1], [6], [8].

4. RESOURCE RELATED ISSUES

4.1 Lack of Educational Infrastructure

India currently has around 347 institutes of higher education and 16,885 colleges with a total enrollment of over 9.9 million. They produce around 495,000 technical graduates, nearly 2.3 million other graduates and over 300,000 postgraduates every year [7].

IT sector (Application Development and Maintenance, product development) absorbs about 15% of the total engineering graduates. An important problem is that the sector is growing at a rate of about 30-35% per year, but enrollment in CS courses is not growing at the same rate. Also, the number of employable graduates is just 15% of the total graduate population. This will create a huge gap in demand and supply. NASSCOM ((National Association of Software and Services Companies, a consortium that serves as an interface to the Indian software industry and Indian BPO industry) fears that by 2010, the demand supply gap may be around 500,000 graduates [3], [4]. The issue has been recognized and there is pressure building on the academia to produce more employable graduates.

4.2 Large Class rooms

Following the software boom, the enrollments in CS courses increased multifold, though not sufficient to satisfy the requirements. Since the number of technical institutes that provide quality education is low, hence number of students enrolled in each class is high. The class size may climb above 100 in many institutes. Bigger class sizes negatively affect teaching efficiency. Additionally they are not conducive for non-traditional methods of teaching that can be more effective for SE education [2], [1]. [5].

4.3 Time Constraints

Software Engineering is a one semester course in most of the computer science programs. This usually means 2-3 hours of classroom teaching per week for about 16-18 weeks effectively. This time period is hardly sufficient to properly cover even the complete breadth of a SE course. Here, we refer to the course breadth guided by SWEBOK [7]. There is just not enough time for in-depth discussion of any topic. If some project work is also the part of this SE course, then students as well as faculty get more squeezed for time.

4.4 Lack of Qualified Teachers

Since the software boom, the number of Software Engineering course offerings has increased many folds, but not the number of qualified teachers. Most of the Software engineering teachers do not have a considerable experience of software development themselves. Their SE experience is limited to the SE course they have done as a student. Such teachers can talk about the theoretical aspects, but cannot discuss about the intricacies of software development or practice of software development in a convincing manner.

When given a choice of teaching a programming course and a software engineering course, most teachers opt for the programming course, because they do not enjoy teaching the SE course and are well aware of the reputation of the SE course among the students. Many of them opt to teach SE out of compulsion.
5. OTHERS CHALLENGES AND ISSUES

5.1 Programming is More Attractive than SE
Usually, students are more interested in learning programming as compared to Software Engineering. There may be many reasons for this observed phenomenon. One is that students are exposed to programming much earlier than SE. When they learn SE, they can not relate to the field, and find that following SE principles and practices would not help them in their programs (which are usually academic in nature and SE they learn is useful for building complex and large scale software systems). Students also tend to favor programming as they can see quick results there, unlike SE where results are intangible and documentation, design kind of jobs take up a lot of time when done correctly. Thus students favor programming over SE and rarely opt for advance courses in SE.

5.2 Lesser Inclination towards Higher Education
Indian social fabric is such that students are interested in joining a high paying job as soon as they finish up minimum educational requirements. Very few join post-graduate or research programs. Most of the students join post-graduate programs to improve their market value in the job market.

6. CONSEQUENCES
We have discussed various issues as observed by us in Indian SE educational scenario. These issues cause many short and long term people related problems in the software industry. Usually multiple issues are linked to a single problem. In this section, we look at the consequences or negative effects of these issues.

6.1 Missing the Holistic Picture
As an outcome of a faulty educational system, most of the software developers become task oriented, i.e. their focus is on the given task. They do not understand the context of work and miss the bigger picture. They intend to finish the given work by employing whatever learning they have and do not think about ways to improve the efficiency of task or effectiveness of deliverables. Thus, they do a task because it has been assigned to them and do not try to understand the rationale behind the task. This may lead to various issues, such as:

- Dislike towards work (lack the motivation). It is common observation that many employees dislike the process part of the projects they are working on. But companies have to employ processes for management and branding reasons. This creates un-necessary tensations between the management and employees, where employees feel that the process is un-necessary and a burden.
- Employees not working at maximum effectiveness level.
- Very few employees contribute to innovation, research or further development. Reluctance to adopt research and innovation coming from academia as they do not understand the benefits.

