# "PRIVILEGED PRECARIAT" OF INDIA'S SOFTWARE INDUSTRY

By

Madhumanti Sardar

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Development Studies- Sociology)

at

UNIVERSITY OF WISCONSIN-MADISON

2019

Date of final oral examination: 04/16/2019

The dissertation is approved by the following members of the Final Oral Committee:

Gay Seidman, Professor, Sociology
Jane Collins, Professor, Community and Environmental Sociology
Brad Barham, Professor, Agricultural and Applied Economics
Stephen Young, Assistant Professor, Geography
Jeremy Foltz, Professor, Agricultural and Applied Economics

## ABSTRACT

India's software industry jobs are filled by the privileged upper caste and top income earning class. US based MNCs subcontracted their routine, modular processes to save on labor costs. These software jobs are monotonous, low in technical skill with precarious labor conditions. Managerial roles are the only upward mobility opportunities for members of this privileged class and most of them cannot reach these positions. Mobility from the low-end software jobs into high-end, high skill cutting edge software jobs is minimal. Thus many members of India's privileged professional class remain "stuck" in low value, poor quality software jobs, making them "privileged" and "precariat".

*Dedicated to Seth, my best friend, philosopher, emotional support and now husband. You make me laugh, push me to stretch my boundaries, explore new worlds and ideas. Fellow partner in the journey of life, you are the best thing that happened to me.*

## ACKNOWLEDGEMENTS

I thank my advisor Gay Seidman, Professor of Sociology, UW-Madison. She has been a mentor, teacher and guide right from the time I joined in Fall 2011. She assigned readings at the right moment, nudged me to ask the right questions and tirelessly read through my drafts and gave me constructive comments.

I thank all the respondents, employees of Indian software industry that agreed to be interviewed for this research. I am indebted to your generosity in taking out time from extremely busy schedules and accommodating my interview requests. For the sake of anonymity I am not naming you. I thank all the coordinators and members of ITEC that generously gave me their time and insights into working in the software industry, especially the labor and gender issues and for putting me in touch with several other members for conducting interviews. Also big thanks to volunteers, students and parents of non-profit FSMK, Bangalore.

I am grateful to my friends and acquaintances in the Indian software industry especially, Vrushali and Sachin and my roommates in Bangalore. All of you got me interested in researching the software industry. And also Ravi, Raju, Ajeethji, Nidhin, Sowjanya, Ashwathi, Akash, Priya, Gautam, Manoj ji, Deepanwita and Rajeshwari for allowing me to ask initial uninformed questions. I am using your first names to maintain anonymity. All of you opened my eyes to the realities of software employment in India. I also thank Uma Kandasarma for putting me in touch with several of her friends.

I am indebted to the instructors and coordinators of the Software Technology Program for sharing their insights and experiences. Especially Ashok Tamhankar, Prof Ranganathan, Prof Rajagopalan. And all the students of the program that graciously gave me their time.

I am indebted to Balaji Parathasarthy (Professor, Indian Institute Information Technology- Bangalore) and Carol Upadhya (Professor, National Institute for Advanced Studies) for generously accommodating me, sharing research insights and helping me get in touch with key informants.

I must thank my professors at TISS, my field work supervisor, Prof Rajshri Mahatani for teaching me to make detailed field based observations and grounded theory and Prof Parthasarathy Mondal, my MA research thesis supervisor for “kindling” the initial spark of interest in academic research on social realities.

My gratitude to Becky and Rod Lakowske for being my family in a new, foreign land and all the generous warmth you have given me over the years.

Thanks to Kath Loving, manager of Healthy Campus, University Health Services. You are one of the most compassionate and student centric supportive supervisor/ manager I have ever come across.

Finally I am grateful to Shri Chakradhar for everything.

# TABLE OF CONTENTS

# INTRODUCTION

*A study conducted across 150,000 engineering students over 650+ engineering colleges in multiple Indian states found a software career as highly valued. Aspiring Minds, the employment solutions company conducting this study found that 97 per cent of graduating engineers in 2015 wanted jobs either in software or core engineering*[1].

Software jobs are a popular career choice amongst India's educated young adults. Software industry jobs are coveted for relatively high incomes, a ticket to middle class lifestyle and high social status. Working for reputed global MNCs, Indian software employees enjoy high occupational status and opportunities for global mobility. For many Indians, gaining entry into the software industry is an upward mobility in itself.

The industry is symbolic of India's "new economy": the decline in state sector employment and emergence of free market, export-based jobs. India's new economy is a consequence of deregulation and privatization reforms wherein well-paid, high status jobs moved from state owned, public sector corporations into the private sector.

As employees of India's "new economy", software employees are a part of India's "new middle class[2] (Deshpande 2003; Fernandes 2006; Upadhya 2011). Estimates by (Birdsall 2015)[3] and research by (Krishna and Bajpayi 2015) on income and consumption show India's "upper middle class" strata, earning at least $10/day or more,

---

[1] Chakrabarty, Roshni (2016). "Only 7 Percent Engineering graduates Employable: What's Wrong With India's Engineers?" http://indiatoday.intoday.in/education/story/engineering-employment-problems/1/713827.html

[2] There are disagreements on what constitutes the "middle class" of India. For instance see Krishnan and Hatekar (2017) for new estimates; Birdsall (2015) for middle class income based strata; Krishna and Bajpai (2015) for literature review on India's newly emerging middle class; Kapur, Sircar and Vaishnav (2018) on self reported methodologies for estimating the middle class. In 2014, with the lower limit of "middle income" group at $2/day, just above the official poverty line of $ 1.90/day, a majority of this so called "middle class" is very poor, earning between $2-$4/day, followed by those earning under $10/day. Both groups are too poor to sustain a "middle class" lifestyle and consumption. Also Biswas (2015), "Is India's Middle Class Actually Poor?" https://www.bbc.com/news/world-asia-india-41264072

[3] According to Birdsall (2015), to capture the variations "middle class" in developing countries like India may be divided into three income-based strata: "lower middle", "middle-middle" and "upper middle".

have significant disposable income to sustain a middle class lifestyle. This definition places Indian consumerist, upwardly mobile, "new middle class" at under five percent of the population. In absolute numbers this number is between 50-70 million people[4], given India's massive population of 1.7 billion.

The top five percent income earners of India, earning well over $10/day[5] make up most of the software industry employees. This "new middle class" is also known as the "global middle class"[6] or "cosmopolitan class" because they are the beneficiaries of high paying service sector jobs in India's globalized, liberalized economy. This higher class is largely upper caste in composition.

But the term "middle" class is confusing because it means middle-income earners. Instead I call them the "professional class" indicating the top-most income earning class, employees of India's software industry. Employing less than five percent[7] of Indians, the industry is urban; mega-city biased and employs highly educated labor. Main centers of software industry concentration are Bangalore, Hyderabad, Gurgaon (NCR) and Pune.

*When Raj joined the software industry in 1999, it was just starting to take off. His Bangalore based employer was a renowned US MNC. According to him, "Joining this software job was a big achievement for me. I got good pay but the feeling of working for*

---

[4] Meyer and Birdsall (2012), estimate India's high income "middle class"($10/day) is 70 million people. https://www.cgdev.org/sites/default/files/archive/doc/2013_MiddleClassIndia_TechnicalNote_CGDNote.pdf

[5] Putting it into perspective, salaries in India's global MNC call centers, considered "upper middle class" jobs pay approximately $10/day ($300/month). Software industry salaries are much higher and at the minimum in par with call center pay. See Murphy (2011), "Indian Call Centre Workers: Vanguard of a Global Middle Class?"; Majumdar (2010). "Globalization and Relative Compensation in India's Information Technology Sector"; Walker and Morgan Hartley (2012), "The Culture Shock of India's Call Centers". https://www.forbes.com/sites/morganhartley/2012/12/16/the-culture-shock-of-indias-call-centers/#576aa1af72f5

[6] Kochar (2015) "A Global Middle Class is More Promise than Reality". https://www.pewresearch.org/global/2015/07/08/a-global-middle-class-is-more-promise-than-reality/ Also see Krishnan and Hatekar (2017), Birdsall (2015), Radhakrishnan (2011).

[7] The estimate includes software, Business Processing Outsourcing (BPO) and call center industries making up the entire IT and ITES sectors.

*a top notch US MNC was exhilarating. I worked in this ultra chic, office building- a high rise designed in the most modern fashion with central air-conditioning. Those days even the top-notch bureaucrats in India didn't have such luxurious workplaces! Then it had all these fancy things like a stylish cafeteria giving free food to all employees, well-manicured lawns and walkways. Every thing just felt "high class" and international. I was thrilled to be working in such an environment! When relatives from my hometown would come and visit me in Bangalore, I would take them for a tour of my office. These folks had never been to a big city or seen high rises before except perhaps in foreign movies. Stepping into my ultra modern and stylish office they would be astounded, admiringly commenting, "you must be some big, important man doing some important job". Though most of them had no idea what it meant to be a software employee, my workplace symbolized my high status and position in their eyes. This was the symbolic power of software workplaces in its early days.*

Currently Bangalore has one of the highest concentrations of software workplaces, gaining it the moniker "India's Silicon Valley". Bangalore's landscape is dotted with software workplaces. Modeled after the developed west and "world class" in appearance, software company buildings often stand out against the rest of local landscape (Radhakrishnan 2011). Drawn into Bangalore by software career opportunities, many employees are recent migrants to the city. Working in the industry accords these employees a sense of upward mobility and social status attainment.

In India's state-owned "old economy", Bangalore was a "hub" of technology and engineering institutions. The region had a high concentration of engineering and technology colleges. Several prestigious state run technology and engineering institutions

such as HAL (a major aeronautical engineering facility), ISRO (Indian equivalent of NASA) and BEL (state owned electronics engineering facility) were located in Bangalore (Parthasarathy 2000). With a relatively well-developed electronics and communication infrastructure, the city became a favored location for India's software industry.

To attract software corporations, the government set up "state of art" software technology parks (STP)s- massive gated complexes equipped with the latest internet and communication infrastructure. Software exporters setting up their units in STPs benefited from subsidized land and several kinds of tax subsidies.

In "Privileged Precariat of India's Software Industry", I show the nature of careers and jobs that this professional class holds in the software industry. Bangalore is the setting for this research. How are careers organized in the software industry? What are the ways in which work is performed in this sector? What discourses around jobs and labor surround the software industry? These are the main questions I answer.

**Theoretical Framework and Questions**

*New Economy, High Tech Knowledge Sector*

The software industry became a major provider of middle class jobs in India once the state sector retreated with 1990s economic restructuring reforms. In pre-restructuring India, jobs providing middle class mobility were mainly in state owned sectors (Fernandes 2006). They tended to be "cradle to grave" careers, where employees joined at the entry level and left upon retirement. Employees worked with a single company through their lives, moving up the career ladder gradually with predictable seniority-based promotions. Exemplifying job security and career stability (Deshpande 2003,

Fernandes 2006), labor rights and guarantees in these jobs were enforced by state patronized labor unions and labor laws (Chibber 2003).

With relaxed labor laws, software industry employers operate with limited state regulation. Market forces determine employment processes in this privately owned sector. Labor unions/collective bargaining is prohibited, and employees are unable to exert collective power. As one of the most globally integrated sectors, it symbolizes the end of import substitution industrialization and the end of restrictions over foreign trade and investment. But it is also vulnerable to all forms of global economic upheavals and market volatilities such as global recessions. How have these economic features reshaped middle class software jobs?

Secondly, the industry is knowledge work based and high technology intensive. Using knowledge as raw material, software work is mind and intellectual labor intensive. All software industry workers are highly educated. Yet digital knowledge work is also easily amenable to many new forms of electronic control and surveillance at workplace (Aneesh 2006), enhancing managerial authoritarianism and supervisory control over workers (Prasad 1998). How does "knowledge work" shape software jobs?

"Technology dynamism" (Benner 2002), or uncertainty caused by frequent technological changes, is a unique characteristic of high technology sectors especially the software industry. Technological booms and busts are rapid and unpredictable. New innovations are diffused and adopted fairly quickly. Rapidly changing technologies fuel the industry and at the same time create continuous upheavals.

It is quite possible for a top tier corporation in Silicon Valley to go under within a few years, taking jobs and careers with them. For instance Benner (2002) found that in

the Silicon Valley within 15 years, the top 75 software corporations eased to exist- either shut down or merged with another company. The US Silicon Valley is characterized by rapidly changing work, new forms of networking and mobility, and contingent employment. How does "technology dynamism" affect Indian software jobs and careers?

*Vertically Disintegrated Globalized Production*

The software industry is export oriented i.e. production in the industry is meant for foreign markets. Literature on India's software industry describes it as an outcome of global offshoring i.e. globally dispersed, vertically disintegrated production process (Hanson, Mataloni and Slaughter 2001).

Global offshoring process began with the manufacturing sector in the 1980s (Frobel, Heinrichs and Kreye 1981), especially in textiles and garments (Collins 2003), electronics hardware (McKay 2006) and extended to knowledge sectors such as call centers (Nadeem 2011) and back end business processing/data entry (Freeman 2000). India is a major recipient of knowledge sector-outsourcing jobs (called Information Technology Enabled Services or ITES). This cross border economic integration is facilitated by advances in communication and informational technologies i.e. Internet and wireless technologies and advances in transportation (Eden 1991). India is the recipient of about 80 percent of all global software outsourcing processes; most of its clients are US based MNCs (Lakha 1994).

Why is India a favored destination for global software outsourcing? What kind of global software offshoring is done in India? What logics drive this process? Many offshoring processes involve MNCs in the developed world sending out parts of their labor-intensive production processes to cheaper locations in the poorer, developing

world. Getting their processes done in low wage locations saves these MNCs on labor costs. It is often the main motivation of offshoring processes. But there may be other motivations such as access to better resources, access to foreign markets, ease of business and operations (Hanson, Mataloni and Slaughter 2001). What forms of offshoring arrangements underlie India's software industry? What broad patterns of global production processes are sent out to India?

The global value chains framework (Gereffi 2005, 2008) is a useful theoretical framework to understand the global dispersion of production work. Arranged in a sequential hierarchy of its position in the production process, each link in the chain represents a part of the production process and a relative valuation. Conceptual processes lying at the higher end of the value chain tend to be highly valued. Postproduction processes such as maintenance occupy the lowest levels of the value chain and considered low value. The value chain framework assumes task fragmentation of the entire production i.e. production broken into modular components amenable to cross border transfer.

Where is India's software industry located on the global software value chain? Some research on Indian software industry describes it as concentrated at the lowest levels of the software value chain, performing monotonous, deskilled work (Prasad 1998). Upadhya (2008a, 2008b, 2010) has shown the modular and standardized nature of India's software work. But there is disagreement on the extent of India's software deskilling and modularization.

Many have argued that software work precludes sufficient "deskilling" (McBreen 2002), because of its technical and unpredictable nature (Eischen 2004; Ilavarasan 2008;

Parthsarathy 2009; O'Riain 2010). Others claim that India began with low-end work but there is a "movement up the value chain" as more development and research work is being outsourced to the country (Athreya 2006). What larger patterns of work define India's software industry structure? What power dynamics underlie relationships between Indian software workers and foreign MNCs that offshore their software work?

*Privilege and Opportunity Structure*

Projecting itself as neutral in terms of gender, caste, religion and linguistic affiliations, the software industry claims to represent competitive fairness and middle class values of upward mobility through hard work (Upadhaya 2011). It is surrounded by a powerful discourse of "merit" and individual achievement over nepotistic, clientelist networks of favoritism. (Jodhka and Newman 2007). It is consistent with studies in other parts of the developing world where MNCs operating in the developing world reduce the salience of ethnicity-based discrimination- in contrast to domestic companies entrenched in long standing discriminatory practices (Weisskopf 2010). Representing the deeper entry of capitalism, these New Economy sectors tend to displace non-capitalist structures, replacing them with competitiveness and discourses of meritocracy, eliminating overt discrimination.

Yet these sectors also tend to give premium to certain competencies, which favors upper classes and dominant ethnic groups (Weisskopf 2010). Surveys on the Indian software industry have found it to be upper caste and professional, upper middle class in composition (Upadhya and Vasavi 2006). Who are the people that get into software jobs? What are the social identities that tend to fill software jobs and why? Who are excluded?

In the Indian context access to cultural capital- competencies (educational, credentials, social skills) are tied to ascribed status such as caste and class. Cultural and social capital is necessary to enter the software industry. Cultural capital bestows upon its owner distinct advantages and benefits that can be transferred and transmitted across generations and other classes can be excluded from accessing it (Laureau 2011). In middle class life privilege is enacted through ownership of Bourdiesian cultural and social capital. Krishna (2014) shows that getting an engineering degree is out of reach for most Indians, precluding their participation in New Economy sectors. Clearly production of merit is highly unequal in New Economy sectors, merit being linked to cultural capital, eliminating marginalized groups. What forms of capital are needed to enter the software industry? How are they acquired? How is social privilege enacted in this acquisition process? How does social privilege, Bourdieusian social capital, cultural capital and habitus operate in software careers' opportunity structure?

Effects of New Economy on historical forms of inequality and identity-based discrimination such as unequal treatment, partiality and unfairness are unclear (Centeno 2010). But as developing nations liberalized their economies, increased concern with market logics of efficiency and competition has often entailed reduction in social justice policies[8] because they are found inconsistent with market logics (Weisskopf 2010).

With liberalization, India's state owned public sector has shrunk, taking with it caste based reservations (affirmative action) in employment, and reducing opportunities for India's historically-marginalized "dalit" or untouchable castes. Employment opportunities in India's New Economy are led by private capital, which tends to be

8. Especially positive discrimination or affirmative action policies to remedy ethnicity based historical discrimination.

staunchly opposed to affirmative action[9], citing "merit" as main grounds for opposition (Jodhka and Newman 2007).

