December 2005

I Don't Remember What My Designation Is:Classifying Indian Software Workers

Vigneswara Ilavarasan
Indian Institute of Technology Delhi

Arun Sharma
Indian Institute of Technology Kanpur

I Don’t Remember What My Designation Is: Classifying Indian Software Workers

P. Vigneswara Ilavarasan
Indian Institute of Technology Delhi
evignesh@gmail.com

Arun Kumar Sharma
Indian Institute of Technology Kanpur
arunk@iitk.ac.in

Abstract

The success of Indian software industry is attributed largely to the quality of workforce it holds. Harnessing the software workforce is indispensable to India to maintain its growth in the global software services industry. Given this, need to reflect on the nature of software workforce necessitates an understanding of occupational classification in India. Review of literature shows that occupational classification of software workers is outdated, and is not adequately studied. On the basis of the work content, present paper classifies the software workers into four categories: writers, developers, designers and supervisors. Data for the paper was collected through task inventory and semi-structured interviews primarily from two software firms located in Bangalore.

Keywords: Software workforce; Occupational Classification; Software Work Content; Four categories; Bangalore; Indian Software Industry

1. Introduction

India is one of the developing countries to successfully exploit the opportunities created by the growth of global information technology (IT) industry (Schware, 1992). The success of Indian software industry has influenced many developing countries to target software sector as a catalyst for economic growth (Correa, 1990; Heeks, 1999; Heeks and Nicholson, 2002; Nicholson and Sahay, 2003). The existing investigations suggest that certain issues need to be addressed to maintain the stronghold in the growing global competition. The issues are (Arora. et al., 2001; Dossani and Kenny, 2002; D’Costa, 2002; Joseph, 2001; Heeks, 1996; Kumar, 2001; Nath and Hazra, 2002; Parthasarathy and Joseph, 2002):

- dominance of low skilled labor-intensive services
- lack of globally visible products, and poor product orientation and innovation
- weak domestic market
- over-dependency on US market
- poor venture capital environment
- scarcity of middle level managerial workers
- market dominance of fewer larger firms, and
- want of improved communication and internet infrastructure.

One issue, which is completely ignored by the literature, is the ‘occupational classification’ of software workers. This paper aims at exploring this.

Occupational classification systems define occupations in terms of selected attributes; and group them on the basis of these selected attributes into broader categories for purpose of statistical description and analysis (Levine et al., 1999). To quote TBCS (2002):

> The classification of work is fundamental to any human resources management system. In large and complex organizations, both public and private, it provides a means of grouping similar types of work together so that they can be ranked by levels of difficulty and differentiated from other, dissimilar work.

Despite the fact that the present software service capability of India was built by exporting workers to western countries (Athreye, 2002), and the declaration of Government of India (NASSCOM, 2002) that software industry is a thrust area for national development, detailed data on software workforce is not available. To quote Parthasarathy and Joseph (2002): 'It may not be an exaggeration to state that, paramount importance attached for so many years to software and service export not withstanding, reliable data on different aspects of this industry is difficult to obtain' (p.21).

According to an estimate, presently there are about 0.65 million software workers in India, and it is projected to reach 2.2 million in 2008 (DIT, 2003). The *Indian National Classification of Occupations* (1968) mentions four categories of software workers, namely, programmers, system analysts, automatic data processing operators, and computing machine operators. Programmers and system analysts are mentioned in Group 10 of Division 1. Group 10 denotes mathematicians, statisticians and related workers, and Division 1 denotes professional, technical and related workers. Automatic data processing operators and computing machine operators are included under Group 34 of Division 3. Division 3 denotes clerical and related workers, and Group 34 denotes computing machine operators. While the latter category has become redundant for the present period, the former category is grouped together with mathematicians and statisticians. This classification represents the period when software work had not emerged as an independent occupation in India. The same categories have been used for the population census in 1991 also. Thus it is clear that the present classification of software workers in India is inadequate to capture the workforce of today.

The need for the occupational classification of software workers becomes all the more necessary, as the Indian software industry continues to operate in the HR (human resources) augmentation mode (Tschang, 2001)[1]. The major part of industry revenues, 67%, is generated by software exports and services (Dataquest, 2003). It is also noted (Athreye, 2002) that even when Indian firms participate in the product segment, they retain the services activities to generate constant revenue. As the service-orientation nature of industry suggests, workforce is indispensable and India needs to caress the human resources to sustain its place in the global software industry.

Recently a 'Task Force on Human Resources Development in Information Technology' was established by Dept. of IT, Government of India to 'prepare a long-term strategy for significantly increasing the number of well trained IT professionals in line with economic projections' (Lakshminarayanan, p.1). Occupational classification is indispensable in executing this strategy. As Pollack et al. (2002) noted that occupational classification is important to facilitate cross-country statistics; examine occupational migration patterns; understand the nature of workforce

[1] Revenue increases with number of contracted projects and number of workers in the projects.

present in a country; monitor the changing occupational structure and related social change aspects; and devise policies which aim at balancing demand and supply of the workforce. This paper attempts to classify software workforce in India.