6.2 Low quality Products and Services
Software product or service quality is directly affected by the process quality and the engineer’s view and commitment to quality. But when employees do not understand and possess the essential SE knowledge and skills, the processes they follow are not effective and the product quality gets affected in turn. The engineer’s attitudes towards qualities of a product also affect the decisions taken by him, which in turn affect the quality.

For example, software quality gets affected by decisions taken during early development phases (analysis and design), but many software developers are not well versed with these SE skills, do not have a clear idea about customer requirements. Hence, even good programmers can’t produce end products of desired quality. The long term effect is on the brand value of Indian software industry. A side effect of this is that software verification and validation activities take up a lot of time and are disliked by majority programmers as they are not hard-core programming activities.

6.3 Additional Training Problems
Effectively, purpose of academia is to impart necessary programming and SE knowledge and skills to the students in order to prepare them for taking up professional responsibilities in industry. In this context, purpose of any industrial training should be to train new entrants on organization specific skills.

But current training programs have become re-training programs because incoming entry level programmers are not well versed with the basic programming and SE skills.

One important problem that arises is that since the entry level programmers lack vital industry skills such as self-learning, and effective communication, they spend more time in training, because these skills are required for learning newer skills as well.

Another problem is that the since the CS and SE programs throughout the country are not structured or standardized, hence training should bring all the entry level recruits to the same level.

Most industries would prefer the entry level recruits to be billable from the very first day after they join the company. Many client organizations do not like the working of new recruits of their important projects as they are not sure of their quality of work. Hence such recruits are a financial liability on the project for a while, because nobody has the confidence to give them tasks that demand responsibility. Also, a lot of training continues even after joining the project.

We have observed that the most of these programs consist of traditional lecture based teaching and performance based assessment that are not only less effective, but also labor intensive and require constant support from training staff or mentors. This increases the overall burden on staff as well as trainees.

Long duration training programs are a necessary evil, as they are expensive in terms of time, effort and money.

6.4 Inadequately skilled man-power
This is one of the most obvious challenge being faced by the Indian software industry. Indian industry is short-supplied not only with good programmers [3], but also with good software engineers.

When entry level employees are good at programming, they lack other essential knowledge and skills such as problem solving skills, self learning, communication skills, etc. Software maintenance and testing services are other important services being provided by the Indian software industry, but the current SE curricula either cover these topics very briefly or not at all.
Another issue is that these entry level programmers take up various important roles in the software development life-cycle in order to climb the value chain. These roles include requirement engineers, software and solution architects, testing leads, business analyst etc. All these roles and related responsibilities require higher order cognitive skills (problem solving, synthesis, evaluation etc.). They also require self-learning ability to learn and successfully perform these duties. Now our SE education system neither provides explicit role based training, nor does it facilitate the development of these higher order cognitive skills. Many organizations are of the view that they can and have been finding work-around to deal with lack of talent at the entry level positions, but the real problems in future will be to find the appropriately skilled personnel for these very specific and high caliber roles.

Indian Software industry is growing very fast and many organizations are trying to incorporate newer tools, techniques, methods and standards in their development process. But the developers do not possess the skills such as self-learning and analytical skills to understand and adopt newer technologies efficiently and effectively. Thus the research and innovation is not getting heartily absorbed by our industry, hence not really utilizing he opportunity to develop newer areas of competency.