What are the dynamics of caste based exclusion and discrimination in the software industry context? Studies of "dalit" students' employability show the salience of "desirable" family background[10] as a major barrier in their hiring (Jodhka and Newman 2007). Most dalit students fail their interviews because of their answers about their family backgrounds; poor, needy and "undesirable" are used as proxies for ascertaining their "dalit" backgrounds. Yet the same studies also show other overt forms of caste-based discrimination are limited. How is being a member of a "dalit" caste a barrier to entry into the Indian software industry?

*Flexible Labor, Employee Autonomy and Agency*

Software companies are freed from labor protection laws. They hire and fire freely; there are no permanent jobs in the industry. Labor unions, labor organizing and collective bargaining are prohibited. These characteristics of flexible labor markets enabled corporations to reduce labor costs and market risks, and to increase profitability (Kalleberg 2001, 2003). Globally, under flexible labor conditions workers are forced to work long hours, work overtime and take work home (Sharone 2004). Studies theorize labor flexibility in terms of numeric, wage and functional levels (Wallace and Junisbai 2003; Benner 2004).

---

9. Industry argument against affirmative action is typically that these policies hinder India's global competitiveness as marginalized groups perform poorly at the expense of better performers from the mainstream groups (Centeno 2010).

[10] Employers often associated applicant's credentials with the family background. Merit is formed in the family as are credentials and expressed attitudes, they are the source of soft skills and therefore good fits for the corporate environment. (Despande and Newman 2007; Jodhka and Newman 2007; Attwell and Thorat 2010).

How is labor flexibility operationalized in Indian software industry? What forms of labor flexibility exists? What labor rights are there in the industry? Studies on Indian software workers reveal tremendous job insecurity and uncertainty, fear of layoffs and powerlessness (Upadhya 2006; Illvarasan 2008). Workers are under continuous pressure to upgrade their skills and find newer avenues for growth (Upadhya 2008a, 2009). Effects of "technological dynamism" seem to be similar to the experiences in the Silicon Valley (Benner 2002), in the American software industry in general (Sharone 2004) and in Ireland (O'Riain 2002). What kinds of risks and insecurities do software employees face? How do employees deal with these insecurities?

But the Indian software industry promotes an image of a "professional" workplace that rewards individual achievement and talent (Upadhya 2009). Kunda (1992) showed engineering corporations exercise worker control based on individualism, individual achievement and interpersonal competitiveness. Florida (2004) found the "creative class" works autonomously, motivated by a passion to "create" something new and money seems to follow these individuals. Saxenian (1996, 2006) found technically skilled employees in high technology corporations are driven by "self actualization" goals and self-motivated to work hard, precluding needs for employer created controls.

But Prasad (1998) has shown work in Indian software industry involves very high levels of managerial authoritarianism. Aneesh (2006) has shown software employees routinely face a high degree of impersonal forms of monitoring, control and supervision enabled by new technologies. How is labor control operationalized in the software industry? Who are the supervisors and managers, who are the workers? How are rewards and incentives used to motivate employees and acquire their consent? How is upward

mobility in software careers constructed? What are the upward mobility pathways and patterns in the industry? What makes software careers desirable?

**Organization of Chapters**

In **"Chapter 1: India's Software Subcontracting Model: Low Skill Work, High Social Status",** I describe the software subcontracting model in India. I show that software-subcontracting work done in India is low skill, but that these jobs provide high salaries and opportunities for higher consumption. Thus subcontracting software jobs have gained high status in Indian society.

In **"Chapter 2: Privilege and Exclusion in Software Subcontracting Recruitment"** I show that hiring in the subcontracting software industry shows a distinct structural favoritism for the professional, affluent, well-networked class of India, and an exclusionary bias towards individuals from socially disadvantaged groups. Privilege plays an important role in securing a software job, for instance through exclusive private colleges and information networks on one hand, and through professional "habitus" cultivated at professional homes and English-medium grade schools on the other. This explains the over representation of professional class and upper caste in the software industry.

In acquiring the necessary cultural capital to make it into the industry, many in the upper class resort to un-meritocratic means and strategies. They use nepotism and bribery to get around the admission process. Private engineering colleges operating under lax state regulation enable this behavior. To fulfill the "business requirement" needs for engineering degree holders in the subcontracting sector, many private colleges churn out graduates with questionable skills and abilities. But they get away with providing

substandard engineering education because the subcontracting sector doesn't really need engineering skill holders.

The relatively lower skill, standardized work of software subcontracting allows major companies to train recruits who lack technical skills within a relatively short period of time and make them ready for project deployment. It is ironic that the subcontracting software industry has acquired a strong meritocratic and fairness image in India.

In **"Chapter 3: Privileged Precariat: Insecure Jobs, Flexible Workers and Precarious Work Culture"**, I show that software subcontracting sector jobs are precarious and insecure. Employees of software subcontracting sector have to live under extreme career uncertainty. Fear of job loss is chronic. Companies hire and fire employees based on the market demand for their technology skills. Employees have to continually keep their eyes on the technology market demand and try reskilling to maintain employability. At the same time their jobs are high pressure, demanding long work hours. It is very difficult to reskill under such conditions. Employees often leave their jobs to do that. Unpredictability in their lives is worsened by the control foreign clients exert over subcontracting workers.

Managerial authoritarianism in performance assessment based rewards makes their jobs precarious and promotes toxic competition. The low skill, monotonous nature of these jobs makes many employees feel disinterested. Despite the insecure jobs and poor work culture in subcontracting companies, most employees continue to work for the industry. They hope to save enough and invest in property. This way they hope to secure their future.

In **"Chapter 4: Mobility into Managerial Roles"** I show that upward mobility in subcontracting software jobs usually implies mobility into managerial roles. It is the next step in a subcontracting employee's career and professional growth. In corporate hierarchy, managers imply middle and senior management. Mobility into middle level management roles is through incremental promotions. But these mobility opportunities are very few and selectively awarded in practice. Mobility paths are not standardized. Ability to move into middle level management depends upon individualistic strategies. Mobility into senior management role is rare. Very few subcontracting employees are able to move up into managerial roles.

In **"Chapter 5: Mobility into High Value, High Skill Software Work",** I show that mobility opportunities into the high value software work is highly selective. Mobility is constrained by the high skill, specialized "niche" technology abilities needed to work in this sector. The high skill software sector in India produces high value, cutting edge products for developed world markets, led by MNC subsidiaries that collaborate with India-based high skill software engineers and developers in developing these products. Their employees work in specialized, niche technology areas. They need to show high innovation ability, autonomy and creativity at work. Their skills are in-depth, built over many years and they work hard to acquire skills that make them irreplaceable. These companies reward their employees highly. Managerial control over them is low. These are some of the best jobs in India's software industry.

Mobility into this area is generally restricted to the graduates of highly meritocratic elite engineering colleges like Indian Institute of Technology (IIT). Admission into these colleges is generally possible for the most privileged children of the

urban professional class of India. It is very difficult for software subcontracting employees to move into this high value, niche area of software development.

**Methodology**

The research is based on life histories of Indian software employees based in Bangalore. Using purposive snowball sampling, I interviewed 100 respondents for this research. A total of 80 respondents work in India's software companies as regular employees and managers in these companies. The remaining 20 respondents are closely associated with Indian software industry. I conducted these interviews in 2015.

For purposes of my research, software "employee" is anyone employed in the Indian software industry on the technical side: engineers, developers and programmers. They constitute the vast majority of the professionals employed in the software industry. I use the term "employee" and "worker" interchangeably for them.

A much smaller number of the respondents are "managers". I use the term "managers for respondents that do not perform technical work but supervise technical teams and software projects. But there are some managers that are members of sales teams that bring in business and clients. The remaining 20 respondents are either aspirants seeking to enter the industry, members of Human Resources (HR) in software companies and members of software industry training, recruitment and placement agencies.

A majority of the respondents (about 70 of them) worked in the subcontracting sector. Of these 70 respondents most were either at entry level or junior management. A few were freshly recruited campus hires, undergoing their technical training at the time of

interview. A quarter of these subcontracting sector respondents were in managerial roles. About 10 respondents worked in the high value software sector.

In addition to the 80 fully employed respondents, I interviewed around 10 respondents associated with the Software Training Program (STP) run by Infosys Foundation. STP training is meant to make engineering graduates from historically marginalized backgrounds job ready for the software industry. These respondents were "trainee" engineering graduates aspiring to enter the software industry or their trainers and coordinators in the STP program. I interviewed another five respondents who are members and trainees of NGO Free Software Movement Karnataka (FSMK). The NGO provides free software training to youth from marginalized backgrounds.

Primary data used in this research is based on narratives of respondents' perceptions, attitudes and opinions. Through a set of detailed, open-ended questions I captured their work and employment histories, job, education and skill/training histories. The average interview was three hours long and conducted over multiple sessions.

Many respondents were not able to accommodate my requests for interviews in-person because of busy schedules. They preferred to be interviewed on phone. Most respondents declined to be interviewed on company premises. A majority of the telephonic interviews were conducted after office hours, usually late evening or nighttime. A few telephonic interviews were conducted before office hours. In person interviews were conducted at respondents' homes, office cafeteria and restaurants- in venues decided by the respondents.

Respondents are drawn out of my personal networks of software employees' institutions, friends and family. An important source for respondents was the software

employees' forum Information Technology Employees Center (ITEC) based in Bangalore, that put me in touch with their members for interviews. Using these contacts I was able to expand on the number of respondents. Other respondents were drawn from personal alumni networks, family and friends. Using these points of contact I expanded my respondent base through snowball sampling.

Though snowball sampling is not representative as random sampling, and can be biased, it has its advantages. Randomly sampling Indian software corporations and getting primary data from them is challenging. It is impossible to directly walk into most software companies in India and meet employees or HR managers on site because company premises are strictly off limits to outsiders. Strict security checks, electronic surveillance at building entrances prohibit outsider access. For outsiders to get inside company premises, a company employee needs to escort the visitor in and out of the heavily guarded gate, obtain a visitor badge and accompany the visitor while on premises at all times.

Randomly contacting software corporations (usually the HR managers) through emails proved to be unfruitful. They did not respond to emails or requests for interviews. Software corporations face a lot of poaching of workers from rival companies. Outsiders are viewed with suspicion and are not welcome. Software employees in India are bound by confidentiality clauses and formally prevented from talking about their work to outsiders. This makes it harder to approach software employees randomly.

This kind of snowball method is common to studies of the software sector. Upadhya and Vasavi (2006) conducted a survey of software companies and employees in Bangalore and facing a problem of access with random sampling, and not getting enough

interviews, they finally successfully used the snowball method to move forward. McKay (2006) echoes a similar experience in his research of electronic hardware workers in Philippines, where he was finally able to move forward through personal contacts in companies rather than through random sampling.

*Limitations*

This research focuses on workers' opinions, attitudes, strategies and lived experiences. Unlike experimental methods where the effects of the variable under study can be isolated and causality established with greater accuracy, perception based research can only provide insights and patterns through lived experiences. I did not cross check or triangulate respondents' perceptions with other stakeholders to check for accuracy. Since perceptions can be one sided and flawed (Pager 2010), it may limit the data's internal validity.

I referred to secondary data whenever possible to cross check my findings. A key secondary reference is "Final Report on the Work, Culture and Sociality in the Indian Software Industry" by Carol Upadhaya and AR Vasavi. It is an empirical data report of software work across 200 companies covering 600 software employees. Technology section articles in India's leading newspapers such as "Economic Times", "Times of India", reports of the Indian software industry lobby-NASSCOM, white papers of prominent Indian management institutions, company websites and reports from software industry consulting companies and newsletters of the Indian software workers' forum – ITEC were other secondary data sources.

# CHAPTER 1

## India's Software Subcontracting Model: Low Skill Work, High Social Status

*Two decades back when the Indian software industry was taking off, many Indian software programmers made prayers for successful US travel at a local temple in the Indian state of Andhra Pradesh. The high rate of success in getting US visas and travel opportunities upon praying in this temple gave it the name "Visa Bajaji Temple". Typically, worshippers perform a wish-making ceremony that includes making 11 laps around the inner temple. Sometimes they proffer their passports and make offerings of fresh coconuts. If the wish comes true, they must return and make a further 108 laps*[1]*. Over the years the temple's reputation has grown by word of mouth and now attracts crowds who line up to make the rounds. Its popularity is helped by pamphlets handed out at the temple that contain testimonies of success. In another Sikh temple in New Delhi*[2]*, US travel hopefuls make offerings of airplanes to the presiding deity. There are three other visa temples in India where devotees pray and make offerings so they can travel to US and other developed country destinations and earn large dollar amounts. Many of these aspirant devotees are employees of Indian software industry hoping for "onsite travel" to foreign client locations.*

The Indian software industry has opened doors to global mobility for many Indians. Indian software subcontractors send their employees on fixed -term "onsite" deputations to client locations. Most of these clients are US based followed by other

---

[1] CNN (2016), "Indians Flock to Temple in Hyderabad Believed to Help in Getting U.S. Visas" http://www.cnn.com/travel/article/india-temple-visas-hindu/index.html

[2] "Got A Visa Problem? Try These Four Temples". Economic Times, March 6 2017. https://economictimes.indiatimes.com/nation-world/got-a-visa-problem-try-these-four-temples/trouble-getting-visa/slideshow/57491869.cms

developed regions like Australia, UK, Canada and other western European countries. The main draw for these onsite deputations are the dollar amount salaries paid to Indian employees.

*Currently Indian software employees on deputation to the US are paid between USD 3500 to USD 4000 per month. Though modest by US standards, these salaries are considered extremely high in India. Many software employees are able to buy fully paid off first or second homes upon returning US onsite placements. Fully paying of such homes would take them several decades normally.*

Average salaries in these software jobs are relatively high by Indian standards. Most employees are engineering degree holders. Working in the software industry is considered high social status.

**Genesis of Software Subcontracting Sector**

*It was the global Y2K bug issue, the conversion to Euro and a few other large scale needs for low level programmers that kick started the Indian software industry. The Y2K bug affected all computerized systems globally. Though the fix itself was a relatively simple process it was nevertheless very crucial and needed a large army of people with basic programming ability and basic software skills. Also because almost 80 percent of software products are still developed in English and then localized, English skills were also necessary.*

At that time India happened to have both these skills and a large army of trained manpower to provide them at a fraction of developed country wages (Parthasarathy 2000). This is where India's advantage lay over other countries. Thousands of Indian software programmers traveled to the US on L1 visas, meant for fixed term work for US

based clients. These programmers worked on low-end software support and maintenance work to developed country locations on fixed term projects.

Technology limitations at that time constrained the amount of software work that could be sent out to India and done from Indian shores. Therefore Indian programmers had to physically travel to client locations. This process of "renting out" programmers to foreign client locations lies at the roots of Indian software industry (Arora 2004). It is known as the "body shopping" model (Biao 2007).

*Body shops hired Indian workers for fixed term projects, took care of their travel and visa formalities and sent them to overseas, foreign client locations. Typical body shopping projects were done at foreign client locations rather than in India. These body shops are essentially labor contract agencies that supplied Indian labor with software skills on demand.*

Later technological advancements: developments in telecommunications, Internet and mobile telephony allowed for remote working. Meaning this work could be done out to India. Indian programmers could directly work out of Indian shores i.e. they would not need to travel to overseas client locations. This led to a decisive "shift" in the industry:

*With the development in information and communication technologies entire software projects could be sent out to India. Labor contractors could hire India based workers for subcontracting projects. These developments gave a major boost to India based software subcontracting. Companies working on foreign subcontracting projects mushroomed overnight. Most of these subcontracting companies began small, operating out of apartments and one room setup. Low operating costs enabled the proliferation.*

These types of small companies dominated the market in the 2000s as the industry

was starting to take off in India, evident in writings of (Lakha 1994, Prasad 1998), including bodyshops[3] (Biao 2007). They provided low skill software subcontracting work to foreign MNCs.

**Growth of Major Software Subcontractors**

Over time, Indian companies developed the ability to manage bigger projects. Individual subcontracting projects became much larger in size and value. As the software-subcontracting model started to mature, bigger volumes of software subcontracting work began relocating to India. This led to higher profitability. Higher profits allowed some of these small companies to eventually scale up.

*In the early days of its inception in the 1980s, subcontracting giant Infosys started very small, as a one-room company and handful employees. It began with a meager capital of $250 that founder N Narayana Murthy famously borrowed from his family. Now it's grown into a giant company with $11 billion turnover[4], with over 100,000 employees.*

Many present day giant software subcontractors originated small like Infosys, later scaling up to their current size. Major software companies Wipro, TCS, IBM-GBO, Accenture, Infosys etc. are similarly large sized with large-scale operations.

*TCS, India's largest mass recruiter has a headcount of over 300,000 employees. Infosys, Wipro, Accenture, IBM and some other major mass recruiters have over 100,000*

[3] In the early days the software industry began with a subcontracting arrangement where labor contractors supplied Indian software labor to foreign locations. These body shops are therefore labor contractors that hire Indian workers for fixed term projects from foreign clients and take care of their travel and visa formalities and sent them to overseas, foreign client locations. So typical body shopping projects were done at foreign client locations rather than in India.

[4] "Start-up Stories: NR Narayana Murthy, Infosys". April 4 2011. BBC News. https://www.bbc.com/news/business-12957104

*employees each. So these companies are the largest employers in Indian software subcontracting.*

The software subcontracting market is "project" driven. Currently Indian software subcontracting sector receives about 80 percent of all global software outsourcing arrangements, in the form of "projects".