2. Literature survey

Software work includes all the activities directly involved in producing or developing software. It is also called as software engineering and programming. Software work was recognized globally as an occupation during early 1980s when software was introduced in the market as a commercial product (Greenbaum, 1976). Software production involves the following series of stages as noted in System Development Life Cycle (SDLC) model (Heeks, 1996): Analysis and specification of the software requirements → Designing → Coding/Writing → Testing → Delivery and installation → Maintenance. In general, software workers are classified on the basis of the stages in development or production of software. As Heeks (1996) wrote, "a set of different job roles" arises in connection with software production including *project manager*; *analyst*; *designer*; and *programmer* [who does coding and testing]" (p.27).

Kraft (1977; 1979) observed that occupational titles were not reflective of the actual job content in software work. He proposed that series of job titles would give an illusion of vertical mobility to the workers, but in actual content there would not be any significant change. Supporting Kraft's propositions, Loseke and Sonquist (1979) rejected prevailing occupational titles to classify the software workers. They prepared a comprehensive list of activities of software work and asked each worker in the sample how much time they spent on each activity. The workers were classified into four categories according to the tasks that formed the largest component of their tasks: craft persons, designers, implementers and maintainers or users. If the workers performed tasks that were distributed across all the stages of SDLC, they were called as craft workers. If the workers did a set of low-level activities they were called as maintainers. Orlikowsky (1988) criticized this study (for the following reasons: (a) the sample was confined to self-selected members of Association of Computing Machinery that would have more technical orientation than business orientation, and might not actually reflect the commercial workforce; (b) incorrect assumption that a worker (craft person in the proposed categories) would do all the activities in the project. Each stage in the software development requires lot of time and varied skill set to perform, and a single person cannot do all the work; (c) incorrect assumption that stages in the software development were sequential and non-overlapping, since the activities involved in software work like analysis, design and implementation task are typically iterative (Jensen and Tonies, 1979).

Similar approach was used by Kraft and Dubnoff (1986) to differentiate the workforce into various categories of workers. But their main objective was to test whether software work is task-fragmented. They did not propose a new classificatory scheme. Freeman and Aspray (1999) in their analysis of supply of IT workers in US classified the workers into four categories depending upon the functionality in their occupation: conceptualizers, developers, modifiers/ extenders and supporters/tenders. Conceptualizers are those who conceive and sketch out the basic nature of a computer system artifact. Developers are those who work on specifying, designing, constructing, and testing an information technology artifact. Modifiers/extenders are those who modify or add on to an IT artifact. Supporters or tenders are those who deliver, install, operate, maintain or repair an IT product.

In further explanation, Freeman and Aspray (1999) showed some existing job titles present in the industry under their classificatory scheme. Conceptualizers composed of systems analysts, system architects, and requirements analysts. Developers composed of system designers, programmers and software engineers. Modifiers or extenders included maintenance programmers, database administrators, software engineers, and computer engineers. But the authors failed to show the validity of the scheme on empirical basis. They did not use their own proposed categories in the further discussion of the study. They tend to use the same ambiguous explanations or titles as used in the other studies. For example, in the discussion about the necessary skills for an IT worker, they wrote: "A programmer needs to know how to design, program, test, debug, and modify programs" (p. 37).

According to the above approach, programmers should fall into the conceptualizer category, but they are classified under modifier and developer category. Thus the proposed classification seems to have rather low validity and cannot be used as a base for further studies.

Howard (2001) differentiated various types of the software workers based on the work personalities. He suggested what kind of software workers would be appropriate for a particular type of project undertaken. For example, software workers who are good at taking cost effective decisions, but relatively less inclined about perfection were called as 'prototypers'. Prototypers were best suited for the projects where system requirements were initially unclear and building the right system was more important than producing something quickly. 'Deliverers' are good as emergency software maintenance staff. They can be used effectively in short bursts on rapid application development projects. Other types of work personalities were 'perfectors', 'producers', 'fixers', 'convergers', 'divergers' and 'finishers'. This classification derived more out of personal experience rather than empirically validated procedures, and may be useful in identifying the right persons for a project inside the organization, but not in classifying workers at the industry level.

There have been attempts made by Governmental departments and other agencies to classify the software workforce. They have helped in understanding the nature of software workers but are not free from ambiguities. For instance, ITWDP of US (1999) delineated four occupational groups: computer scientists, computer engineers, systems analysts and programmers. But it failed to describe how these categories are arrived and what these categories of workers perform in their job. While describing about the job of programmer, *California Occupational Guide Number 81* (1999) mentions that programmer analyst is an alternative job title for the programmer. According to this, both programmers and programmer analysts form a single group. However, programmer analysts are considered to be a part of different occupational sub-group by others. *The World Employment Report* (2001) mentions 12 categories of software labor and compares the annual labor cost across eight countries including India. But the report does not give any description of the job titles. Also these categories are not used in the further discussions presented by the report.