6.5 Growth Issues

Indian industry is in the danger of losing its cost advantage in software engineering services to newer competitions like China, and Latin American countries [3]. Industry is looking at newer avenues of growth, which not only include basic software development services in more effective and efficient manner, but also adding value to the customer’s business by providing innovative software solutions. Possibilities of innovation lie in software products, SE tools and applications. Further, the efficiency and effectiveness can be achieved through deeper understanding of quality attributes pertaining to software product and software development process. But this seems like a tough challenge right now. This can be addressed by focusing on meta-level cognitive skills such as problem solving, innovation and efficiency aspects in addition to doing well in a task oriented way.

7. GUIDELINES

We believe that solutions to many of these problems require conceptual changes in our educational system. We need to prepare a learning environment that is conducive for proper alignment of our long and short term requirements (educational goals, expressed as curriculum), and our approaches towards instruction and assessment.

Guidelines given below will help in creating such an environment and would help to tackle the negative effect of various SEE challenges that threaten the Indian Software industry.

7.1 Identification and inclusion of Relevant SE skills in SE Curriculum

Survey results confirmed our belief that Software Engineering curricula require a radical shift from being just theoretical to a well balanced mix of theory (knowledge), skills and dispositions. The knowledge component consists of SE factual, conceptual, procedural and meta-cognitive knowledge (referred as SE knowledge further in this paper). Being an engineering discipline, SE education should include the SE skills, and application of these knowledge and skills into the curriculum. This is because computing and engineering services essentially mean application of computing science related knowledge and skills to solve practical problems in an efficient and effective manner [7]. It is important to note that problem solving involves deeper understanding and analysis of problem, evaluation of alternatives, decision making and finally application of programming and SE knowledge and skills. Thus including problem solving essentially means including the facilitation of these higher order cognitive skills in curriculum. Though it seems like a difficult task, but it is essential, because it is related to the nature of SE.

Since people work in large teams, hence communication and team skills are important. Self-learning ability is another such crucial skill because SE and IT are fast evolving fields. Technologies change very fast and it is important that the software engineers can update their knowledge independently.

SWEBOK [5], may help to come up with a well balanced and structured curriculum, especially for a breadth oriented or a first course in SE [7].

A proper, engineering focused curriculum would help to tackle the problem of inadequately skilled man-power and will have subtle but long term positive effect on quality of products and services and hence on growth of Indian Software industry.

7.2 Standardization of Curriculum

The SE curriculum can be divided into two parts: a) The basic or a first level course and b) advance topics. This first level course curriculum should include basic SE knowledge and skills along with problem solving skills. Soft skills should also be prescribed. Advance topics can be covered as per the requirements. At least the basic part should be standardized by policy makers and all education and training institutes should follow it.

Such an approach would bring quality control to SE education. The recruiters can be ensured that the entry level programmer is well versed with this common minimum knowledge and skills. Such standardization will facilitate in building accreditations programs accordingly for quality control of educational institutes.

The training staff is also benefited as now they can utilize their resources for training for the new entrants on organization specific requirements instead of basic SE.

7.3 Balance between Long-term and Short-term Goals

It is essential to strike a balance between long term learning goals such as fundamental SE concepts and enduring engineering knowledge/skills and short term learning goals such as technology or platform or domain specific skills. We suggest that incorporating contemporary process standards, tools and techniques, vendor certifications etc. in the curriculum will help to keep the students in pace with global advances in practice. Mahanty et al also confirm to this view in their study [2]. This balanced approach will help the software service organizations in brand building, and would provide an edge over competitors. A balanced curriculum would also help the training departments to focus on more important aspects of training instead of getting involved with training on basic SE knowledge and skills.
7.4 Pedagogy Reforms
The teaching pedagogy should be chosen as per the learning goals and should increase the effectiveness of SE education by making it contextual, interesting and learner centric [1]. Traditional class room teaching approaches do not work well for SE education owing to their suitability for teaching theoretical aspects effectively but not higher order cognitive skills. There are alternate and supplementary teaching approaches that facilitate the development of higher order cognitive skills in addition to the theoretical knowledge. Such pedagogies should be encouraged and adopted.