*A "project" is a specific set of peripheral, routine software tasks within a larger production process, subcontracted out by foreign MNCs "clients". Most major mass recruiters have hundreds of such "projects".*

Major subcontracting companies dominate the market. They have thousands of projects from major MNCs, foreign government departments and educational institutions. Approximately 70% to 80% of their work is export project driven, catering to developed country markets.

Main clientele of Indian software subcontracting are the global banking, finance and insurance sectors. Major banks, insurance and finance MNCs subcontract their software operations to India. These three sectors (Banking, Finance and Insurance) are combined together to form the BFSI segment in software subcontracting business. The global BFSI sector is the biggest segment of Indian software subcontracting exports. It contributes about 40 percent of all software subcontracting work to India. Retail, manufacturing and infrastructure sectors are other important areas of Indian software subcontracting.

*For client corporations, subcontracting to India is a major cost saving exercise. In the initial days an Indian worker was paid 1/10 of the US salary. Given the labor-intensive nature of software work, it translated into major cost savings.*

Labor cost savings is the primary basis for Indian software subcontracting. The largest volume of software subcontracting is meant to serve US based clients. US based companies benefit the most out of Indian software subcontracting. In addition to low labor costs, subcontracting to India reduces US clients' downtime because of the 12-hour time difference between the two countries.

*For US based clients, moving system maintenance work (a major component of Indian software subcontracting) to India involves another layer of cost savings because of the peculiar spatial-temporal differences: India and the US have a 12-hour time difference. As bugs, issues and errors in their systems (system maintenance work) get fixed at US downtime hours; US client employees can resume normally without extra downtime during work hours. This way the US client's workday is stretched.*

**Low End Subcontracting Model**

*Clients pay subcontracting agencies on a "per project" basis. Each project is billed and paid for differently. Typically agencies bill foreign clients for the total number of hours needed to complete the project times number of workers needed to complete it. The rate at which the client pays for each worker-hour is market determined and fixed in advance. But the rate at which the billing happens is generally a fixed, standardized amount for the entire industry. Generally these subcontracting agencies bill their clients at a higher rate per worker-hour and subsequently pay each worker a much lower amount than the one billed. The difference is profit.*

Indian companies charge higher prices for the labor while paying labor lower wages. Subcontractors' surplus depends upon "labor arbitrage" i.e. buying labor at a lower price than selling it at a higher price. They are incentivized to hire fewer workers

and make bigger profits by doing so. Indian software subcontractors continually try to reduce costs by cutting the number of workers deployed in subcontracting projects.

Initial contracts that body shops and small subcontracting companies got were for data processing (Alamgir 2005) and fixing the Y2K bugs and Euro conversion issues (Parthasarathy 2000). Some also got maintenance and support projects as well.

These projects involve tasks that are often tedious and labor intensive. Jobs involve maintenance, looking for bugs, troubleshooting, upkeep or adding some modification or feature to existing software systems for big clients, typically based in the US. Another major piece of work involves server maintenance for major MNCs. Majority of Indian subcontracting projects are in these areas. These jobs are considered lowest skill in the global software industry. They are some of the lowest level tasks in the software value chain. Much of the Indian subcontracting work remains concentrated in these areas.

However as the subcontracting sector matured, the industry diversified in the type, size and technical complexity of subcontracting projects. An area in which the subcontracting sector has moved up from the lowest level software skills is custom application development.

*In custom application development" Indian subcontractors design software applications on demand. These software applications are customized to address specific business needs and demands of a particular foreign client that owns the project. These projects generally need much more coding expertise than say low-end purely software support, maintenance and testing projects that most of the industry is still reliant on.*

Custom application development is still considered relatively low skill, low end on the software value chain. Though it is more skill intensive than purely support,

maintenance and testing jobs that Indian subcontracting companies normally perform. Projects that are higher in skill tend to be valued higher as well, implying higher rates of payment. Clients pay these projects at higher rates. Therefore profits in higher value subcontracting projects are higher.

Indian subcontracting companies are trying to diversify into more valuable projects and scale up their operations on the value chain. But projects involving custom application development are still a very small proportion of the software subcontracting market in India. Thus majority of Indian software subcontracting specializes low end, monotonous software jobs.

**Social Status as a Motivation for Software Employment**

*Fresh entry-level salaries in the major software companies are Rupees 350,000/annum (USD 7000/annum). Though pay has stagnated since 2008-10, this pay is still comparable or even higher than many other sectors popular for educated middle class employment in India.*

Relatively higher pay scales continue to be the main draw for software jobs. So educated English speaking youth with software programming skills would come work for them, Indian subcontracting companies began paying relatively large salaries for software jobs. This sort of cultural capital is necessary to work with foreign clients.

*Parag recalls getting campus hired into a small subcontracting company in 2000. "When this company came for campus hiring, everyone one of us wanted to get selected. Given our modest software programming skills, it was offering way higher salaries than we were accustomed to at that time. It was a small company with 50 employees; working for a US based telecom MNC. We had to provide technical support to the client 24/7-*

*troubleshoot technical issues and snags in their systems and solve them proactively. It needed excellent English communication skills and being "there for our" client at all times. The work culture was crushing. I remember working 16 hours a day and most weekends. They had warned us about that at the time of recruitment, so I was prepared. But it didn't matter so much at that time because within two years of working there I was able to buy a car and book a house on loan. For me it was unthinkable at that time. Many of my friends and classmates got hired into similar companies with large salaries. The software industry was starting to boom just then and the salaries being paid were astonishingly high".*

Higher pay implies ticket to middle class lifestyle and social status mobility. Many respondents said that their software job has enabled higher consumption and affordability. It is the main draw for joining and continuing to work in the industry despite the crushing work culture. Software work is one of the most popular employment choices for educated, English speaking Indians.

*Consumerism and Status Mobility*

*Ram has been working for the software industry for several years. With the software career he feels he has moved up because he can spend on daily consumption items spontaneously instead of preplanning. " Though I grew up in a family with middle class lifestyle and my father had a government job, money was tight. We needed to pre plan before taking a cab for buying new clothes. But after I joined the software industry my consumption levels have definitely increased. Buying expensive branded clothes from high-end shopping malls is a regular monthly feature. And I take a cab almost every day while commuting to work. These are my regular expenses now. Now I consider them*

*necessities but growing up they were difficult to afford on a regular basis. We considered them "luxuries" that needed family pre planning".*

Several other respondents similarly mentioned that their software job has enabled higher levels of consumption. Consumer expenditures once considered luxuries occasionally affordable have become regular, daily feature or necessities in their lives.

*For many other respondents, regular eating out in expensive places and trying out new cuisines on a regular basis was an important marker in their upward consumer mobility. For most of them, meetings with co-workers and within teams may happen at an expensive restaurant or coffee shop. Socializing and hanging out with friends happens at pricy venues: nightclubs, five star hotels and shopping events. These were rare events in their lives while growing up because they were difficult to afford. But now this consumer culture is a normal part of their life, made possible by high software salaries.*

Many software professionals I interviewed echoed a similar sentiment. For them the software career has entailed a significant shift in consumption. They framed mobility in terms of enhanced consumer power because of their software salaries.

*Mita framed her social mobility in terms of more expensive consumption aspirations like international vacations. "I grew up in a middle class family and my father, a state employee could take us- a family of four for annual vacations within India. While growing up, international travel was an impossible unattainable dream, out of my family' s reach and others in my peer group. But many of my peer group of friends, all of us in the software industry went to Europe or US for our honeymoons and other special events. This sort of affordability was unthinkable before but now possible because of the*

*software industry salaries. Because of the software industry salaries I feel many middle class Indians can afford international travel now".*

But more expensive forms of consumption such as high end cars, gadgets and fancy homes are now affordable after some years of "onsite" deputations at foreign client locations. Onsite deputations are a major attraction amongst subcontracting employees.

*According to a respondent, Naresh, "The pay I get right now is not enough for me to build a house. My only aspiration right now is to make a house for my parents. They never had a home of their own. I come from a poor family. I don't spend on any conspicuous consumption so I can save for this home. But it's impossible without an onsite opportunity. Software salaries have stagnated since the last recession".*

More expensive consumer items like homes are harder to afford given software salaries have stagnated since 2008-10 global recession. At the time of interview some respondents were waiting for an onsite deputation so they could buy their house. Other respondents informed that they bought their first home after a long-term onsite travel. A few respondents confessed that they got multiple onsite opportunities and bought multiple homes through onsite salaries.

*Neha informed me that that there is a standing joke in the Indian software industry: every onsite assignment means one more house. So the moment someone gets an onsite assignment they are teased about where do they want to buy their next house once they come back to India.*

For some others, onsite opportunities allowed them to pay off debts and gain more financial freedom. Respondents opined that the hope for onsite deputations often

motivates many software professionals to put up with poor working conditions and an exploitative work culture.

*Adit admitted that he put up with his company's "horrible", "exploitative" work culture just to get an onsite position. " I ended up working 14 hours a day just so I could get noticed and get an onsite position. I would keep dreaming about work and the next thing that I needed to do. Hoping they would be impressed with my work and give me the onsite opportunity I so desired. It affected my health. At night I couldn't sleep because of the stress and anxiety palpitations. But I hung on. Eventually after three years I got a two-year opportunity to work at an Australian client location. I saved as much of my salary as I could. Once I came back, I was able to pay back my education loan. I also paid back my parents' home loan with my onsite salary and I was able to buy a car with the remaining savings. Ironically my father worked all his life as a government employee and bought this house when he was about to retire. He couldn't afford to buy a car. But I can afford these things in less than 10 years of work in the software industry".*

Many software employees enjoy high consumer standards, according high social status to software jobs. Software employees are often associated with high incomes, flashy lifestyles and conspicuous consumption previously unseen in India (Fuller 2007; Upadhya 2008b, 2011). The software industry took off around the time India was liberalizing imports. Until then the Indian market was closed to foreign imports. This severely restricted consumer availability. As foreign consumer products flooded the Indian market and the software industry boosted affordability, consumerism started booming in India.

Thus "New Middle Class" of India, associated with India's "New Economy" under market liberalization reforms is considered highly "consumerist", "flashy" "self serving" and so on (Deshpande 2003, Liechty 2003; Fernandes 2006; Varma 2007; Ganguly-Scrase and Scrase 2008; Brosius 2012). For most respondents, high consumption because of higher incomes was the main draw to continue working in the software industry.

*Fear of "Falling in Status"*

India's software industry emerged as state owned sector jobs were starting to recede under economic restructuring processes. State sector jobs were the main source of middle class mobility for most Indians. With limited prospects in the state sector, software subcontracting was often the only "viable" employment choice.

*Right after his engineering degree Jeet joined a major software company in Bangalore. " I graduated in chemical engineering. When I went to college I had no idea or desire to join the software industry. I wanted to get into a chemical company and use my degree and training there. During campus job placement season, I waited for some chemical companies to show up. But it didn't happen. Some software subcontracting companies came for campus placements. I sat for one and got selected. They were offering pretty high starting pay. It was a difficult choice to make but I took it. Meanwhile my father was retiring his government job the next year. I needed to take over the earning responsibility for my family. That was the main reason I got into this line of work.*

Several other respondents joined the software industry because the main earning member (father) was retiring from a government job and they under pressure to fill in as the earning member role. Having trained in other branches of engineering, often they had

no interest or inclination for software work. They wanted work in their degree specialization or "core" industries. But with the retreating state sector, such opportunities became scarce.

Fear of falling in status was an important motivator for these respondents to join the software industry. They came from "old middle class" homes, their parents worked for the state owned sector. They were accustomed to the middle class lifestyle and status. They didn't want to lose that.

*Maintaining Social "Status Quo"*

*Anita felt that her software job "helps her keep with" with the status she grew up. As a child of a dual income family with well-paid state sector jobs, she grew up in significant affluence and consumerism. "At the software subcontracting entry level my income is not as high as my parents' pay was at the time of their retirement. But it helps me keep up. I get regular financial help from them so I can have higher consumption standards. My parents are investing in a new house and car for me, that they plan to gift me soon".*

Coming from dual income families with affluent, state sector employee parents, many other respondents didn't view their software jobs as "mobility". Accustomed to high status, their software job merely allows them to hold on to their social "status" i.e. maintain "status quo" in society.

*"Sumit graduated mechanical engineering and joined the software industry through college campus placements. "Normally in my engineering college, state owned companies came for recruitment in many core-engineering areas. Many engineers would find jobs that way. My father trained in engineering also from my college and joined a*

*government job. Many other relatives did the same. I trained in mechanical engineering and hoped to find a similar job. But that year for campus placements, not a single state owned company or mechanical core company came. While software subcontractors were hiring in large numbers. I heard about their awful work culture and complete lack of job security. Reluctantly I decided to join one. It's better to have some job rather than no job. Besides my father was scheduled to retire soon. So I didn't have much of a choice or luxury to wait for the right job ".*

Until early 1990s the state provided secure, well-paid, middle class jobs to many educated Indians. With economic restructuring, government jobs became scarce. Though employment majorly expanded in the private sector, middle class oriented jobs are often limited to few privately owned sectors such as the software industry. Left with few other middle class mobility options, many educated members of India's "New Middle Class", offspring of state sector employees joined the software industry.

# CHAPTER 2

## Privilege and Exclusion in Software Subcontracting Recruitment

*I met Dileep a software job seeker who moved to Bangalore from a rural working class dalit[1] family. It was already several months since he graduated engineering and was still looking for a job. He came from a rural area in a neighboring state. His father was a chauffer in a rich landowner's house in the village. It was his father's dream to make him an engineer so he would have a bright future. His father couldn't finish high school when he was suddenly orphaned and he had to support the family. Working in the landlord's house, Dileep's father was impressed by the good lifestyle and incomes the landlord's sons had. He found out that they were all engineers and therefore earned good salaries. Associating engineering with good jobs and a respectable lifestyle, his father's greatest aspiration was to make his son an engineer. Dileep went to the best English medium school, the best one that his father could afford. He was one of the top students in his school. Later his father admitted him to a private engineering college near his hometown that was affordable. Dileep performed well as a diligent student. He is a first generation college student in his family and they had no idea about the importance of college rankings in hiring. Unfortunately his father didn't know that the college he selected was a very low tier college, with no campus placement process and poor curriculum. His father took out huge loans to pay for his education with the hope that Dileep will get a good job after graduation. So when he graduated, there was no job for him. He lacked the skills to enter the industry and had no idea how to get them so he moved to Bangalore hoping to find something. He was terrified of the loans that his*

[1] Dalits are the historically marginalized, so called "untouchable" castes of India.

*father has to repay and fearful of facing his family with the string of failures in the job market.*

Every year, during job placement season, subcontracting companies visit major college and university campuses across India to hire fresh undergraduates. Most of the hiring in major software subcontracting companies takes place at the entry level through college campus placements. Most of their hiring happens in India's engineering colleges. But these campus placements are not open to all. Companies are very selective in visiting college campuses for recruitment:

"*We conduct our recruitment on engineering college campuses across India. Most of our campus visits are limited to a list of 25- 30 engineering colleges. These are the colleges from where a majority of our entry-level hires take place. And I believe our rival companies also follow a similar policy. So direct college recruitments from these colleges account for a bulk of our software industry hiring", according to an HR manager at a major mass recruiter.*

Several HR managers (from major software corporations) mentioned that they campus hire from a limited number of engineering colleges. Many of these colleges are privately owned and extremely expensive. According to Bhriti, private engineering colleges in India can be ranked into three categories based on their placement record:

*"Colleges with more than 80 percent placements are the highest category and most reputed. Some of these higher-ranking private colleges are very expensive and therefore open only to the rich and upper middle class. But they provide almost guaranteed jobs upon graduation. Then there are those with 40-50 percent of higher placement are the 2$^{nd}$ category of repute. Lowest tier colleges have almost no campus*

*placements. So these students are completely reliant on open recruitment drives where any one can walk in for selection test to get hired".*

Poor families may go into major debts to send their offspring for an engineering education under the belief that their offspring will get a good job in the software industry. But these disadvantaged families may not be aware that an engineering degree is not an entry ticket to a software career. Unless the college has good campus placement records, it is not the right engineering college. With the engineering degree they may fail to get their "foot into the door" making their degree worthless. Such colleges are not worth their investment.

But many families may lack this information beforehand. Systematic data on private college campus placements is not available. It is challenging for families to know beforehand whether a college worthy of their investment. There are instances where colleges might make false claims about job placements. Disadvantaged families may lack informational networks to verify these claims or get the right guidance to make informed choices. They may end up like Dileep who got the coveted engineering degree but failed to get a software job.

*Anu's parents are professionals and she grew up in an upper middle class neighborhood in major Indian city. Despite her good private school education she was only an average student couldn't clear any of the entrance exams in the coveted engineering colleges. Systematic college rankings exist for only these higher tier colleges but she couldn't make it to any one of them. Clearly she had to go to an expensive private college but there are thousands of engineering colleges, which one to choose? Her parents with an active social life and networks got in touch with one of their neighbors*

*who was a board member in a private college in another city. He assured her parents about the facilities, the infrastructure and most importantly an honest assessment of campus placements. Using his recommendation she was able to get into the college. In the campus placements she got hired in a major software company after graduation.*

Well-connected families access their social networks to choose the right colleges. These networks as purveyors of information allow informed choices like how to assess the good colleges from the worthless ones, reliable information on campus placements and relative merits and prospects of different colleges and so on. Entering the right college ensures their offspring will get software placement upon graduation i.e. their education investment would pay off.