Thus it is evident that a classification of workers in software work is still an unresolved issue, and there is a need to conduct systematic study to evolve the classification of workers in the software work.

3. Methodology

For classification of software workers this study employs the job inventory approach, used earlier by Kraft and Dubnoff (1986), and Loseke and Sonquist (1979). It involves developing a checklist of tasks that an incumbent on a given job might perform. The subject matter specialists develop the tasks, and then incumbents are asked to indicate which tasks they actually perform and/or the relative time spent on or relative importance of each (Nishi De, 1995). Sanchez and Levine (2001) note that ratings of task importance, time spent, and difficulty of learning provide relatively independent information. The present study focused on task content of the workforce and the workers were asked how frequently each task occurred in their work.

A list of tasks present in the software work was prepared for the study. Sixteen tasks under four stages of software development were borrowed from the questionnaire framed by Bailyn (1989) for her study on comparison of experiences between home-based and office-based system developers in a UK computer firm. Sixteen more tasks were added by consulting two experts. To confirm the exhaustiveness of the tasks listed and the clarity of the inventory, the task inventory was electronically sent to three software workers who had more than two years of experience in software work.

The final inventory consisted of 32 tasks, divided under six categories. These categories are mapped on the successive stages of the software production process (Annexure A). In the task inventory, each respondent was asked to mark how frequently each much of an activity occurred in his or her work. Five options were given for each task: (1) never (2) occasionally (3) sometimes (4) frequently (5) very frequently.

The data was collected between May 25, 2001 and July 30, 2001, primarily from a sample of two firms located in Bangalore: A Tech and B Tech (psydonyms). A Tech is a domestic company established in early 1980s in Bangalore and has 21 offices worldwide. It earned 98% percent of its revenue through software services. The major export destinations of A Tech are in US and Europe. In the financial year 2000-2001, it had a growth rate of 115.4% and a turnover of US $540 million. According to *Dataquest* (2001), a total of 9800 employees are working in A Tech. It has been rated as one of the top ten software firms in India for the last four years. B Tech is also a domestic company established in 1999 in Bangalore. It has co-headquarters in New Jersey and Somerset, USA. Nearly 66% of its revenue comes from onsite work in USA. During 2000-2001, it had a growth rate of 975% and a turnover of US $14.6 million. According to the Human Resources Manager in B Tech, the firm employed a total of 1500 workers.

The firms were selected using multistage cluster sampling method. One business unit with 450 employees from each firm was selected. In each unit, respondents were selected using quota sampling method. The workers in the firms were placed in ten hierarchy levels, approximately. In order to evolve a meaningful classification of workers, the locally existing hierarchy levels were ignored, and all the workers were clubbed into three categories: Level I, Level II, and Level

III according to their place on the hierarchical pyramid of the firm. Workers who are not involved in the software production or development process, like administration department, marketing department, were not included in the sample. Each level was taken as a quota, and workers were selected using incidental sampling method within each category. In each firm, 80 workers were given the inventory. Finally, a total of 114 filled inventories were returned.

In order to understand the work content of the workers more deeply, additional data was collected through interviews. The interviews were open-ended and semi-structured. Respondents were selected by incidental sampling method and efforts were made to include all the categories of workers. A total of 62 interviews were conducted. Questions were related to what the workers do in their job, how the work is allocated, and the nature of tasks performed. Additional 13 unstructured, open-ended interviews were conducted with the workers who are working in firms other than the sample. These interviews were conducted to check whether sample firms are similar to other firms in the industry.

In all 114 questionnaires were returned from the two firms, 60 from A Tech, and 54 from B Tech. In addition, a total of 62 interviews were conducted, 33 in A Tech and 29 in B Tech. The distribution of sample workers is presented in Table 1.

Although sample size of the study does not seem to be big enough for generalizations, earlier studies show (Arora et al., 2001; Nath and Hazra, 2002) that nature of work performed in the Indian software industry is homogeneous across firms, and thus meaningful inferences may be drawn even from a small sample.

------------------(Insert Table 1 about here)----------------------

4. Classification of Workers

It is observed that in the software organizations, work is executed in the form of projects. Each project involves certain roles to be performed by the workers. The roles in the project give a general outline of the activities to be performed by the workers in the project. It is found that roles performed by the workers in a project are considered more seriously than the organization status. One worker told, "I don't remember what my designation is. Here designations don't matter in work. The role matters in what you do in a project."