We suggest teaching pedagogies such as case study based approaches or project based approaches that simulate real world software development and maintenance (through a context) and where product as well as process related deliverables are well defined and duly evaluated.

When student’s can see the practical aspects of a theory or skill in a context, the learning is much more effective. Such contextualized learning is not only life-long, but also mitigates the people issues such as lack of interest and motivation in learning as well as teaching the subject. It is to be noted that motivation and interest in the basic course are the key to student’s endeavors in advance study or research in SE topics.

It is suggested that approaches such as project-centered learning or case studies adopted and made role based. This will help students to acquire role based expertise through understanding and practice of responsibilities associated with different roles.

It is important to note that non-traditional pedagogies are effective only when the assessment is also competence centered and not performance centered, and not just the theory, but all dimensions of learning (application of knowledge and skills, problem solving skills). Similarly, curriculum reforms would be effective only when teaching pedagogies would be well aligned with the educational goals.

7.5 Research Focus
SEE scenario in India requires advance courses that may attract students towards research. The teaching pedagogies can help here by helping to inculcate problem solving skills that will serve in motivating students and as a pre-requisite material for research.

7.6 Infrastructural Improvements
Lack of proper infrastructure can be a big hindrance to SE educational reforms. The number of institutes that offer CS and SE programs should be increased. But these should be regulated for a minimum basic curriculum, quality of teaching and facilities. These facilities should include well trained teaching staff, access to SE tools and study material.

Another important aspect is allotting adequate time to SE courses. One suggestion is to split the SE coursework over two consecutive semesters. Two semesters will help to cover the breadth of Software Engineering curriculum as well as provide adequate time for practicing the skills, in a consecutive or interleaved fashion.

Motivated, experienced, and well trained teachers are required to carry out the discussed reforms. Their activities would include design of proper curricula, conducting learning sessions and assessments, motivating students for deeper approaches towards learning etc. They should be well versed and experienced with software development and should keep their knowledge current. One way is to train the faculty in industry, or by bringing the practical knowledge to academia through a system of guest lectures and/or adjunct faculty from industry.

7.7 Industry–academia co-operation
Joint efforts of SE Industry and academia can help to alleviate many of the problems we discussed in section 6. It is important to understand that subjects like Software Engineering cannot be taught effectively without first hand experience. The industry should help the academia by providing internships for the teachers and also benefit from their wisdom and perspectives. Thus, academia and industry cooperation can only be a win-win for all the stakeholders and benefit the overall ecosystem of the IT in the country.

There are several initiatives already in place for improving the academia-industry cooperation. These include starting of new era educational institutions (like IIITs – the Indian Institutes of Information Technology) that focus on high end research and post-graduate education. These institutions are expected to come up as public-private-partnerships (PPP) and work very closely with industry to define their vision and goals.

Some other initiatives include offering of accreditation programs and finish the school certification programs by NASSCOM type of organizations, mentorship programs where industry partner will provide mentorship to the students, internships in the industry such that the students are exposed to real world and meaningful challenges early, train the trainer programs where college and university teachers are being trained by the industry experts on latest technologies[4].

8. CONCLUSIONS
In this paper we have tried to address India specific people issues in IT industry in general and software engineering education and training issues in particular. We have tried to make our study as objective as possible by conducting a survey and interviews with senior executives from the software services industry.

We have critically analyzed and classified various challenges according to deep rooted causes that are embedded in curriculum, pedagogy and assessment and infrastructure. We believe by addressing these challenges properly we will avoid the bottlenecks that will severely affect the growth of IT industry in future. We have suggested a few curriculum, pedagogy, assessment, and infrastructural reforms in Software Engineering education that will aid in overcoming the challenges we discussed.

9. ACKNOWLEDGMENTS
We acknowledge the inputs provided by respondents of our survey. We do not provide their details due to privacy issues. Their thoughtful responses have given us a deeper insight into the educational links between industry and academia and education and training related people issues.

10. REFERENCES