**Class Privilege and Engineering Colleges**

Private colleges are able to provide assured placements because they have close connections and networks in the software industry. They have dedicated personnel paid to build industry partnerships and take care of hiring relationships. Many others hire senior management from the software industry as visiting and guest faculty. These faculty members are paid very handsomely for bringing in their industry networks and knowledge of internal hiring dynamics into these colleges. Many other colleges are built through industry partnerships. A good example of such an elite industry partnership private college is IIITB College in Bangalore:

*The college is owned by Infosys, a major software subcontracting company and very expensive. Many faculty members here come from the software industry. Some are former senior executives with Infosys and have years of experience. They bring in their years of experience and industry networks, successfully preparing students to meet the*

*industry needs. These faculty members play an instrumental role in ensuring 100 percent placements. Students have the opportunity to closely interact with such industry leaders as a part of the curriculum. The college also gets regular guest lecture visits from software industry CEOs and similar highly placed corporate executives.*

Dedicated personnel in these expensive private ensure they build and sustain close relationships with HRs in major software corporations. A recruiter of more than 10 years in a major software corporation explained how she decides which engineering colleges to recruit from:

*"Generally this decision depends on several factors: At the basic level we make sure these colleges are accredited either by AICTE or NAAC, the main state accrediting bodies. And of course we are aware of the course work, structure of education, ranking in private magazines for the college. But we also rely on colleges to get in touch with us, make representations to the company about their college and their curriculum and its rigor. This is how we get to know of a college and whether it is worth visiting. Then we make our own observations about these colleges. For instance if we have hired from a college previously, we keep track of how good these hires were. If the previous hires were satisfactory we go back again to hire from those colleges. We also talk to our managerial networks in rival companies to get references for colleges. Generally we do not conduct campus placements in colleges without good references or recruitment precedents".*

To enable campus placements groups of expensive private colleges band together and form exclusive recruitment partnerships with software corporations. To make sure these corporations come for on-campus hiring, these colleges band together and invite software corporations. They ensure corporations have a large crop of students to choose

from. And in these partnerships colleges and recruiters negotiate what kinds of recruitment slots are allocated to respective corporations. Sometimes they pay companies to come and recruit. These efforts ensure their students get hired.

*According to Amar, expensive colleges like SRM group in Chennai have close to 100 percent placements because of close ties with industry. Getting into these colleges is not so difficult as competition is limited. Only those who can afford the high fees apply. Incidentally Amar is from a privileged dual income family of professionals. He admitted had he opted for this college, it would have been a financial burden for his family.*

Graduating from these exclusive private colleges often guarantees a software job at a prohibitive cost. These colleges justify the high fees based on high campus placement records. Most of these placements are in the software industry. But these exclusive private colleges with 100 percent placement records are unaffordable for most Indians.

Graduates from low tier colleges have a much harder time getting into the well-established companies because their colleges lack campus placements. Many rural candidates from poorer families choose these low tier colleges because they tend to be relatively more affordable than the exclusive private colleges with assured industry placements. To get a software job, such students are completely "on their own".

*Girish studied in a rural college with no campus placements. It was affordable for his family but he spoke of the troubles in getting recruited because his college didn't have campus placement system. " In my college students were on their own to find jobs after graduation. In such colleges students go for the walk in recruitment drives that many of the big companies conduct for entry-level jobs. But these are always located in big cities. While my college was in a far flung rural area. Companies don't bother to send word*

*(posters or flyers) to these far-flung colleges. Colleges like mine generally don't receive information about these drives. I was lucky I saw one poster of a major software subcontractor in my college about a recruitment drive. It was happening in a city several hours away. But I was able to make it there on time. And eventually managed to get a job".*

Low tier college students may gain entry through "open recruitment drives". Here they can directly walk in for entrance test and interviews. But entry-level open recruitment drives are very few because they are not convenient for corporations.

*According to an HR manager, "In hiring processes we believe in relying on past experience more than anything else. If we have hired many students from some colleges we would prefer to go to that college again rather than trying out a new one because we cannot be sure of getting a good number of recruits elsewhere. Recruitment takes a lot of time and corporate resources. And trying out new, unknown colleges is risky and a potential waste of our resources. This is the main problem with open recruitment drives as well. We are just not sure of what we will get in return for our investments. As HRs we want to make sure we get the best value for all the resources we invest in the process. So we do not prefer open recruitment drives either".*

Unlike students from low tier colleges, those attending exclusive private colleges have a smooth recruitment experience. Therefore getting hired into the coveted, well-paid software subcontracting jobs is a matter of "affordability and "affluence".

**"Back Door" Entryways**

*Lalita mentioned that she was a poor student all her life. She got a very low rank in the engineering entrance. Therefore based on merit she couldn't make it into the*

*engineering college that her parents chose for her. But her parents knew the management in that college. And they paid a large sum as a "donation" to get into a management seat. The large donation amounted to an additional 1/3rd of the yearly college fees. Her father is a banker and her mother a professor. There were more respondents like Lalita that were not good students and couldn't qualify to join the college of their choice. So they went via the "management quota". After her graduation she was able to get a campus placement in a subcontracting company.*

Many private colleges openly follow a policy of accommodating the rich and admission through "management quota" is one of the chief accommodations. Under management quota, private colleges reserve a certain percentage of seats for students that would pay a large capitation "fee" to get into the college. This large capitation fee is called "management seat fee" or "donation". Clearly it is legalized bribery where students can pay their way into college when they cannot clear the competitive entrance test. "Management seats" or "donation seats" are common in most private engineering colleges in India. But they defy all standards of merit and meritocracy.

*When Sumedh appeared for his final high school exams, he failed to obtain passing grade in the basic 12th-grade math exam. So he failed to clear his high school in the first attempt. He wanted to get into an engineering college. But the minimum criterion to gain admission in any engineering entrance is passing marks in all subjects in the 12th grade exams. This means he was ineligible for engineering college admissions. Yet he was able to gain admission in an engineering college under management quota. A family friend that owned the college ensured entrance rules were relaxed in his favor. Sumedh is*

*from an upper middle class family in a major Indian city. His father is a senior level manager in a major public sector corporation.*

Private colleges may have standardized internal entrance tests and interviews or some other forms of standardized assessment for student selection but management has full powers to bend the rules whenever necessary. This regulatory autonomy serves the rich, granting them coveted "engineering degrees" without merit or ability.

**Engineers by "Degree" not in "Skill"**

Major software companies assume that the fresh engineering recruits do not have software or programming skills. They provide substantial technical training in a "crash course" style after recruitment, to make up for the lack of these skills.

*Training programs typically have 8 hours of classes, assignments and exams over several months. Companies like Infosys have dedicated campuses and faculty for this training. In Infosys, the training takes place in Infosys Mysore campus, a sprawling, fully residential facility set up for this purpose. Though all companies don't have dedicated fully residential facilities, they follow a similar crash course system where large groups of trainees attend classes, write assignments and clear exams. In Infosys, the training program lasts for 6 months and it is fully residential. In IBM's subcontracting unit this training program is for 4 months. Other major software subcontractors like Wipro, Accenture, Capgemini etc. may have between four to six months of crash course training programs. Recruits are not allowed to take leaves at this time.*

This training is necessary because working in these subcontracting companies needs *some* amount of technical skills that individual recruits may completely lack. Trainings impart a set of standardized, modular set of technical skills that can be

generically applied across any project in a single company. After this training, recruits are ready to start working and they are placed under one of the many hundreds of subcontracting projects.

For entry-level campus hiring, generic skill software companies conduct a written aptitude test and an interview. They conduct written tests to assess a candidate's mathematical and logical aptitude. Engineering graduate recruits are not expected to have pre-existing technical engineering or coding skills. They may not be tested for these skills at the time of interview. Lack of these technical skills is made up through the intensive training program.

*At the time of interview Aman, a chemical engineer was recently recruited in a major software company. He got hired even though he has no experience with software programming in college. Yet he is one of the 236 students from his college recruited into the company. His company provides about five months of training, akin to an intensive, residential crash course in latest technologies and software programming. At the time of interview he was three months into this training.*

Subcontracting software companies hire *any* engineering degree holder from *any* engineering specialization. They have no preference for computer science or software engineering graduates.

*Most respondents trained in chemical, mechanical, electronics and other completely unrelated engineering branches. Very few were from software or computer sciences backgrounds.*

Candidates' engineering skills do not hold sway in the selection and recruitment process. Companies do not distinguish between those with significant engineering skills

or technical abilities and those that lack them. They do not test for these skills at the time of campus hiring.

*A respondent hired as a fresh engineering graduate in a major subcontracting company said, "I did many engineering projects in college and I was always good with technology. I won a few awards in inter college competitions. But the company stripped me off my technical skills and knowledge. It started at the time of technical training: I was forced to attend basic technology sessions; their content was appropriate for non-engineers that were recruited in my cohort. Stripped off my individual achievement, intellectual stimulation or technical growth I felt like I was being treated like "herds of sheep". No "self respecting" engineer can take that. After a few years I quit".*

Subcontracting software companies impart technical skills at the time of training. They do not have use for engineering skills. Yet they prefer engineering degree holders during recruitment. N Narayana Murthy, founder of Infosys and India's software subcontracting business model, explains the puzzle in a paper,

*"Initially getting the first few contracts was really difficult. Foreign corporations didn't trust the Indian workforce, "capable" enough to deliver a quality product. At that time India was considered a technologically backward, unreliable country for outsourcing breaking into the market was a significant challenge. Concerned with building his reputation, he hired engineers to do the outsourcing jobs. Even though the jobs initially outsourced were very low skill, simple technically, well beneath the complexities that engineers are trained to perform".*

Hiring engineers signaled reliability to foreign customers and allowed Indian subcontracting agencies to gain their trust. So the engineering degree requirement is an

outcome of business strategy i.e. client confidence building purposes. Hiring engineering degree holders has now become a “business requirement” and Infosys under Narayana Murthy started this precedent.

*Before getting subcontracted out to India, American high school students with programming skills did some of these tasks. But in India the same jobs often need engineering degrees. Working on pre existing pieces of software, subcontracting employees need to know how to apply different types of technologies for troubleshooting and code repair.*

These jobs require a standardized set of coding, logical and problem solving skills. It needs knowledge of some basic software programming skills. These jobs generally do not need engineering skills. Corporations hire engineering degree holders to meet the “business requirement”.

*As a hiring manager explained, “We prefer to hire those who are good students, technically decent and ready to slog it out with us for long hours. These are the people who have perseverance and can troubleshoot problems in the code quickly. This is the character trait we need in our recruits. We have no need for outstanding, technically brilliant coders. Students that are academically brilliant have a passion for coding. They will also be looking for technically challenging work. If we hire them they would find our tasks too boring and monotonous. Most likely they would leave pretty soon for higher studies, MTech or MS or PhD. Or they will try to move to another company. They are not a good fit for our needs”.*

Given the type of work, hiring managers in low end subcontracting do not prefer the topmost percentile students or technically brilliant coders. Being technically outstanding may work *against* an individual candidate in getting hired.

*"We need "resources" with engineering degrees but not high technical merit holding "engineers", according to a hiring manager.*

But the engineering degree requirement has made software jobs out of reach for India's poor. Krishna (2014) has shown most Indians are too poor to afford an engineering degree, excluding many from the software industry (and similar globalized sectors with mobility into middle class lifestyle). Unaffordability may be an important reason for the overrepresentation of privileged, affluent classes in software jobs.

But members of India's poor classes that get engineering degrees may still be left out from the software job market. They may not know that engineering degrees are not a ticket to software jobs, getting into the right colleges with campus placements ensures that. But these colleges may to too expensive for them to afford. Lacking correct informational networks to get into the right colleges; their educational investments may become worthless. Instead of middle class mobility through a software job, they may remain jobless, falling further down because of the educational debt. Hiring and placements in the software industry is unabashedly classist: favoring the affluent and well connected.

**Substandard Merit via Free Market System**

*Some respondents mentioned that after 4 year engineering they never really learnt basic coding or latest technologies. Most of their curriculum was outdated and theoretical, textbook oriented. A lot of the learning happened by rote and not applicative.*

*Some respondents mentioned that in college exams they had to memorize entire codes and reproduce them in the answer sheet. For these recruits, the training program was really the place where they learnt coding, application of latest technologies and so on.*

Many engineering graduates' lack of technical knowledge and technical abilities may be explained by the faulty engineering education system currently in place.

*According a report, India's Information Technology sector recruits the largest number of engineering graduates. But 18.43*[2] *per cent engineers have abilities to work there. Another report presented a worse scenario where 95% engineering graduates surveyed didn't know how to code*[3]*. Many of these poor quality colleges are privately owned engineering colleges, they have little accountability to any external authority and some are not accredited with the state's technology education regulating agency (AICTE).*

Clearly India is over producing engineering "graduates" with questionable employability skills. The fault lies with the engineering education system that graduates these "substandard" engineers.

*In the late 1990s as the software-subcontracting sector started to take off, industry demand for engineering degree holders suddenly exploded. State owned Indian engineering colleges were not able to meet this demand. The state liberalized the engineering education sector, until then under heavy state control and regulation.*

---

[2] "The Employability Challenge". Business World January 14, 2015. http://www.businessworld.in/article/The-Employability-Challenge/14-01-2015-78809/

[3] " "The Real Reason Why 95% of Indian Engineers Can't Code" Economic Times, December 14 2017.https://economictimes.indiatimes.com/jobs/the-real-reason-why-95-of-indian-engineers-cant-code/articleshow/62067588.cms and

With liberalization, private capital rushed to fill in the demand for engineering degree colleges. Since then private engineering colleges have mushroomed. Many are unregulated, with limited infrastructure, under qualified teachers and outdated curriculum. State surveillance and monitoring is severely limited.

*With liberalization of engineering education, engineering colleges have mushroomed in India. In addition to hundreds of state run engineering colleges; there are thousands of private engineering colleges. The Ministry of Human Resources Development (2014:64) lists 6214 engineering and technical colleges but admits that, 60% of engineering graduates remain unemployed. Other reports suggest 15% of India's 3200 engineering colleges have programs accredited by a government body[4]. A report estimates that amongst the 1.5 million engineers that India graduates every year, many of them lack basic mathematical, logical and English skills[5].*

These institutions mass-produce engineering degree holders with questionable skills and abilities. This engineering degree market is intended to serve the "low skill" software subcontracting industry that needs "engineers in degree but not in training". Engineering degrees are meant to fill the "business requirement" need in low skill, generic subcontracting sector. It is not surprising that "dubious" engineering colleges are thriving in India.

Corruption, bribery and favoritism are rampant in private colleges, costing meritocratic principles. At the same time substandard meritocracy tends to exacerbate

---

[4] "60% of Engineering Graduates Unemployed". https://timesofindia.indiatimes.com/home/education/news/60-of-engineering-graduates-unemployed/articleshow/57698133.cms

[5] Chakrabarty (2016), " Only 7 Percent Engineering graduates Employable: What's Wrong With India's Engineers?" http://indiatoday.intoday.in/education/story/engineering-employment-problems/1/713827.html

privilege- the well to do and well connected get special accommodations and eventually get hired in the software industry.

**Privilege in Making "Americanized" Persona**

Fernandes (2006) observes that in globalized India, English skills are sought to interact with the global market. Knowing English is a prerequisite to many white-collar employment opportunities. All the export-oriented sectors including software, call center and backend business processing and data entry (BPO) work are considered lucrative and require English fluency. Big urban cities like Bangalore attract scores of unemployed rural and small town youth aspiring to become a part of the global economy. In globalized India English fluency is perceived to opens doors to middle class jobs.

*Software companies assess English communication skills, interpersonal skills and ability to present one self to foreign clients at the time of entry-level recruitment. There is a written verbal ability test followed by an in-depth personal interview. Both of these are conducted in English.*

To get campus recruited in the industry, English communication and interpersonal skills are essential. Software companies thoroughly test English skills during entry-level recruitment.

*Majority of Indian software subcontracting clients are American companies. Nature of software work is such that subcontracting agency workers need to closely interact with clients. All subcontracting workers need to know how to closely interact with American clients. It's a crucial business requirement in software subcontracting companies.*

To work in the software industry, command over English language is a prerequisite. Ability to interact in cosmopolitan, globalized settings, familiarity with foreign clients' culture especially ability to interact with American clients is highly valued.

*According to several instructors and coordinators at the STP program[6], lack of English skills is a major barrier for their trainees in making it into the software industry. "When I started teaching these students I observed that they are good technically but they were not versed in communication skills. Many are from rural colleges around Karnataka and Tamil Nadu educated in their mother tongue: Tamil, Telugu, Kannada. Being able to speak English is a new thing and a challenge for some of these students. Unfortunately most companies that come for placements in IIITB are export oriented and have customers abroad (mainly US) so English fluency is essential to get hired. Suppose if their customers were Indian then this necessity of English language would not arise. But that is not the case for the software industry", according to a trainer.*

English skill inadequacy is common amongst STP trainees; the program trains historically marginalized engineering graduates to make them job ready for the software industry. Though these students may have technical skills, assessed at the time of intake into the STP program, limited English skills hinder their ability to enter the software industry.

But a thorough familiarity with English language such that employees can easily work with foreign clients needs to be cultivated over a long period of time. Unlike

---

[6] The Software Training Program (STP) is a novel initiative meant to make engineering graduates from historically marginalized backgrounds become job ready for the subcontracting software industry. It is run by Infosys Foundation. Applicants are screened for technical skills at the time of entrance.

generic technical skills that can be easily imparted via intensive training, imparting communication ability takes a lot more time.

*Professional families normally send their children to exclusive private English medium schools so their children can grow up speaking English. They are honed in "cosmopolitan", "urbane" personality skills. These skills need to be purchased in the expensive private education market.*

Disadvantaged are those from non-cosmopolitan, marginalized backgrounds. They cannot purchase expensive English medium schooling. They tend to have limited exposure to English skills and globalized cultural settings. Many grow up speaking and being educated in their mother tongue.