There are basically four roles present in any software organization. They are as follows: developers, module leaders, project leaders and project managers. In different organizations, however, different labels may be used. For example, module leaders are called the team leaders or technical leaders, and project managers are called the business managers. As one worker described, "How project is executed or work is done is constant in the industry or at least in big firms. Only in small firms it might differ. But otherwise it is all same, and designations change. Technical managers may be called as business manager or project manager." Another worker from A Tech opined: "Even in product firms, module leaders are there, project leaders are there and developers too. I feel roles are same in any project irrespective of the company."

A value hierarchy is perceived among the roles as follows: Project managers → Project leaders → Module leaders → Developers. Project managers are at the top and developers at the bottom.

The organizations do not have definite role descriptions like job description. No recorded or written guidelines exist in organizations to divide the work among roles in the project. A worker said, "We don't have any rule book kind thing that says that role should be like this, and the work should be given to them like this." Although role descriptions are not present, a general outline of activities associated with the roles in the organizations could be deducted.

Project managers are the workers who handle many projects at the same time. They may handle the projects of a single client or many clients. Project managers are given responsibility of interacting with clients, making work schedules, deciding project milestones, calculating man-hours, selecting tools and methodologies, separating modules, and allocating work to the teams. They interact with the clients in the initial or formalization stages of the project. They interact with the technical team of the clients, and work as a medium between the organizational team and the clients. They work closely with the team till the project moves into coding stage. When the project gets stabilized, responsibility of the project would be given to the project leaders.

Project leaders assist the project managers during initial stages of the project. They gather specifications, write program specifications, create high level designing and coordinate with the module leaders. Project leaders are in-charge of projects. They are responsible for project execution and delivery to the clients. They handle one project at a time, and work as the coordinator of the project. They work as middle persons between the rest of the team and the project manager.

The rest of the team consists of module leaders and developers. If the project is big enough to be divided into modules, role of module leaders is introduced in the project. The module leaders are primarily responsible for the modules. They assist the project leaders in high-level designing, and do low-level designing part of the project themselves. They work as middle persons between the developers and the project leaders. The module leaders also share the work of developers. The developers are inducted in the project, once it is stabilized. The developers are the workers who actually execute the work. They do the coding according to the design given to them and also debug the program. They report to the module leaders.

In many cases, project managers perform the role of project leaders. The roles 'project leaders' and 'project managers' were used alternatively by the workers, denoting a similar meaning. One reason cited by the worker who has assumed both roles is that, "Whatever projects we have are older ones. We don't have any new projects. If new projects are coming in, I would handle more projects, and help of project leaders needed."[2]

The number of roles in a project varies, and depends on certain factors like project size, stage in SDLC of the project, and client demands. As one B Tech worker opined, "I think many things depend on project size. If the project is small, module leader may work as a project leader also. When the size becomes big, you may need more people to manage things and you have module leader, team leader etc. And it becomes too big, you may need more people to manage things and you have module leaders, project leaders etc. And it becomes very complicated in structure."

[2] The data was collected during the months of May, June and July, 2001. During this period, Indian software industry was severely affected by the economic slow down in US (Swami, 2001)

It was very commonly expressed by the workers that roles are flexible, and might change from project to project. As one worker said, "You see…the roles are flexible and interchangeable. The main motive is to finish the project in time and in an effective manner." Yet one may mark a consistency in the process by which a worker climbs the role ladder. A worker starts his career as a developer and performs this role for about a one year. In the second year, he performs developer and module leader roles alternatively. In the third year he performs module leader and project leader roles. In the fourth year he starts performing the role of project leader. In fifth or sixth year he starts performing the role of project manager. After performing the role of module leaders for considerable amount of time, the workers do not share the work of developers. The researcher is not able to find even a single project leader or project manager performing the role of a developer in subsequent projects. In a few specific cases where the project is low skilled in nature, and does not need any complicated design, a senior developer would perform the role of a project leader. In these cases, he would be reporting to the project manager. An earlier study (Arora et al., 2001) also noted that Indian software workers are promoted to managerial positions based on seniority in the job rather than proven managerial capabilities of the workers.

The three levels taken by the study were combined with the roles of the workers indicating a division of labor in the workforce. There was a high frequency of Level I workers performing the role of developers; Level II as module leaders; and Level III as project leaders and managers. Few workers at any level performed the roles of workers at the next level. When this relationship was reported to the Human Resources Manager in B Tech, and a senior worker in A Tech, they also agreed. Thus the number of roles, four, could be taken as a boundary for classifying workers in the industry.

By performing non-hierarchical cluster analysis, workers are divided into four categories (Table 2). Non-hierarchical cluster analysis is preferred when a priori number of clusters is known before classifying the subjects (see Ketchen and Shook, 1996 for further details).
-------------------(Insert Table 2 about here)---------------------

The workers are differentiated based on the core activities performed by them. The core activities consist of all tasks that are performed 'frequently', and 'very frequently' by the worker. Based on the core activities performed by the workers, four categories are labeled as (Table 3): (1) software writers (2) software developers (3) software designers, and (4) software supervisors.
---------------------(Insert Table 3 about here)-----------------

Since the study is confined to only software industry, the prefix software is dropped for the further discussion. The composition of workers in the sample is presented in Table 4.
--------------------(Insert Table 4 about here)------------------

4.1 Defining Categories of Workers

The Indian software workforce is categorized into four categories: writers, developers, designers and supervisors. Workers who code, debug, test the program, and document the activities of the project are called the writers. These activities are relatively low skilled when compared with the other tasks related to the designing phase of the software production. Since these workers do not perform any high skilled activities, and perform only coding and preliminary testing of their codes they are called the writers.

Developers are the workers who write specifications of the program, create high level and low level designs, test the system, review others' work inside the project, and perform writers' activities. These workers perform more high skilled work than the writers. Presence of tasks related to three phases designing (determining specifications of the programs, high level and low level designing), writing (coding, and documentation), and testing (program testing and system testing), shows that these workers contribute more in developing the software. Hence they are called the developers.

Designers are the workers who perform both high-level and low-level designing activities very frequently along with testing the linkage between the modules, and reviewing others' work inside the project. They also perform activities related to a conception of the idea for software and preliminary analysis like determining users' requirements, estimating the cost and time for the project, finding out whether the project is technically feasible or not, writing system specifications in both users' and technical terms, and coordinate the workers both at module and project level. Since these workers perform high-level and low-level designing very frequently, and other tasks related to designing and conception part of the software production frequently, they are called the designers.

The last category of workers, supervisors perform selected activities of the designer. They determine the users' requirements, estimate cost and time of the project, find out technical feasibility and scope of the project, review others' work in the project, and manage the project. They are responsible for all managerial aspects of the project. They also evaluate the work of other team members for promotion. This is an exclusive right of supervisors. Designers do not make any evaluation. Since these workers are performing only coordinating activities as their core activities, they are termed as supervisors.

5. Discussion

Thus, Indian software workers can be classified into four categories: writers, developers, designers and supervisors. The tasks performed by each category of workers under five levels varying from 'very frequently performed' to 'never performed' are presented in Table 5. These categories can be used to classify the software workforce in India.

----------------------(Insert Table 5 about here)-------------------

Since it appears that task content is overlapping across the categories of workers (Table 3 and 5), Kruskal-Wallis test was conducted to find whether the frequencies of tasks differ significantly across all the categories of workers. The findings support the validity of the suggested classification and show that the categories of workers differ significantly on all the tasks (Table 6). It may be stressed that although different categories of workers perform the same task, the amount they perform differs. For example, both designers and supervisors perform the task 'project management'. The mean value of this particular task is greater for supervisors (4.45) than designers (3.78). Similar pattern is present between designers and developers in the task ‘system testing’. Although both the categories noted in the task inventory that they 'frequently' perform ‘system testing’, mean value is greater (4.44) for designers than developers (3.65).

---------------------(Insert Table 6 about here)-----------------

Of the listed activities in the task inventory, 14 (see Table 2 and 3) activities do not form a part of core activities for any of the four identified categories of workers. These 14 activities can be called as peripheral activities. Activities 1, 2, and 3 are relatively highly skilled activities in software development, and are not present in the core activities of the workers. This absence is due to the dominance of low-skilled software work present in India. As Miller (2000) observed:

> Far from engaging in the high value added, creative and design intensive practices of software development that are today typical among software producing firms in the industrialized countries, professional software trade among firms in developing is commonly limited to low value, repetitive, activities like coding and testing. (p.2254)

Activities 26, 27, 29, and 32 are more management-oriented activities and there are specialized people to execute them in the organizations. Since the sampling frame of the study included only workers who are directly involved in the software production process, these activities do not form a part of core activities.

Since, Indian software industry executes only a part of the software development in the form of services, activities 19, 20, 21, 22, 23, 24 and 25 are not performed as core activities. These activities tend to be performed by the organizations that deliver the final product or service solutions to the end-users.

6. Conclusion

The study showed that Indian software workforce could be classified into writers, developers, designers and supervisors. The workforce largely reflects the later stages in the software development. This is due to the fact that Indian software industry is dominated by exports of labor-intensive services, which are outsourced by the Western clients. These services are mostly at later stages of software development like low level designing, coding etc. The Indian software workforce is recruited to meet the technical needs of their clients and engaged in associated managerial requirements. The workforce will tend to get re-categorized if the industry moves towards product market deviating from the present services mode, or if the requirements of the outsourced projects demand use of high skills. This may result in emergence of separate technical occupational groups that work on initial stages of software development like system architecture, software architecture etc.

7. References

Arora, A. et al. (2001) “The Indian Software Services Industry”, *Research Policy*, 30: 1267-1287.