Jeffrey (2010) observes that mobility of well educated but small town and rural Indian youth is often constrained by their lack of English language and "urban" "cosmopolitan" social skills. Lack of these skills creates a feeling of "inferiority complex" within them. So there is a proliferation of English training institutes in big urban centers catering to this population. And these institutes are overwhelmingly popular amongst the rural youth, working class and "housewives" seeking to improve their employment prospects.

**Emulating the Professional Class**

*"Many STP students clear recruitment written tests, including the verbal section assessing English comprehension, but fail the interview rounds where their comfort with foreign, globalized culture and presentation skills is tested, according to a coordinator for the STP program.*

So the coordinators decided to move the STP training venue to the elite IIIT-B private college so that these rural students could observe and learn the professional, urban "Americanized" etiquettes. This is cultivating the professional "Bourdieusian habitus", an important part of the software industry meritocracy.

*According to STP training coordinators, "In the first batch of STP program we used a different venue for training and conducting our program. But for the next batch we decided to shift the premises to IIIT-B campus. We noticed that these students are not versed in the social cues and social modes of behavior that are mandatory for professional spaces. They don't know how to interact with foreign (American) clients. Most of these students are from rural areas and have never seen a big city. They have never been exposed to how (professional class) students behave. Most of them have never seen this kind of world-class infrastructure and facilities. They are afraid to speak up because they feel intimidated in these unfamiliar settings. Then how can they speak confidently and articulate in interviews? But now they get to interact and observe these IIIT-B students on a daily basis. They get to interact with world-class faculty. We often have CEO visits from major software companies. They can observe them and learn. After six months of having these students on IIT-B campus, I have seen major changes in their confidence, articulation".*

Based on interviews with STP students and trainers, I found that during software subcontracting campus recruitments, interviewers routinely assess a candidate's cultural "fit" through "soft skills". I also spoke to a trainer teaching skills like teaching English, etiquettes, software and programming skills:

*"They don't understand the cues of professional setting or how to be professional in conduct. Many of them would just turn up in flip-flops and unprofessional clothes. They have no idea it is not proper to do so. They don't know that better dressed people have better chance to get employed. They do not know many other aspects of professional conduct like what is good/acceptable touch (with the opposite sex) versus unacceptable or bad touch. Most of them have no idea about these things. These students do not know how to present themselves like urban, polished professionals".*

English fluency is an essential but a small component within a gamut of abilities under "communication skills". Desirable communication ability in the software industry implies a clear and distinct "bias" for candidates from urban, cosmopolitan areas and professional class homes.

I spoke with one of the communication and cultural skills trainers for the STP program. He trains students, changing their behavioral patterns and attitudes so they can be presentable in professional and corporate settings:

*"We teach them the right posture and body language. We teach them how to dress and we start with what are business formals. What kind of clothes to wear to an interview, what kind of dress is needed to meet with foreign clients or business meetings. There are things like "you don't put fragrant oil in your hair" or your clothes need to be well ironed and crisp to make a good impression. We provide them with exposure that was denied to them because of their disadvantaged backgrounds".*

Cultural training is an essential component of the curriculum. Here trainees are taught how to present and appear to people. In software corporations, desirability of a candidate is based on whether they are a good "fit" into their workplace culture.

*According to a STP trainee student, "All my life education has been a top down approach: the teacher would come for lecture, we would take notes, learn and memorize and pass exams. There were no presentations or discussions in class. Here it's different. There aren't a lot of textbooks or memorizing. We are made to think on our own. I used to have trouble speaking up or articulating clearly but now I have overcome that through training and practice. So they are trying to build our confidence and I can see that. I have found that the STP program is very interactive. We are trained to make presentations and come up with innovative solutions on the go. This approach is very different from where I studied.*

Students' from disadvantaged backgrounds may go to schools and higher educational institutions. But these institutions may fail to provide them with opportunities to learn skills needed to have a professional career. "Self confidence", ability to present oneself in professional settings are such skills. Recognizing presentation and articulation skills as crucial hindrance for their trainee students, STP program has several training courses and modules. Students are encouraged to make presentations and learn to address groups of people in an articulate manner.

Jodhka and Newman (2007) observed that in meritocratic globalized India, corporate interviewers are very interested in a candidate's social or family background. Typical questions are "how educated are your parents", "what is their occupation". And candidates with professional family backgrounds are deemed a much better fit culturally than those from clearly working class, or agrarian backgrounds (Attwell and Thorat 2010). The latter thus identified and filtered. India's scholarship on caste discrimination shows this as one way in which SC/ST (historically marginalized) candidates are

routinely filtered out of the hiring process. Most of them come from disadvantaged therefore "unfitting" family backgrounds (Deshpande and Newman 2007).

*An STP instructor provided a very interesting insight into "what it means to be a good fit" for India' subcontracting corporations, "When I say these STP students do not know how to present themselves I mean they do not know how to make professional, intelligent conversation. For instance to impress the interviewers, one needs to know the latest developments in technology, what are the new gadgets, what new technologies are out in the market. This shows their intelligence and their passion for this sector. We try to encourage them to visit websites where they can learn these things. We cannot teach all this in the short time we have. And besides these abilities need to be honed over time".*

Lareau (2011) in a study between working class and middle class American child rearing practices found that middle class families are successfully able to transfer "habitus" to their offspring. This habitus helps offspring of professional families negotiate with authority figures much more successfully than working class children. This inter-generational transfer of "habitus" may explain a key trait categorized under "self confidence" that modern globalized corporations look for in their employees. A confident professional persona is created at the educational institutions and at home.

*Most of my respondents are from the professional class, educated in English medium schools. Many of them mentioned that while growing up, their parents prioritized English medium schooling. Their families saw it as an essential investment. Some respondents mentioned, their family moved residence or changed jobs so that they could be closer to a reputed English medium school. This is how important good schooling is for the professional class in India.*

These families take cultural capital investments seriously by sending their children to reputed English medium schools. These schools have high academic repute and teaching standards. But they also hone and cultivate urban, cosmopolitan, professional etiquettes that help these students flourish in their globalized sector jobs. “Habitus” is cultivated in the formative schooling years that emerge by enrolling children in “good schools”. These good schools are generally higher status, expensive, English medium schools.

*Janani ‘s father started as a janitor and eventually moved up into blue collar trades occupation in a public sector corporation industrial township. “Since my father was an employee of that company, my sister and I studied in the high quality English medium school run by the company for all employees in our township. I had the rare opportunity to study with the children of our CEO and top management. And I feel proud to admit that most of my classmates have high profile engineering and medicine jobs. A number of them went to the US and US to get their PhD degrees and most of them are highly successful professionals. Being in the company of these successful classmates made me aspire to be professionally successful. Though I have been an average student all my life I was able to claw my way up in the software industry and today I am a manager in a major MNC. And I credit my school for my achievements. It honed in me the extroverted, smart, fluent in English persona that I have today. It gave me the confidence to converse easily with American clients, make them feel at ease, so they trust me. And my communication and networking abilities with American clients allowed me to bring more business for the company so I am professionally rewarded for it. With my confidence and abilities I could make many friends and contacts in my company,*

*including at the higher-level management". Compared to her, the rest of her extended family is into manual labor and other blue-collar work like tailoring. Girls in her family are married off by the age of 18. So she is considered the most successful person in her entire extended family.*

When students from working class families gain access to these exclusive private schools, they get to "rub shoulders" with children of high status, professional families. They can learn correct conduct and behavior; their schooling trains them in acceptable cultural codes or Bourdeusian "habitus" necessary to operate in globalized, cosmopolitan settings.

Software subcontracting companies favor candidates from privileged professional class families over candidates from poor, historically marginalized and disadvantaged social backgrounds. "Habitus", "professional persona" "not knowing the cosmopolitan professional codes" may be an important reason impairing candidates from disadvantaged families to get into the software industry.

**Exclusion in Mentoring Networks**

*Growing up in Bangalore's slums, born to daily wage laborers, Nisha always looked with awe and yearning at the distinctive way in which the smartly dressed professionals around her looked with their laptops and badges. And she just knew that whatever work they did must be important and well paid. So really wanted to be like them and do the job they are doing. But she didn't know what job or education they had. . Later on it was through the local NGO run by Free Software Movement Karnataka (FSMK) operating in her slum that she these people are software professionals. Volunteers working in the software industry run the NGO. And these volunteers helped*

*her prepare for the engineering entrances, select the right college, fill out the admission forms and so on. Her parents are primary school dropouts and no one in her family or neighborhood is educated and most work in daily wage labor. Her mother works as the domestic help in those "big" houses and his father works as a painter in the local mechanic workshop. No one in her family had ever done engineering or had any idea of software profession. Without the NGO's help there is no way she could think of a software career.*

For the professional strata, informational networks provide essential mentoring and career guidance. They get this mentoring at educational intuitions, through networks with professionally connected friends and relatives and also through their family members at home. Working class families are generally deprived of this mentoring because they usually do not have points of contact into the professional world. The professional world of software career and higher education is very far removed from their everyday realities. Therefore many of them have no idea about the software career, engineering education, and importance of choosing the right schools or colleges.

*Sumit is one of the few dalits who made it into the software industry. His traditional caste occupation is "Chamar", skinning and disposing off dead animals and working with leather. Growing up in untouchable ghetto with slum like conditions, he painfully admitted to living in filth and squalor in a run down shanty he called "home". His neighbors and fellow residents were barely literate. Many children didn't go to school. His neighbors and fellow residents worked in low paid wage labor. His parents are primary school educated but they were always very keen on giving good education to their children. They knew education was the only way to get out of their misery and have*

*a decent, respectable life. His father was able to secure a government blue-collar job on reserved quota for dalits so there was steady income. But beyond admitting him in a good English medium school, his family had no other way to guide or help him in his career. He had no idea about engineering careers and no one had ever heard of a software career in his family and neighborhood. Being bright and studious, Sumit was really liked by his teachers and one of them took a personal interest in mentoring him. It was this teacher that guided him into the engineering field and exposed him to the software career. He helped him select the right college and prepare for the competitive engineering exams. Sumit worked hard and gained admission into a prestigious engineering college and eventually made it into the software industry. Without these external interventions it is doubtful that he could have made it into the software profession.*

Sometimes disadvantaged families may come to know that they need to provide their children good schooling and professional education. Even if their children do well academically, their families are not able to guide them into a professional career. Their families are unable to provide the necessary mentoring or guidance. Such students tend to rely on external sources- a stroke of chance or luck. This mentoring is generally assured for the urban professional classes. They have several points of contact into the software industry and the professional world. This is the asymmetric nature of these mentoring networks: privileging the well-connected professionals and excluding those from disadvantaged backgrounds.

**Privilege is Getting Multiple Chances for Success**

Graduating from low tier colleges is a major set back for a software job aspirant. But for candidates from wealthy families there are multiple strategies and pathways by which one can climb back into the software industry; these pathways boil down to families' spending abilities and access to informational networks:

*After graduating engineering Preet couldn't get any job placements because her college had no placements. For her it was impossible to join any open recruitment drives because she was from a small town and such opportunities don't exist. She found many of her classmates were moving to big cities to join some reputed software training institutes. These institutes provide rigorous training in coding, programming and latest technologies. Afterwards they have on campus placements. So she and a few of her classmates moved to New Delhi to join one such reputed training institute called CDAC. She knew nothing about this institute till she moved to the big city. Here she got to know about the entrance exam for this institute and trained for 6 months to clear it. After training in this institute she got trained in some hotly demanded technologies and made it into a major software company through campus recruitment. Meanwhile her affluent professional family, paid for the high living costs in New Delhi, costs for preparing for the entrance test, tuition and living expenses when she gained admission in the training institute.*

When aspirants cannot make it into their chosen jobs in the first round, informational networks provide guidance on what next steps they need to take so they can climb back into their chosen career. And affluence allows families to bank roll these "next steps". Offspring of well-connected, professional families gain multiple chances in life to succeed.

*Mobility via Bottom Tier*

*Deep graduated in BE electronics and electrical engineering from a very low tier college. He was always very poor in academics. It was clear there was no way he could land a good software job with such credentials. However he has several relatives in prestigious software companies. They provided him guidance on what to do next: They advised him to join a small, low tier software company. While in his engineering degree, they helped him do a good project that honed his technical skills that he could showcase it in his technical interview. He trained in the latest areas of JAVA and SAP. Both were highly demanded at that time. They advised him to attend many technical seminars to hone his knowledge of technologies. Later he joined as an intern in a low tier company. It was a few years he started getting enough pay to cover his living expenses. But eventually he transitioned into an MNC and now earns a good salary.*

Using small, low tier companies as transit, one may move on to major software corporations. Typically those who graduate from low tier colleges or do not get campus placements in major software companies try to work their way up via small, lowest tier companies.

*Ramu graduated his correspondence based computer application course when the industry was just starting to take off in the later 1990s. But with his correspondence course he didn't have much chance to land any good job i.e. a job with a major company. Instead his family members already in the software industry advised him to join any software company and gain experience so he could move to a major company eventually. So he ended up working in a Hyderabad based low tier software company as an unpaid intern for almost two years. After that the company absorbed him and started paying him*

*a nominal salary. He had to borrow money from his family to meet his living expenses. After 5 years of experience in multiple companies like these he was able to join a major software company on a lateral hire.*

Ramu was somewhat fortunate, even though he is from a rural, lower middle class family, he had significant social networks in the software industry who could guide him in his career. Despite his poor educational credentials his social networks mentored him in his software career. And he was not burdened by family obligations so he could bear the opportunity cost of interning with low tier software corporations on nominal pay.

*According to an STP instructor, "There was a student who dropped out midway in the STP program merely because he got a job in daily wage work that paid slightly higher than the current STP stipend. So he prioritized his immediate economic survival over professional development. Most of these students are from farmer, daily wage laborer, domestic help type parental backgrounds and they are first generation college students. Their parents are not educated or capable to guide them correctly in their careers. Then there are issues of glaring economic deprivation. I know of one student from a poor family that would send almost her entire STP stipend money home. The stipend is meant to cover living costs in Bangalore. Then there was one student who had only two shirts and two slippers and he wore that the whole 8 months of STP. Under so much financial duress it's difficult for them to try out new things to enrich professionally. Families have no idea that with poor academic backgrounds, building a software career has to be achieved through a lot of struggle over several years. So they expect to join a major software company right away. Therefore lack of familial awareness about the*

*software career adds to their constraint. They cannot get adequate support from families to build their career".*

For many economically disadvantaged and burdened by family obligations finding alternate routes into the software industry may be difficult. At times they are under pressure to earn right away because they may have several younger siblings and family members who are depending on them to provide. Opportunity cost for them to build their professional skills by joining low tier companies is too high. These companies either pay nominal wages or none at all for the first few years. Their poverty and short-term survival orientedness inhibits their professional growth.

*Investing in Technology Training*

*Kaushik graduated from a low tier engineering college with poor grades. His college has zero campus placements so he was on his own to get placement in a software company. He knew he had no chance of making it into an entry-level hire in the major software companies. At the time of interview his strategy was to join the industry through lateral hire. "In the lateral hire these companies test technical and coding skills because they do not provide in-house training. It helps that I have several relatives in major software companies across India. They are able to provide me with information on the industry, what skills and technologies are in demand and how to prepare for entry. I joined technology-training classes in the final year of my degree program. Some of these latest technology courses like Red Hat courses are highly demanded and may cost a years' salary of many middle class families. Additionally I joined another training institute providing mathematical and logical skills to clear entrances of companies. Many of these training centers are good quality and have ties with software companies that*

*come to recruit in these centers. Many of my seniors have followed this route to make it into low tier software companies. Eventually they transition into major software companies" according to Kaushik.*

At times small companies may not have in-house training programs. The onus of getting this technology training lies on the aspirant employee[7]. After gaining technical skills these aspirants try to get hired in low tier companies and after some years of experience they hope to eventually transition as lateral hires into major companies. And this technology training is usually prohibitively expensive.

*Informational "Mentoring" Networks in Training*

Cities with software concentration like Bangalore are teeming with thousands of software classes and training institutes. Fresh engineering graduates come into these cities to get training in latest technologies hoping to land a good software job. But additional reskilling means additional expenditure on training and living costs. Affluence enables this sort of enrichment for job readiness.

Unless the trainee has a clear knowledge on what technologies to train in so they can get hired, such training can be worthless. Typically this sort of information flows easily through personal networks. For instance ITEC, the Bangalore based forum for IT professionals found out that there are many fraudulent training institutes operating all over Bangalore. These training centers especially target out of town job seekers and dupe them into getting training from them at very high cost and promise of recruitment after

[7] This trend has become the norm since the 2008-10 financial crash as the Indian software industry has experienced an overall slowdown and a glut of engineering graduates looking for software jobs.

completion. These trainees have very little awareness about the industry or what skills are needed. ITEC has received numerous complaints from such victims. Many paid a lot of money to learn useless courses that did not help them move into the software industry. Some borrowed money from their poor families to join these courses and got duped.

*A member of ITEC said, "Even if a training center is not fraudulent and provides genuine technical training it may still be worthless. This is because of the wide variety of courses offered and the trainee needs to accurately know the current technology trends and demands in the industry. Only then such a trainee can intelligently choose the correct course. And these courses are often very expensive. For instance JAVA is a popular technology that can get someone hired into the industry. Many aspirants know this as a part of common knowledge. But there are many versions of JAVA with differing intensity being taught: there are JAVA courses that cost 3000 rupees and there are JAVA courses that cost 30,000 rupees. The difference is in the components of JAVA that need to be learnt. What features and components of JAVA should an aspirant spend their money on learning? For this one needs to have an in-depth knowledge of what is needed and demanded in the industry at that point. Without such prior knowledge its almost impossible for a newcomer to know what technologies to train in that will land up a job and it will be a waste of money. One needs to know very clearly what is needed in the industry at that point to make this investment worthwhile".*

It is clear that privilege in the software career provides multiple opportunities in life: When poorly performing students from affluent families fail to progress via merit, their families can buy them college admissions. Upon graduating if they are still unable to make it into a software career their families are ready to invest in additional skills that

make them job ready. Informational networks that professional upper middle class families can easily access bolster their privilege. These networks provide vital guidance on the next steps. This is how professional class of India reproduces its privilege.