Athreye, S. (2002) "The Indian Software Industry and Its Evolving Service Capability." Working Paper, Development Policy and Practice, The Open University, UK.

Bailyn, L. (1989) “Toward the Perfect Workplace”, *Communications of the ACM*, 32 (4): 460-471.

California Occupational Guide Number 81 (1999) Online posting. <www.calmis.ca.gov/file/occguide/COMPUPRO.HTM> Downloaded on 8 October.

Correa, M. C. (1990) "Software Industry: An Opportunity for Latin America", W*orld Development*, 18 (11): 1587-1598.

Dataquest. (2001) "The DQ Top 20 Giants", *Dataquest*, 15 July.

----. (2003) "The DQ Top 20", *Dataquest*, 15 July.

D'Costa, A. P. (2002) "Export Growth and Path Dependence: The Locking-in of Innovations in the Software Industry", *Science, Technology & Society*, 7 (1): 51-89.

DIT. (2003) "Background note for Task Force on HRD in IT", *Department of Information Technology*, Government of India, New Delhi. Downloaded on 10 November. <http://www.mit.gov.in/eiel>

Dossani, R. and M. Kenny. (2002) "Creating Environment for Venture Capital in India", *World Development*, 30 (2): 227-253.

Freeman, P. and W. Aspray (1999) *The Supply of Information Technology Workers in the United States*, Washington: Computing Research Association.

Greenbaum, J. (1976) "Division of Labor in the Computer Field", *Monthly Review*, 28: 41-55.

Hari, P (2001) "Bangalore: Technopolis", *Business World*, 26 February.

Heeks, R. (1996) *India's Software Industry, State Policy, Liberalization and Industrial Development*, New Delhi: Sage Publications.

----. (1999) "Software Strategies in Developing Countries", *Communications of the ACM*, 42 (6): 15-20.

---. and B. Nicholson. (2002) Software export success factors and strategies in developing and transitional economies. *Development Informatics Working Paper Series*. No. 12, Institute for Development Policy and Management, Univ. of Manchester.

Howard, A. (2001) "Software Engineering Project Management", *Communications of the ACM*, 44 (5): 23-24.

ITWDP. (1999) "Core Occupations of the U.S Information Technology Workforce" *Information Technology Workforce Data Project*, Online posting, January. <http://www.uefoundation.org/itworkfp.html>

Jensen, R. W. and C. C. Tonies. (1979) *Software Engineering*, New Jersey: Prentice Hall, as cited in Orilikowski, W.J. (1988) "Data Processing Occupation: Professionalization or Proletarianization", *Research in Sociology of Work*, 4: 95-124.

Joseph, K. J. (2001) "ICT Boom, Economy and Labour", *Labour and Development*, 7(2):1-36

Ketchen, D.J. and Shook, C.L. (1996) "The Application of Cluster Analysis in Strategic Management Research: An Analysis and Critique", *Strategic Management Journal*, 17: 441 - 458.

Kraft, P. (1977) *Programmers and Managers: the Routinizing of Computer Programming in the United States*, New York: Springer Verely.

---. (1979) "The Routinizing of Computer Programming" *Sociology of Work and Occupations*, 6 (2): 139-155.

Kraft, P. and S. Dubnoff. (1986) "Job Content, Fragmentation and Control in Computer Software Work", *Industrial Relations*, 25 (2): 184 – 196.

Kumar, N. (2001) "Indian Software Industry Development: International and National Perspective", *Economic and Political Weekly*, 36 (45): 4278-4290.

Lakshminarayan, S. (2003) Taskforce on HRD in IT. Notification. Sept. 28. <http://www.mit.gov.in/eiel/notification.asp> Downloaded on 10 November

Levine, C. et al. (1999) “Revisiting the Standard Occupational Classification System”, *Monthly Labor Review*, May: 36-45.

Loseke, D. and J. A. Sonquist. (1979) “The Computer Worker in the Labor Forces: New Occupations and Old Problems” *Sociology of Work and Occupations*, 6 (2): 156-256.

Miller, J. (2000) “Sustaining Software Teletrade in Bangalore, Fostering Market Agility Through Economic Competence”, *Economic and Political Weekly*, June 24: 2253-2262.

NASSCOM. National Association for Software and Software Companies, New Delhi. <www.nasscom.org>

Nath, P. and A. Hazra. (2002) Configuration of Indian Software Industry. *Economic and Political Weekly*, 37 (8): 737-742.

Nicholson, B. and S. Sahay. (2003) Building Iran’s software industry: An assessment of plans and prospects using the software export success model. *Electronic Journal of Information Systems in Developing Countries*. <www.ejisdc.org> (forthcoming).

Nisi De, A. (1995) “Job Analysis”, In: Nicholson, N. (ed). *Encyclopedia Dictionary of Organizational Behaviour*, Oxford: Blackwell Publishers; 262-263.

Orlikowski, W.J. (1988) “Data Processing Occupation: Professionalization or Proletarianization”, *Research in Sociology of Work*, 4: 95-124.

Parthasarathy, A. and K. J. Joseph. (2002) “Limits to Innovation with Strong Export Orientation: The case of Indian ICT Sector”, *Science, Technology & Society*, 7 (1): 13-51.

Pollack, J. L. et al. (2002) “A Common Language for Classifying and Describing Occupations: The development, Structure, and Application of the Standard Occupational Classification”, *Human Resources Management*, 41 (3): 297-307.

Rosenberg, D. (2002) *Cloning Silicon Valley: The Next Generation of High-Tech Hot Spots,*, New Jersey: Prentice Hall.

Sanchez, I. J. and Levine, L. E. (2001) “The Analysis of Work in the 20th and 21st Centuries”, In: *Handbook of Industrial Work and Organizational Psychology*, Volume I, Personnel Psychology, London: Sage Publications: 71-89.

Schware, R. (1992) “Software Industry Entry Strategies for Developing Countries; A “Walking on Two Legs” Proposition”, *World Development*, 20 (2): 143-164.

Swami, P. et al. (2001) “Can Software Shine Again?”, *Business India*, 3 February: 46-52.

TBCS. (2002) “About Classification”, *Treasury Board of Canada Secretariat*, Online posting, downloaded on 10 June. < http://www.tbs-sct.gc.ca/classification/AboutUs/JobClass.asp>

Tschang, T. (2001) *The Basic Characteristics of Skills and Organizational Capabilities in the Indian Software Industry*, ADB Working Paper *13*, Asian development Bank Institute, Tokyo, Japan.

World Employment Report 2001. (2001) *Life at Work in the Information Economy*, International Labour office, Geneva.

Appendix A: Tasks included in the Stages of Software Production Process

1. Conception of idea for the software and preliminary analysis

Selling an idea to potential user; Determining user requirements; Estimating cost and time; Technical scooping / feasibility of the study; and System specification in users / business terms

2. Designing

System architecture; Software architecture; System specification in technical terms; Writing program specifications; High-level designing; and Low-level designing

3. Writing

Coding; Documentation; Preparing house keeping programs / data files

4. Testing

Debugging; Program testing; Link testing; System testing; Internal reviewing; and External reviewing

5. Delivery and Installation

User guides; Training users; First level support to the users; System maintenance and support

6. Others

Recruitment; Training of members; Evaluation of members; Firm level policy making; Module management; Project management; and General administration

Table 1: DISTRIBUTION OF SAMPLE WORKERS

	Position of workers in the organization						Total	
	Level I		Level II		Level III			
	TI	I	TI	I	TI	I	TI	I
A Tech	33	11	12	10	15	12	60	33
B Tech	30	12	9	7	15	10	54	29
Total	63	23	21	17	30	22	114	62

Note: TI = Workers who filled task inventory; I = Workers who are interviewed.

Table 2: CLUSTERING OF RESPONDENTS USING THE AMOUNT OF EACH TASK PRESENT IN THEIR JOB

No.	***Tasks***	***Amount of Task Present in each Cluster***			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4
1	System architecture	1	2	2	2
2	Software architecture	1	1	2	1
3	Selling idea to users	1	1	0	2
4	Determining users' requirements	3	3	2	1
5	Estimating cost and time	3	3	2	1
6	Technical scoping	3	3	2	1
7	System specification in user terms	3	1	2	1
8	System specification in technical terms	3	1	2	1
9	Writing program specification	3	2	1	3
10	Hi-level designing	4	1	2	3
11	Low-level designing	4	2	1	3
12	Coding	3	2	4	3
13	Debugging	3	2	4	3
14	Documentation	3	2	3	3
15	Program testing	4	2	3	3
16	Link testing	4	2	1	1
17	System testing	3	2	1	3
18	Internal reviewing	4	3	1	3
19	External reviewing	1	1	2	2
20	Auditing	1	2	0	0
21	Preparing housekeeping programs	1	0	2	2
22	User guide preparation	1	0	2	2
23	Training users	1	2	2	2
24	First level support to users	1	0	2	2
25	System maintenance and support	1	2	2	2
26	Recruitment	2	1	0	0
27	Training	1	2	2	2
28	Evaluation of members	1	3	0	2
29	Firm-level policy making	2	2	0	0
30	Project management	3	3	0	2
31	Module management	3	1	2	1
32	General management	1	1	0	0

Note: 0-Never; 1-Ocassionaly; 2-Sometimes; 3-Frequently; 4-Very Frequently.