**Affirmative Action Politics**

*In 2005-06 the movement for affirmative action (reservations) for India's historically marginalized groups SC/ST/OBC peaked in India's private sector peaked. Indian affirmative action policies were restricted to the state owned sectors; a fixed percentage of all state sector jobs were exclusively reserved for the historically marginalized, increase their participation in middle class jobs. As the state sector shrank, middle class mobility for these groups has allegedly declined. Thus India's historically marginalized demanded reservations in the private sector alleging that New Economy sectors like software are actively excluding them. This movement gained much traction and generated national level pressure. India's professional, middle class, largely upper caste in composition and privately owned software industry was a major vocal opponent. Industry leaders Narayana Murthy of Infosys and Azim Premji of Wipro however acknowledged the limited representation of "weaker sections" poor, dalit (SC/ST) and OBC in the "New Economy". Their contention was that reservations are un-meritocratic, so they would let in less capable people, compromising India's global competitiveness. In the end the pro reservation movement fizzled out.*

My research shows that the subcontracting software industry favors privilege and actively excludes disadvantaged groups. Hiring from private engineering colleges with questionable meritocratic and fairness standards, it has failed to uphold meritocracy.

Claims of its CEOs and leaders that reservations would somehow reduce the industry's "meritocratic" standards seem unfounded.

*Acknowledging the limited representation of India's historically marginalized, N Narayana Murthy, founder of Infosys argued that underprivileged candidates are lacking in merit so they are not able to enter the software industry. But with sufficient training these candidates could be given the same competencies to enter the software industry without reservations.*

According to Murthy, to address the limited representation of SC/ST/OBC, providing them with right training and skills would solve the problem. Reservations are not necessary. To prove his point he started the STP program. Evidence from the STP program shows that intensive training can make some of these marginalized students job ready for the software industry. After training these students for over 6 months, STP program helps with placing them in different companies.

*The STP Program*

Infosys Foundation runs the Software Training Program (STP). Eligibility is for economically disadvantaged SC/ST/OBC candidates[8]. Minimum eligibility is engineering graduate[9]. Applicants are screened for technical ability and also there are several rounds of interviews. According to one of the founding trainers of the program, that trained the first batch of students:

*"We received about 200+ applications the first time we started this program. All these students were from disadvantaged backgrounds. We had the capacity to train 100*

[8] They also admit candidates from poor families using an income criteria. But a significant number of their trainees are from these disadvantaged castes.

[9] They need to maintain a minimum grade of 2.4/4 in college and minimum score of 60 percent marks throughout high school.

*people. But only 90 students could clear our screening tests and interviews. Initially we didn't want to relax our selection criteria because we wanted to retain meritocracy. But the next batch we decided to relax our selection criteria by five percent just to make sure we could adequately fill our seats", according to a founder coordinator.*

Since the first training cohort in 2006 it has trained multiple batches of students from such underrepresented backgrounds, making them job ready for the software industry. The training program runs for 6 months in a fully residential setting and provides trainees with a monthly stipend to adequately cover living expenses in Bangalore.

*At the time of interview, the trainers and coordinators expressed frustration that in the current batch they were able to place less than 50 percent students so far. In their first year of operation in 2006-07 when the industry was still in the growth phase and hiring rates were robust, the STP program was not able to place all students. It never achieved 100 percent placement record. It is unlikely in this round they will place all students.*

The STP placement program had limited success in placing all students with software companies. It never had achieved 100 percent placement record.

*Ironically the STP program is modeled on the post graduate diploma in software development (PGDSD) curriculum of Infosys owned private engineering college IIITB (Indian Institute of Information Technology-Bangalore)- a paid program with 98% placement record at a time. They got the best practices of the PGDSD program into designing the STP. It is designed to make the people employable and enhance*

*employability of students. Its current instructor ran the PGDSD program for four batches before taking up the STP program. The program is very expensive.*

The STP program curriculum is tried and tested with excellent industry placement records. The limited success in placing STP students is a source of frustration for its teachers and coordinators:

*According to a STP trainer, "I feel people from marginalized sections (SC/ST) with good education can get employed in the software industry but they don't have enough exposure. Many of these students in STP are very good technically but they don't have enough soft skills, don't know how to present themselves and sell their skills in the market. Personally for me it's frustrating that I cannot get all my students placed. Teaching these marginalized students also made me realize that some of interpersonal, cultural skills are not easily taught or transferred. In my entire instructor career, I have never really experienced this sort of a hurdle".*

Since 2008-10 economic recession, India's software industry growth phase has tapered off. Hiring in the software sector has slowed down significantly. Given the oversupply of candidates wanting to work for them, companies have become pickier in the hiring process. Such trends may have worsened prospects for candidates from marginalized backgrounds.

**Summary**

Hiring in the subcontracting software industry shows a distinct structural favoritism for the professional, affluent, well-networked class of India and exclusionary bias against the socially disadvantaged groups. Privilege plays an important role in securing a software job for instance through exclusive private colleges and informational

networks on one hand and professional “habitus” cultivation at professional homes and English medium grade schools on the other. This explains the over representation of professional class and upper caste in the software industry. In acquiring the necessary cultural capital to make it into the industry, this class resorts to un-meritocratic means and strategies. Some use nepotism and bribery to get around the admission process. Private engineering colleges operating under lax state regulation enable this behavior.

To fulfill the “business requirement” needs for engineering degree holders in the subcontracting sector, many private colleges churn out graduates with questionable skills and abilities. But they get away with providing substandard engineering education because the subcontracting sector doesn’t really need engineering skill holders. The relatively low skill, standardized work of software subcontracting allows major companies to train recruits lacking technical skills within a relatively short period of time and make them ready for project deployment. It is ironic that the software industry has acquired a strong meritocratic and fairness image in India.

# CHAPTER 3

## Privileged Precariat: Insecure Jobs, Flexible Workers and Precarious Work Culture

*Major subcontractors like Infosys, TCS, Wipro, IBM (GBS) and so on employ between 100,000 to 300,000 employees. They have hundreds of projects. Given the large scale of operations, they have a highly bureaucratic and tiered corporate hierarchy: entry level, junior management, middle management and senior management. The management structure is pyramidal. All major software subcontractors are bottom heavy. Entry and junior level workers do all the production work. Middle and senior management perform supervisory roles. Fresh trainees join at entry level.*

Project allocation is the next step in an entry-level recruit's career. It is done once freshly recruited graduates complete their intensive technical training. The training lasts for four to six months. It is meant to make them ready to start working on live projects. Major software companies administer this training to over 100,000 fresh graduates they hire every year. Most of their hiring happens at the entry level through campus placements.

*For better management, each project is subdivided into modules so technical teams can work in separate sections within a project at a time. At the entry level, groups of junior and senior software engineers make up teams and work in each module. Team leads and module leads (junior management) manage project teams and modules. Every project has a project manager and all workers for a particular project report to the project manager (middle management). Each project operates under the direction and ownership of a delivery manager, considered senior management. Projects have large headcount: over 50 people in each project.*

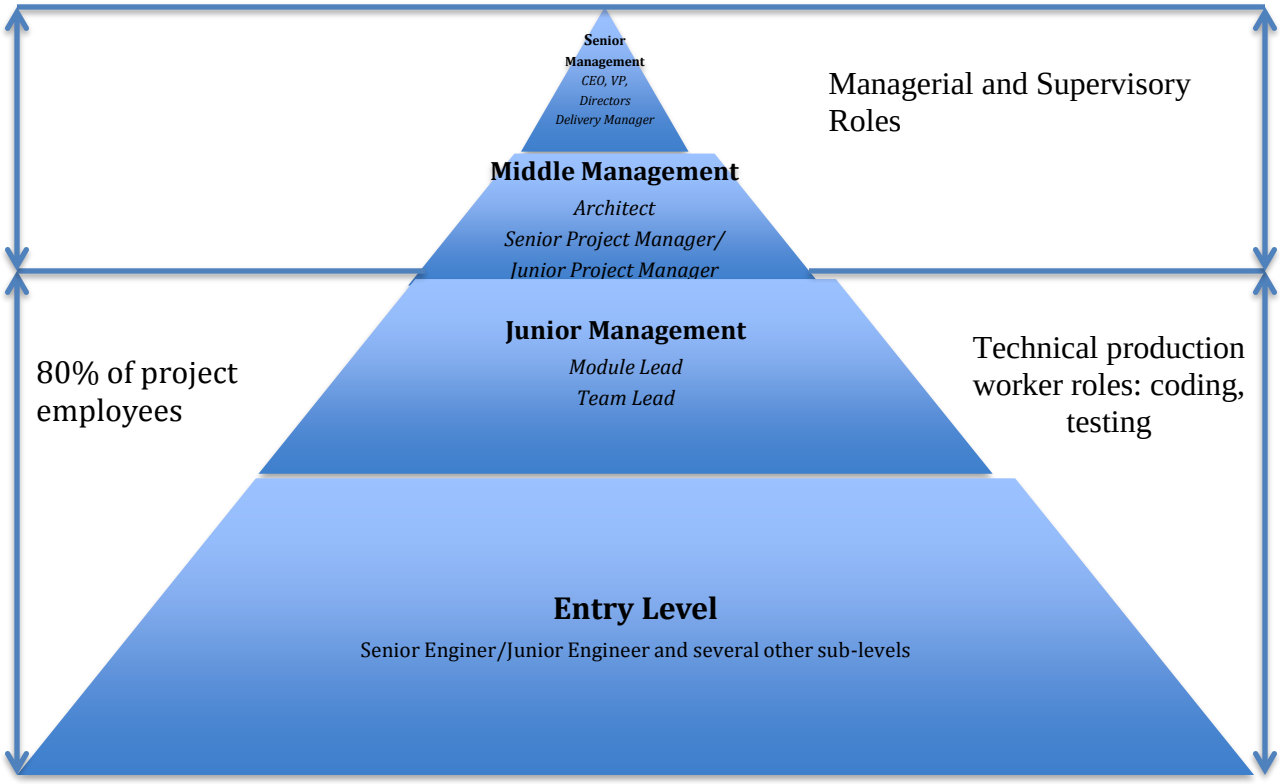
With project allocation, trainees "formally" start working for the company. As employees they are given the job title of junior engineers (designations may vary by company). Their direct supervisors and mentors are the project team leads i.e. junior management. All project teams work under project managers (middle level management). The delivery manager (senior management level) is the final authority for the project. This is the project management structure in major subcontracting companies. (In the diagram below I describe the management hierarchy in major subcontractors).

*Eighty percent of project employees make up the bottom two levels in the corporate hierarchy. Bulk of project employees lie at the bottom levels.*

This is where the actual "production" work needed in the project takes place. It can be coding, testing, documentation or some other tedious, labor intensive task. This work needs large manpower deployment. At the same time it is relatively deskilled i.e. neither requires significant years of experience nor many years of technical training. Entry-level level workers generally fill these jobs. Junior level management i.e. team leads also do production work. But they also provide mentoring and direct training to entry-level workers. **In this chapter I address these bottom two employee levels in the subcontracting hierarchy.**

Project managers and employees above them do not perform technical production work. They are the managers. Middle and senior level management do all the supervision, coordination and "people management" work**. I address these higher-level employee functions in the next chapter on "Mobility into Managerial Roles".**

**Project Management Hierarchy in Major Subcontractors**



**Unequal Opportunity Projects, Uncertain Career Prospects**

*Major software companies have hundreds of different projects needing lower end services like support, maintenance and testing. Fresh trainees are allocated to any one of these projects. Here they learn how to apply their skills on the job. These projects are written in different technologies that have unequal market demand. Until actual project allocation, individual trainees have very little idea where they will land up: in technologies with dying market demand or technologies with expanding demand. This allocation significantly shapes an individual employee's career prospects and future employment opportunities.*

Through project allocation, fresh trainees develop "on the job" technical skills. These skills determine their market value and future career prospects. If trainees get allocated to projects in technologies where subcontracting work is dying out, their career prospects will suffer. They are headed for a career setback. If trainees are allocated to projects in technologies that are expanding in the subcontracting market, their careers will get a major boost. Project allocation significantly determines a trainee's future.

*When software work started relocating to India, mainframes based work was a key segment of the growing subcontracting market. Ironically mainframes were phased out elsewhere. But its demand in India was driven by US based finance sector companies that shipped all their mainframe based systems, called "legacy systems" to India. With projects in mainframes work increasing, demand for mainframe skills suddenly grew in India's subcontracting market. Currently clients of mainframe-based projects are migrating into newer technologies. So mainframes support projects are dying out in India as well. It's a matter of time until all jobs die out in this area.*

Until mainframe projects kept pouring in, market demand for these skills remained high. Employees holding these skills had assured employment. Though mainframe has been obsolete for many years, subcontracting companies kept hiring new employees in that area. With sudden fall in projects in this area, employees in this technology are becoming jobless. Volatility in technology demand creates job insecurity for employees. It starts at the time of project allocation.

*When Manish got his entry-level project allocation in a mainframes project he just took it and continued in the same project. It's been several years since he has been working on that project. But he fears his client will move to another technology because mainframes are becoming obsolete. He fears his project will send soon. He is unsure about his future and job in the company. He fears he will be fired because there is very little demand for his mainframe skills.*

When subcontracting employers run out of projects for existing employees, they fire those employees. Employees with skills in dying technologies become job insecure. Future employability of an employee depends on what technologies current and future subcontracting projects are likely to demand. Their salaries are tied to the market demand for their technology areas.

*Right after his training, Subhash was allocated to a big data project. "I worked in this project for two years and picked up enough skills such that I could provide support and maintenance to many big data based projects. I admit I am no expert in big data, my skills are basic but they are in high demand right now. Big data support projects are increasing in the Indian subcontracting market. I get many job offers every month from rival companies. They are offering relatively higher pay if I switch my job. I am*

*considering taking up one of these offers. But at the time of allocation I had no idea this technology would be so highly demanded. It's really a stroke of luck that I landed in this project".*

Currently expanding technologies are big data, cloud computing and automated testing. Subcontracting projects in these are increasing rapidly. Demand for skills in these technologies is high. Companies pay much higher for these skills. Fresh trainees allocated to projects in expanding areas and technologies get to develop skills and knowledge in areas with high market value. They are headed for a secure, lucrative career in the near future.

Unequal market demands for technologies create an unequal opportunity structure for employees and uncertain career prospects. These inequities start with project allocation. But trainees do not have much choice or say in technology assignment at the time of project allocation. Companies randomly allocate them to technology areas that employees are forced to accept.

*Unequal career prospects also arise out of functional area allocation. Fresh recruits are randomly allocated projects in either testing or code development functional areas. Code developers have better long-term career prospects than testers because it tends to be better paid in long run. There are more job openings in the code development area than testing; some respondents estimate that averagely, the ratio of developers to testers may be 6:1 per project. Testing is relatively more monotonous and more likely to get automated. Testing career track is less secure.*

Like technology allocation, companies randomly allocate individual trainees to functional areas. Individual recruits do not have much say in these allocation decisions. But these decisions significantly determine employees' professional futures.

*According to a HR manager, "Code developer career path is more popular compared to testing career path. We do ask recruits about their technology preferences so we can best match them with project requirements. But we are dealing with thousands of recruits at a time. Given this volume of people, finding the right match can be complicated. But most importantly if we have more projects in testing functional roles then we have to send them there. If there were greater demand for code developers, then more developers would be trained. A similar logic applies to technical platform allocations i.e. Java vs .NET. Those unhappy with their allocation are free to leave. No one is forcing them to comply".*

Companies make these decisions based on their internal project demands and requirements that individual employees have to accept. Many respondents said that switching between functional areas or technologies is not allowed in their companies. Those wishing to switch have to find another job.

*Smita was allocated to a manual testing project after her training. " I worked in manual testing for five years on a fixed term project. When I did a market check, I discovered demand for manual testing jobs was declining rapidly. They were rapidly getting automated. I suspected I might become unemployable unless I reskilled in another technology. Then I found out that selenium-testing area was expanding. I quit my job, took a one-year break to skill in that technology and found a better-paid job with my new skills".*

To continue remaining employable, those with dying technology skills need to proactively move into new technologies. For that they need investigate the new technologies, decide the ones with the most promising future and gain skills in that. But most importantly they need to know the current marketability scope for their technical skills.

*Software companies are allowed to hire and fire employees under the state's labor laws. Based on internal requirements and needs they terminate or retain employees. Depending on internal policy major subcontracting companies may give few months of termination notice.*

Job insecurity in Indian software subcontracting market is enabled by labor laws that allow companies to "hire and fire" employees flexibly. The Indian state doesn't provide employment protection or labor rights to software industry employees. In India's subcontracting sector, companies adjust their employee pool based on market demand for employees' technical skills. If the employee has skills in an obsolete area with no projects, the company fires them. Employees with skills in dying technologies are constantly afraid of losing their jobs. They cannot get comfortable with their current skills.

*According to Manish who works on mainframes, "My biggest mistake was that I got comfortable with this company and project. When I joined mainframes, it was a technology skill in demand. Over the years I didn't quite realize this technology was rapidly losing its employability. But I know very soon my current project will end because the client is likely to migrate into another technology. My company will fire me because I don't have skills in any other technology right now. Then how will I find*

*another job? I have no idea. I tried to train in .NET or some other area with demand. But it's impossible to do that on the job. My company is not allowing me to train "on the job". I need ready made .NET skills if I have to join a .NET project. Given my current work pressure, I cannot acquire those skills while holding on to this job. I have to quit my job to reskill. But then how will I pay the bills? So I am caught up in a difficult situation right now".*

Once allocated to specific projects, companies generally do not allow employees to switch technologies or functional areas. But once their skills become obsolete companies do not take responsibility in retraining existing employees.

*According to a manager, "When clients move to a new technology, companies would like to look internally and find those technology skills that match their needs. Else they would find a new set of "hires" with these skills. And companies would get rid of workers who have technical skills with no projects in that technology or area of specialization".*

Given the oversupply of generic skill software workers in the market, companies expect to find replacement workers on demand. They don't have to invest resources in training existing workers. They can hire new ones easily. To remain continuously employed, workers have to anticipate future trends in the technology market. They need to reskill ahead of time in another technology that has market demand.

*According to a respondent, "To survive in the subcontracting industry, every three to four years you need to gain new skills in a new upcoming technology. Acquiring skills in latest "hot" technologies is essential because existing technologies have a "shelf life" and then they will expire. So new ones need to be adopted on time. This way you can*

*remain marketable. "Reskilling" takes several months and training needs are intensive. You need to quit your job to do this. But every 6 months you need to upgrade your skills in an existing technology to stay relevant, keep up with the changes in the current technology you are working in and incorporate these changes in your work. You can do this technology "upskilling" on the job. This pressure increases our working hours".*

Employees' market value depends upon the demand for their technology skills. When subcontracting companies get fewer projects in a technology area, market demand for skills in that technology declines. Employees with skills in that technology lose market value. Individual employees need to be aware of these technology trends so they can be ready with newer skills whenever there is a technology shift. Technologies tend to start losing market value periodically. Employees need to train in new technologies so they can remain employable. Otherwise they lose their jobs.

*According to Avanti, a manager with a mass recruiter, "You have to spend your personal time after work to keep up with the technological up gradation pressure. The company is not going to provide extra time or relax deadlines. It is challenging to manage this with the current workload. And at times it can be harder if you are in a project that is very demanding and makes you work 12 hours and takes up you weekends as well. Companies provide limited resources like discussion groups and presentations to help employees. You could avail them if you can find the time".*

Individual employees are responsible for all technology-training needs. They are essentially left on their own to figure out ways to meet technological up-gradation needs. Technology "upskilling" or learning new upgrades in an existing technology can be done "on the job". It translates into longer working hours. But it is difficult to "reskill" in

entirely new technologies while holding on to the job. Whenever they need to reskill, they have to quit their jobs. So employees need to periodically quit their jobs to meet their reskilling needs. They remain in a state of continuous and perpetual learning, training and development.

**"Benching" and Pervasive Job Insecurity**

"Benching" or "being on the bench" is "officially" the time an employee spends either transitioning between projects or waiting to be deployed to a new project.

*Before bagging new contracts companies need to show potential clients the number of workers they have available for the proposed project. So companies recruit in anticipation of future projects. Newly recruited workers can be benched once they get hired. When a project comes to an end, workers in that project are moved to the "bench" meaning they are waiting to be allocated to another project and currently sitting idle. Sometimes contracts end midway then workers deployed to that project are similarly benched. Therefore bench refers to workers ready to be deployed to a project.*

Software subcontractors practice benching because it gives them the flexibility to move existing workers from one completed project to a new one within a short period of time. This way subcontractor companies can deal with the unpredictable client project market.

*Some projects come with very short notice, i.e. within a month a project can materialize. Project requirements may change within a very short time. But project deliverables tend to be strictly time bound, i.e. they need to start by a fixed date. So companies need a pool of workers ready to be deployed at a moment's notice. Benching allows them to do that.*

Therefore benching gives extreme labor flexibility to subcontractors but it creates uncertainty in employment terms for existing workers. It penalizes employees and forces them to go through uncertainty even though they are continuously employed that I discuss next.

Firstly, companies automatically layoff any worker not tied to a project after a certain period of being benched: *In many companies it's a 3 month period of being benched after which an employee maybe fired.*

Being benched means reduced pay. *This is because benched employees make only the base pay. While there is no performance based variable pay. This can mean a significant pay cut for the "bench" period".*

Benching is a career setback for any worker causing fear, uncertainty in career and professional prospects. In subcontracting companies, individual employees can be "benched" anytime in their career. To get out of the "bench" individual workers have to get alternate project deployment i.e. find a new project and join it.

*Benched employees are not entitled to automatic project allocation. Those that don't get a project allocation within a fixed period of time are terminated. They are under pressure to find alternate project deployment.*

The onus of getting next project deployment is entirely with employees. Software companies do not take responsibility for finding the next project for the benched employees.

*"In our company we have an internal tool where we are able to see project requirements and skill set needs so we can see where we can fit. The next step is to make requests with the HR. And you hope to get a deployment because it depends on whether*

*the project managers want you or not. So it's a hit or miss scenario and not automatic project deployment. But until you get a new project deployment you have to keep trying, while your career and your future hangs in limbo", according to a respondent.*

Many respondents said that relying on personal networks is much more helpful in alternate project allocation than going through official channels.

*Asit and his 50 fellow cohorts of new recruits were allocated to a project when it was ending. After working very hard for 6 months all of them got the dreaded "bench". "I had just joined less than a year, I was new and I had no idea what to do to get out of the bench. I kept making several trips to the HR for the next project deployment but that didn't help. Luckily I had many classmates in the same company working in different projects. I spread the word through this network. They started making enquiries on my behalf with their managers. And in two months I landed a project in a former classmate's project", according to this respondent.*

Benched employees become desperate to get out of the bench as quickly as possible. Some of them may be ready to go to great lengths to get the next project.

*According to a respondent, "When I got benched I started to get stressed out about the next project allocation. I didn't trust the official channels for help. I approached my personal network of friends within this company to find me find the next project. When I found a good project match through a friend, I still needed to go for a formal interview with the project manager. I knew I would get it but I was ready to leave no stone unturned to get that project allocation. I decided to make friends with the project manager just in case. At that time I was so desperate for a new project, I didn't want to take any chances. Through my friends I got to know the usual places he hangs out*

*in office. I found the office gym is his favorite spot. I stalked him around the gym for several days. Chatted with him causally, got to know him. And by the time the interview came around, he knew me and it was comfortable, I got the project. It was a sure shot".*

For employees, benching is a time of extreme uncertainty. Once they go on bench, employees are never certain whether they will have their jobs in the near future. Unless they get an alternate project allocation, they may lose their job. Some respondents admitted going to great lengths to avoid getting benched.

*According to a respondent, "In my past job, I got to know thorough networks in a rival company that it was bidding for our current project. Well ahead of time I got the news through network grapevine when they got our project. With my project gone there was no future for me in my current company because there were no other openings here in my current area of work. So I knew it was time to move before they benched me. So I started looking for jobs well ahead in advance. Now I work for another rival company".*

Under the official policy of "benching" companies can suddenly remove a worker from a project allocation, then reduce their pay, give poor ratings and even terminate them. It is common for new recruits be "benched" suddenly and unexpectedly as they begin their career. It is a normal part of the subcontracting business. Cultivating professional networks in the industry helps employees cope with uncertainty and unpredictability in their careers.

*Amit was thrilled to get a campus placement in a mass recruiter but it was just before the 2001 recession hit. When he graduated, the company didn't honor the joining date, nor did it give him the offer letter. Calls to the company revealed more uncertainly. He lived in limbo for two years until the recession lifted. "Each time I called they would*

*just say, please wait for some more time. I didn't know if I still had my job or I should look for another job elsewhere. And I spent 2 years like this, living in complete uncertainty about my career. I didn't get paid the whole time".*

Job uncertainty gets worse during times of global recession and downturns. Companies may suddenly suspend job offers given out to fresh recruits indefinitely. Some others may rescind new job offers made to fresh hires. It is a precarious time for fresh graduates hoping to start their software careers.

**Performance Ratings, Oppressive Work Pressure**

At the entry level, salary based incentives are the main forms of employee rewards. These rewards are operationalized in terms of variable pay. It is a portion of salary that varies by individual employee performance. Variable pay is determined by a "performance assessment" of individual employees.

*Individual performance is assessed through "Performance Rating System" or "Performance Appraisal", a tool or a metric to relatively assess a worker's performance*[1]*. It is done on a curve and relative for the team members. Each team member is awarded a relative grade (letter or numeric grade) with respective to other team members. A quarter of the team members, the highest performers are awarded the highest or top grade. The next level performers are awarded the next grades and so on. The bottom quarter of the performers are awarded the lowest grade.*

Variable pay is competitively determined through performance assessment. Higher performers get higher performance ratings and therefore make significantly higher pay than lower performers. Workers that make the highest grade get the highest variable

[1] At the time of primary field data collection (2015), all subcontract companies had performance rating systems.

pay based salary. In some mass recruiters, lowest grades might mean an actual salary deduction. Thus employees within the same team make different salaries based on individual performance. However for entry-level workers, performance ratings are also tied to some other incentives like promotion to higher levels and onsite opportunities.

*Performance assessment records play a crucial role in further promotions and mobility. Firstly, mobility into junior management- team lead, module leads and above is tied to performance assessments. Those that are consistently top rated performers tend to have a strong chance of making it into the junior management from entry level.*

*Secondly, performance assessments are tied to "onsite placements". Under onsite placements subcontracting project workers are sent to client sites (usually the US) to directly work with foreign clients. At onsite postings, employees are paid in US dollar wage rates. These salaries are several times higher than "normal" Indian salaries when they work in the same company. Onsite placements are the most lucrative and coveted incentives in the major subcontracting companies.*

*Managerial Authoritarianism, Normalizing "Overwork"*

*"During project deadlines, every one of us has to compulsorily "stretch". This means we have to give in our weekends and a typical workday can be as long as 16 hours. We are not allowed to take time off, go on leave or already sanctioned leaves may be cancelled. This is the sort of work ethic we uphold and its absolutely non negotiable. Anyone who fails to honor this requirement will lose their job".*

At project deadlines and end times employees have to mandatorily work longer hours. Many respondents said this sort of mandatory pressure to meet deadline arises every quarter.

*"Generally every 4 months we have a quarterly deadline to meet. Then we need to give 14 hours to 15 hours every day for approximately three to four weeks. We are also expected to come on weekends around that time. Then there are the project end times, about every few years when our project is about to end its lifecycle. Then this sort of crazy work pressure may last for six months. All leaves are suspended around these deadlines. No one can decide not to comply with these periodic work pressures. Anyone that refuses to comply will be given the lowest ratings and terminated".*

The performance ratings system is designed in a way so employees that get lowest ratings may be fired. Project managers are the final authority in granting performance ratings. To avoid getting the lowest ratings employees must have a good relationship with the project manager. It implies junior employees must comply with all work demands that project managers make.

*According to a senior project manager who has done performance ratings for hundreds of employees, "Managers like those who are team players, such people are valued highly and rated best. In teamwork you have to pitch in and expected to find out where your team members need you. Not just stick to your own job. If you are not seen as a team player your team will stop cooperating with you. You need them because they help you complete your work as well. The strength of your network within your own team and within your own project is a great asset. Especially this is important to your manager".*

As the final authority in granting performance ratings, project managers decide the parameters and metrics that count in the rating system. If a project manager believes employees that show willingness to take on extra work, give one's extra time, give

"more" than expected are deserving of high ratings, it becomes the project norm and work culture.

*According to a respondent, "Some managers have made clocking in longer working hours in office a "work ethic and workers below them are under pressure to comply: they feel compelled to stay back, work late just to show that they are working hard. This is the best way to be in the "managers" good graces for higher ratings, promotions and onsite opportunities". Another respondent said, "A lot of times I feel compelled to stay back and work just because others in my team are present. Then I end up staying in office even if I have no work because I can't leave. I might be judged poorly by the manager especially if he is present, it looks bad on me if I leave before my manager does. It may affect my ratings".*

Project managers set the tone of the work culture in the project. When they stay back late working, junior team members interpret it as their project managers wanting them to work longer hours and stay back after work. Employees believe that complying with these "soft forms" of pressure will earn them the manager's favor and get them high ratings.

*According to a senior manager, "If someone is allocated code development work and completes it, then that person has to know what are the other components- like say testing and go and help that team. Else talk to the client and find out what they need. So the idea is to proactively look around, find more work and do it. To be a consistently high performer means to be up to date with existing technology skills and knowing how to solve technical problems*

Many employees stay back for several hours after meeting project needs and

deadlines. They spend time talking to the client and developing a better understanding of their needs. Often it implies venturing into newer technical areas and improving their understanding of the technology market.

*I spoke with a manager who admitted to have been a highest rated performer his entire career. He has received a few "best employee" awards in his career. He personally admires those who are driven, ambitious and desire to succeed at all odds. He proudly admitted that all his waking hours are spent thinking about work and how he can continually improve his performance. He doesn't believe in having a much of personal life out of work. And extremely contemptuous of those who need to leave on time because of family or because of some hobby. "The ideal worker is the one who is completely dedicated to the company".*

Some managers deliberately push a culture where team members are expected to prioritize their employer's interests over their own personal life. This sort of discourse prioritizing the company's interests over one's own personal life projects has gained wide acceptance in India's software subcontracting work culture. Average work hours have risen- from the official 9 hours to 11 hours on an average. Work pressure increases further during project deadlines and around project end times. A worker's life is entirely controlled by the manager through performance ratings. Project managers may have created a culture of managerial authoritarianism.

*Unfair System, Toxic Competitiveness*

A serious critique of the rating system is that it's not a fair system. It is set up in a way that only a quarter can make the highest ratings. Ratings are done on curve, which means an individual's performance is relative to other team members:

*Even if the whole team performs well, only a certain fraction of the team members can make the top rating. And some people will still make the lowest grade even if they worked very hard. And regardless of how well the whole team performs or how many extra hours all team members have put in hoping for high ratings, only a few would actually make the top ratings.*

The ratings may not be a fair reflection of an individual's hard work or performance because the system fails to recognize everyone's efforts adequately. Those who do not get good ratings may feel let down and resentful especially since they may have worked very hard, logging in extra hours. Since the entire team has to compete with each other to get higher ratings, it fosters a culture of fear and suspicion instead of trust and cooperation. It creates a culture of toxic competition between fellow team members.

Secondly, it is coercive; it forces workers to give in to overwork just to remain employed. This is because *in many companies, getting the lowest ratings consistently over a certain number of assessment periods means certain termination.* So performance ratings determine who gets to remain employed because low performers are expelled.

*Babu is a director at Mapro and he has been with the company since its inception days. He was around when this rating system was adopted. And he admits that this performance-based system is not boosting employee morale. Everyone works long hours, well and beyond expectations. But everyone cannot have high ratings despite performing well and working very hard. And everyone cannot be promoted because there are not enough spots higher up i.e. the industry is pyramidal. But those that get lower ratings feel angry, resentful and uncooperative. It harms employee motivation and performance.*

Many respondents felt that the rating system is oppressive and vitiates the work culture. And managerial authoritarianism in performance assessment creates a culture of fear and suspicion

*A respondent said, "I don't have family or friends here. So I end up spending all my waking hours at my office. I stay back late at until midnight either taking on extra work, learning new technology or something like that so I can remain the highest rated performer. I am doing this so I can get an onsite placement. That way I can make a lot of money and build a house for my mother in my ancestral village. That's what keeps me in this company. But even though I have sacrificed my personal life for work, so far I haven't got any onsite opportunity. I am disappointed because it's been three years here".*

Onsite opportunities are few and limited; over the years, onsite opportunities have sharply come down. Subcontractors actively try to minimize onsite positions and maximize work done out of India[2]. But aspiring workers are not necessarily told this. They work harder hoping to get an onsite position. Software companies are constantly cutting down non-entry level workers to keep labor costs down. Upward mobility opportunities have reduced drastically over the years. Workers are spending longer times at entry level. Performance assessment system pushes workers to work longer hours for fewer rewards because rewards and incentives are being actively cut down for cost cutting measures.

**Monotony of Low Skill Subcontracting Work**

[2] For bottom lines, companies have actively reduced promotions and onsite opportunities i.e. promotions and onsite opportunities are antithetical to corporate interests. Firstly onsite opportunities imply subcontractors have to pay the worker in deputation, salaries at par with the country where they are sent. For instance, work done out of India pays $1/6^{th}$-$1/10^{th}$ of the US level salaries. That is very costly.

*Avya worked for a major software company as generic skill recruit for several years. "The level of creativity in this work was very low and intellectually dissatisfying. The job was a no brainer, very standardized and routine: I needed to code each portion of the fields if the program syntax didn't run. I also maintained a big US retail website that involved making sure each line of code was correct and in its correct position and interfacing with customer databases correctly. Often this maintenance work boiled down to pinning down minor glitches preventing the code from running correctly. But this pinning down process needs a lot of patience, knowing the technology in which it is written and troubleshooting abilities".*

Indian software subcontractors provide server and website maintenance, technical support to clients, looking for bugs in the code, troubleshooting errors, upkeep or adding some modification or feature to existing software systems for big clients.

*Typical tasks in Indian software subcontracting include: custom application development, coding, testing, maintenance (including websites, servers, legacy products) and customer support services. Major clients are MNCs ranging from banking and finance sectors like Bank of America, infrastructure like British Petroleum, American Airlines, retail companies like Starbucks and Macy's and so on.*

These jobs require a standardized set of coding, logical and problem solving skills. Working on pre existing pieces of software, employees need to be familiar with many types of technologies at the same time so they can correctly assess the technology in which a piece of software was written, and correctly apply their standardized skills for troubleshooting and code repair. This work is labor intensive, monotonous and needs a lot of patience. Many respondents complained that the work they did was monotonous

and mind numbing. There was very little scope for expressing creativity or achieving intellectual growth through this work.