Table 3: CORE ACTIVITIES PERFORMED BY THE CATEGORIES OF WORKERS

Category of Workers	Core Activities Performed in the Work
Writer	Coding; Debugging; Documentation; Program testing
Developer	Writing program specification; High level designing; Low level designing; Coding; Debugging; Documentation; Program testing; System testing; Internal reviewing
Designer	Determining users' requirements; Estimating cost and time; Technical scoping; System specification in user terms; System specification in technical terms; Writing program specification High level designing; Low level designing; Coding; Debugging; Documentation; Program testing; Link testing; System testing; Internal reviewing; Module management; Project management
Supervisor	Determining users' requirements; Estimating cost and time; Technical scoping; Internal reviewing; Evaluation of members; Project Management

Table 4: DISTRIBUTION OF CLUSTER ANALYTIC CATEGORIES OF WORKERS IN THE SAMPLE

Category of Workers	Frequency
Writer	48(42%)
Developer	26(23%)
Designer	18 (16%)
Supervisor	22(19%)
Total	114(100%)

Note: Figures in the parentheses represent percentage of workers in each category.

Table 5: WORK CONTENT FOR CATEGORY OF WORKERS

Frequency of the tasks performed	Activities
Writer	
Very frequently	Coding; Debugging
Frequently	Documentation; Program testing
Sometimes	Writing program specifications; Low level designing; Link testing; System testing; Internal reviewing
Occasionally	System architecture; Software architecture; Determining users' requirements; Estimating cost and time; Technical scooping; System specification in user terms; System specification in technical terms; High level designing; External reviewing; Preparing housekeeping programs; User guide preparation; Training users; First level support to users; System maintenance and support; Training; Module Management.

Never	Selling idea to users; Auditing; Recruitment; Evaluation of members; Firm-level policy making; Project management; General Management
Developer	
Very frequently	
Frequently	Writing program specification; High level designing; Low level designing; Coding; Debugging; Documentation; Program testing; System testing; Internal reviewing
Sometimes	Determining users' requirements; Estimating cost and time; Module Management; System specification in technical terms; System specification in user terms; Technical scoping; Software architecture; Link testing; General management
Occasionally	Project management; Evaluation of members; External reviewing; First level support to users; Preparing housekeeping programs; Selling idea to users; System architecture; System maintenance and support; Training; Training users; User guide preparation
Never	Firm-level policy making; Recruitment; Auditing; General management
Designer	
Very frequently	High level designing; Low level designing; Program testing; Link testing; Internal reviewing
Frequently	
Sometimes	Software architecture; Evaluation of members; External reviewing; First level support to users; Preparing housekeeping programs; Selling idea to users; System architecture; System maintenance and support; Training; Training users; User guide preparation; Auditing
Occasionally	Firm level policy making; Recruitment
Never	
Supervisor	
Very frequently	
Frequently	Determining users' requirements; Estimating cost and time; Technical scoping; System specification in technical terms; System specification in user terms; Writing program specification; Coding; Debugging; Documentation; System testing; Module management; Project management
Sometimes	High level designing; Software architecture; External reviewing; Selling idea to users; General management; Recruitment; Module management; System specification in technical terms; System specification in user terms
Occasionally	Link testing; Low level designing; Program testing; System architecture; System maintenance and support; Training; Training users; Auditing; Firm-level policy making; Coding; Debugging; Documentation; System testing; Writing program specification
Never	First level support to users; Preparing housekeeping programs; User guide preparation

Table 6: KRUSKAL-WALLIS TEST TO EXAMINE THE DIFFERENCES BETWEEN THE CATEGORIES OF WORKERS ON THE LISTED TASKS

No.	Tasks Listed in the Inventory	Chi-Square	df	Asymp. Sig.
1	System architecture	23.537	3	.000
2	Software architecture	24.405	3	.000
3	Selling idea to users	40.643	3	.000
4	Determining users' requirements	40.215	3	.000
5	Estimating cost and time	50.823	3	.000
6	Technical scooping	28.963	3	.000
7	System specification in user terms	54.068	3	.000
8	System specification in technical terms	48.837	3	.000
9	Writing program specification	33.086	3	.000
10	High level designing	48.248	3	.000
11	Low level designing	31.553	3	.000
12	Coding	53.669	3	.000
13	Debugging	52.538	3	.000
14	Documentation	37.946	3	.000
15	Program testing	44.136	3	.000
16	Link testing	39.001	3	.000
17	System testing	28.067	3	.000
18	Internal reviewing	16.662	3	.001
19	External reviewing	23.736	3	.000
20	Auditing	30.410	3	.000
21	Preparing housekeeping programs	34.268	3	.000
22	User guide preparation	24.334	3	.000
23	Training users	9.680	3	.021
24	First level support to users	23.977	3	.000
25	System maintenance and support	22.147	3	.000
26	Recruitment	47.741	3	.000
27	Training	24.381	3	.000
28	Evaluation of members	84.651	3	.000
29	Firm level policy making	33.208	3	.000
30	Project management	77.279	3	.000
31	Module management	44.034	3	.000
32	General administration	35.818	3	.000