*According to a respondent in a major software company, "Almost 70% of the work in Indian software subcontracting is that of testing or support roles, needing no expertise, basic aptitude is enough. In testing and product support roles, there is a set process you need to follow day in day out. The monotony of these tasks makes people fatigued and exhausted. There may be some software development work but it really boils down to making few changes to the existing system, rather than building something new. That is why even in these development projects, you don't learn a lot more than the syntax in the programming language that you have worked on. Since major development jobs are never subcontracted out to India, there is little (in these jobs) that ever challenges you".*

Doing a limited and standard set of operations like maintenance, support, testing jobs across technology specifications, scope of learning and building on creative development skills are limited. Working on a standardized set of operational skills across platforms, languages and technologies gives them superficial and basic knowledge in technology. They cannot develop in-depth skills or build software products.

*"Even after working for many years, we fail to develop "in depth knowledge or skills" in any particular technology or area". Often we work on projects someone else created but we will never be able to build it ourselves. Doing the same set of functions over and over again makes me feel so "brain dead", according to a respondent.*

Working on a tiny portion of a larger software product, subcontracting workers learn to use limited functions. Scope of expressing their creativity or in depth skill

development is limited. Without in-depth expertise in technology or the ability to build software products they cannot move into high skill engineering work.

*According to a respondent, "Once you are past the first few years, the work gets really boring because there is very little scope to learn: the depth to which you can delve into a technology is severely limited. There is no challenge, little intellectual stimulation, but you do the same thing again and again. So you feel stuck in one place. I think it is difficult to sustain productivity and interest when you hate your job so much".*

Most respondents felt no passion, little interest for the things they do at work and they felt helpless most of the time. Some openly hated their jobs and wanted to quit but couldn't because it "pays the bills".

**Unpredictability of Subcontracting Work: "Client Centered Flexibility"**

For major subcontracting companies, every project is governed by a separate and distinct set of labor control systems. Every project has a different work culture: when workers report to work, work demands, work-life balance, rules governing workplace attendance and behavior varies by projects. Work culture is not homogenous within these companies. Variations arise because clients that own these projects determine the work culture. This is "client-centered flexibility".

*"Client centered flexibility" means the subcontractor is completely responsive to clients' needs that is deemed essential for satisfying and retaining contracts. It takes many forms; ranging from custom tailoring internal organizational culture and systems to best suit each client's specific needs to honoring anything the client wants or demands, proactively anticipating what the client may need and attending to it.*

In software subcontracting companies the foreign clients is “ever present” (O’Riain 2002, 2010), though geographically dispersed. The nature of software work requires that subcontracting workers frequently and continually interact with foreign clients. That is why mass recruiters allow client- owners of individual subcontracting projects to significantly shape the work culture according to their needs and convenience.

Under client centered flexibility, individual workers operate under two distinct “layers” of control: one from the client that owns the project therefore variable by projects, and another from the employer subcontractor that is relatively more standardized.

Most projects have a “client manager”, main liaison between the client and subcontractor, so clients have more control over the project. All project workers are expected to report to this client manager. Workers operate under a dual reporting structure:

*The client manager’s decisions about all aspects of the project are final and over ride all other decisions. Therefore the client manager is the ultimate controller and owner of the project, including reporting and control systems applied to its subcontract workers.*

Generally the client manager decides the timing of reporting and attendance for a project, often based on time zones. The day begins with the foreign client meetings.

*In Arijit’s mass recruiter company a typical workday begins with a daily standup meeting with the US client and the Indian team managing the project. The meeting takes place at 8:30 am daily. Arijit and his project mates have to be there for it first thing in the morning, so they can take all the inputs and briefly report the previous days’ work. In the*

*evening they have another meeting at around 4:30 pm where all workers across teams meet together to discuss technical issues, progress made and trouble shoot. Every Wednesday they have a detailed foreign client meeting where they discuss all progress made and future time lines for project deliverables. On days with foreign client meetings, it is mandatory for everyone to dress formally. And it is mandatory for everyone to attend this meeting, without exceptions.*

Mass recruiters do not have uniform start times when workers report to work. Especially in the case of US based clients, workdays tend to "stretch" to accommodate the 12-hour average time zone difference. *Since the workday for US client is nighttime in India, subcontracting employees* routinely extend their workday to accommodate client meetings- time that is unpaid unless they were specifically hired for the "night shift"[3].

Attendance expectations for workers thus vary by individual projects. For each new project, workers have to flexibly adjust to a new system. But there are standardized reporting structures as well: all workers are officially hired to work nine hours a day as standardized company policy. All employees have to mandatorily clock in the official nine hours at the minimum.

*To keep track of employees, most companies follow a strict policy of electronically recording workers' in and out time. All workers are assigned corporate ID badges that gives them access and entry to their workplaces. All employees have to carry these badges whenever they are in office. Some companies also monitor workers' lunch breaks, making it mandatory for workers to punch in all their movements.*

[3] In some mass recruiters, for technical support subcontracting projects there is a separate night shift hiring. Workers are specifically hired to work through the whole night to accommodate the US client.

Similarly most companies have standardized monitoring of employees' electronic activity, including the websites they browse, how much time they spend at the terminal, the nature of their electronic activities during office hours and so on.

But this monitoring and control varies by projects. Based on foreign client requirements, the degree of control and surveillance in projects may be reduced or intensified. For instance, in some projects surveillance and monitoring is much more severe than others:

"*There was one project for a health care provider where we had access to individual health records of US citizens. Given its sensitive nature, our company completely "sanitized" it. We were not allowed to use the internal corporate chat and server system that we normally use in all other projects as a way to talk to other co-workers in other parts of the company. For this one we used a separate server and only those of us allocated to this project could talk to each other. This made technical discussions outside the project difficult. All of us project workers were given special project badges and we had to swipe these badges to gain access to computers allocated to the project. We had to carry these badges around at all times. The project computers were housed in a separate enclosed area making it off-limits and inaccessible to outsiders. Getting out of that room and taking breaks was more difficult than normally otherwise. Except for electronic activity pertaining to the project, all other electronic activity was forbidden and strictly monitored. The surveillance and control over our movement was overwhelming. I have never experienced this level of secrecy or security for any project", said a manager with 12 years of work experience in the industry.*

Workers are required to flexibly adapt to dual surveillance systems whenever

needed. Though many are opposed to these "sanitized" projects and excessive control, some projects have lower levels of monitoring. In fact on some projects, workers are allowed to work remotely some days of the week. And then they can log in their hours remotely.

*According to a respondent, "Some projects allow "flexi timings" i.e. workers can report to work at different times in the day, as long as they clock in the mandatory 9 hours. It is so much easier to work in these projects and attend to personal needs at the same time. There are some other projects that allow, "work from home options" a few days of the week. We are allocated laptops and we can connect from home using these laptops and get our work done. These are the best projects because they save us so many hours in commuting and beating Bangalore traffic. But these projects are few in my experience".*

When the client is very large and therefore more valuable, subcontractors willingly tailor and customize many more internal systems to suit their needs. Their "internal management systems" mimic the client to make them feel more comfortable. It is called "seamless integration". The extent to which a subcontracting company "bends backwards" for "client centeredness" depends upon its size and importance. Clients that bring in bigger business volumes are given greater flexibility and decision-making.

*For instance in early 2000s when -Nortel Networks became the largest customer of Wipro, it adopted several internal management systems mimicking this client. Their entire personnel management system and hierarchies were also borrowed from this client. This way processes between the client and subcontractor integrate better and left*

*less room for confusion in terminologies. And the subcontractor could serve this client better.*

But for an individual employee, every change in project may mean flexibly adjusting to new systems and new ways of working. When employees switch from one project to the next, there is tremendous uncertainty regarding the work pressure and control they will encounter. They vary by projects and workers have to flexibly adjust to these demands.

*Clients Determine Overwork*

The most significant client-centered flexibility lies in the demands made on a worker's time. Projects vary by workload and compulsory overtime demands.

*Girish has been working with a major mass recruiter for the last 10 years and he has worked on 4 different foreign client projects. He is no stranger to giving in extra work hours during project completion times. It is normal. But there was one particular project where the client was exceptionally demanding. He was allocated to this project about 6 months before its end so he knew it would be stressful. But the client expected to be assisted on very short notice and the whole team was on their toes the whole time. They worked on the project for 7 days a week 14 hours a day. He became sick and exhausted once the project ended. Of all the foreign clients he has served, this one was the worst- "like a nightmare".*

Based on the project they are allocated, workers have to be flexible in putting extra work hours whenever needed. But certain projects demand much more work than others. It is especially true for projects that follow "agile methods":

*Agile methods keep pace with the changing business climate, needs and pressures*

*that the client experiences on a regular basis. For instance if a client in retail business faces a sudden change in demand for a product, they will want to reflect that change in the software we maintain or develop for them. Or if they change their mind last minute or they are not satisfied with the output and want something to be added or removed last minute, such demands can be incorporated "dynamically". This is unlike the traditional "waterfall" method where deliverables are rigidly defined.*

So "agile methods" are good for the foreign clients and in meeting their needs "dynamically", and the push for adopting agile methods is largely foreign client driven. But projects using agile methods also demand extra work from subcontractors because they may need to redo something or add extra features, taking extra time. Yet submission deadlines are rigid and non negotiable.

*Atish, an entry-level worker with 4 years of experience in a major mass recruiter was still at work until 8:00 pm on Sunday. " I and my team have been working three consecutive weekends, averaging 12-14 hours per day. Our project follows the "agile method" so the client wanted a different technical feature in my module recently. To incorporate this feature we are working around the clock because it has to be submitted along other project deliverables by the deadline next week. That's why the whole team is working so hard".*

The extra work that needs to be squeezed into the prefixed timelines because of agile methods' has to be borne by the subcontracting worker.

*If deadlines are not met over the weekday employees are expected to come in over weekends to pick up the slack. Many times it can be late nights they have to work to ensure changes are incorporated within the deadline and until the work gets done up to*

*the client's satisfaction. These decisions need to be dynamically made according to client's changing needs.*

Projects with "agile methods" push extra work hours on the subcontractor and create much unpredictability in workers' lives, requiring them to make their personal lives flexible to workplace demands. This turns their personal lives upside down.

*"If the client continually changes their mind about what they want, then we do not know well in advance how much time it will take to do a piece of work. If we have some planned events like attending family weddings or going on vacations then we may have to cancel or postpone. We are expected to make such personal sacrifices for the sake of our software career and the foreign client".*

Software subcontractors have a multitude of parallel labor control mechanisms operating "under the same roof". The multiplicity of systems is driven by clients and enables subcontractors to achieve "client centered flexibility" so they can suit client needs dynamically. But "client centered" flexibility makes individual workers' lives unpredictable.

**Surviving the Subcontracting Career**

*Ramu has worked in the software industry for many years, "I have invested in two homes over the last few years for income security. I cannot depend on my job or my salary. My company can get rid of me anytime. We don't have social security. Then what will I do? How will I pay my bills? I have been laid off once earlier at the time of recessions. It can happen again. If I lose my job suddenly I can sell my property or put it up on rent. That can keep me afloat for some time. I am also worried that with the crazy work culture in this industry, putting up with interrupted sleep while attending to foreign*

*clients, the constant stress and anxiety we have daily is destroying my mental and physical health. I can fall sick anytime. My property investments will keep me afloat and pay my medical bills whenever that happens".*

Software employees are aware of their job insecurity. Every one of them shared fears of losing their job any time. Some respondents faced a sudden layoff once in their career. Software jobs don't have provisions for social security. If employees lose their jobs they lose all income. The Indian state doesn't have safety nets for them.

Most of them tried to be prepared in advance for any eventual job loss. Those that had the means invested in property and homes. It gave them security against job loss. Others invested in the share or stock market. Some others had savings.

*According to Adheesh, "I got my previous onsite placement based on my technical skills. I bargained hard for this onsite position. Unless they sent me onsite I threatened to quit. After a three-year onsite position I was able to save enough to invest in a home. Now I have a fully paid off home. The current pay in these subcontracting companies is barely enough to survive in Bangalore. Entry-level salaries have stagnated since the last recession in 2008-10. Unless you are from the city and living at your parents' home, you cannot save much after paying for food, rent and other living expenses. If I lose my job suddenly, at least I have a home. I can save on rent. I feel a little more secure. That's why we need onsite placements".*

Getting some years of onsite positions makes it possible to invest in property and build savings. Aiming for an onsite position, many respondents tried to hone their technology skills. Onsite positions are few and they are selectively awarded. Latest

technical skills can be a bargaining tool to get an onsite position. Many respondents said that with an onsite position, they were able to invest in a home or property.

*According to a respondent, "Ever since I developed skills in SCALA and other latest tools, my market value has gone up manifold. These days I get so many job offers offering mush high salaries than being given out for other technologies. In the last performance appraisal, I threatened to quit, unless they raised my pay. So my company raised my pay significantly. They didn't want to lose me, given my latest skills. Pay is increasingly tied to the demand for your technical skills".*

Many respondents believed that if they continually reskilled in new upcoming technologies they could have higher earnings. When employees want to get raises or higher salaries, latest technical skills hold leverage. It gives them better bargaining power to negotiate better salaries. High incomes allow software employees to have a secure future. Keeping their skills up to date ensures they can hold on to their market value and continued employment.

But most entry-level subcontracting employees are not paid enough to accumulate significant savings and make property investments. For instance entry-level salaries have stagnated since 2008-10 recession. It has not kept up with inflation. Investing and saving for future security is out of reach for many entry-level employees. Only a few employees are able to achieve this security. Such employees may be considered the "best case scenario" ones.

There might be many others that cannot keep with the reskilling demands. Neither can they move up into managerial roles. They try to hold on to their jobs as long as they

can. Once their skills become obsolete, they are forced to leave the industry. I have not included them in this research.

**Summary**

Employees of software subcontracting sector have to live with extreme uncertainty in their careers and fear of job loss. Companies hire and fire employees based on the market demand for their technology skills. Employees have to continually keep their eyes on the technology market demand and try reskilling to maintain employability. At the same time their jobs are high pressure, demanding long work hours. It is very difficult to reskill under such conditions. Employees often leave their jobs to do that. Unpredictability in their lives is worsened by the control foreign clients exert over subcontracting workers. Managerial authoritarianism in performance assessment based rewards makes their jobs precarious and promotes toxic competition. The low skill, monotonous nature of jobs makes many of them feel disinterested. Despite the insecure jobs and poor work culture in subcontracting companies, most employees continue to work for the industry. They hope to save enough and invest in property. This way they hope to secure their future.

# CHAPTER 4

## Mobility into Managerial Roles

Upward career mobility path in major subcontracting software companies implies mobility into managerial roles. Mobility into project management positions is the most common form of upward mobility in these companies. In the company hierarchy, project managers are considered middle level management. Becoming a project manager is the next (higher) step in a junior software employee's career ladder.

*Project managers are supervisors and coordinators for projects. They are paid much higher than entry-level workers. Depending on the years of experience it can be double or three times their salary. It comes with a lot of power and authority over junior project employees.*

Many entry and junior level employees in major subcontracting companies may aspire to become project managers. They need consistently high performance ratings to be promoted into managerial roles. But they also need to demonstrate significant interpersonal and communication skills.

*According to a respondent, "Project managers are generally liaison for all client centric communication on behalf of the project team. They communicate all client needs, requirements and changes to the junior level members. Project managers take care of the execution of individual projects i.e. assessing, keeping track of foreign clients needs and requirements and communicating them to junior teams. Then making sure these requirements are incorporated and delivered on time. Project managers are in charge of the day-to-day execution of the project and employee reporting structure. They are the*

*final authority in handing out junior employee's performance appraisals. All junior team members report to the project manager.*

Broad role descriptions in the corporate hierarchy show that major subcontracting companies have a clear role separation between the senior, mid level and junior level staff. Junior and entry level staff is entrusted with all the technical work of coding and testing for the project. Moving up the management hierarchy, value placed on technical skills successively decline and "people management " skills start to gain significance. Project managers at middle level management do not perform any technical coding or testing work. Their job is to manage individual project teams, liaise with the client owners of their project and coordinate with senior management. Therefore middle level management is generally called "people management".

*Project managers are liaison between senior management i.e. delivery managers- and junior workers- the actual production workers of projects. Every project has a delivery manager- the final authority and owner of the project. The delivery manager is not involved in the day-to-day technical production work in the project. They make the larger decisions for the project, mainly maintaining relationships with the client. In the organizational hierarchy, delivery manager and officials above them are the "real" power centers in these companies.*

Beyond the delivery manager, this project based management structure ends in many mass recruiters. Senior management above delivery managers becomes integrated. It takes care of large agglomerations of projects or area specific groupings of projects. In the industry jargon these groupings are called "verticals".

*For instance the main clientele of Indian software subcontracting is the global banking, finance and insurance sectors. These three sectors combined together form the BFSI "vertical" in industry jargon. It is the biggest segment of software subcontracting, contributing about 40 percent of all software subcontracting work. All projects in the BFSI area are grouped together under the BFSI vertical. Similarly other important but smaller verticals are retail, manufacturing and infrastructure respectively.*

Each vertical is separately managed by a set of directors: BFSI vertical directors manage all projects from clients in banking, finance and insurance sectors and so on. They do not oversee individual projects workings in their verticals. They may not be involved with individual clients. Delivery and project managers do that.

*According to a Vice President of a mass recruiter, "I manage the testing and quality assurance portion of my company. I keep track of the broader corporate concerns relevant to testing work like how to best maintain our current client demands, what kinds of testing business we can get in future, arrange our staff accordingly, possibilities of growth areas and so on. Of course my concern is immediate bottom lines but its not that simple. For instance we can sacrifice current bottom lines for future growth. These are strategic decisions we make. And we communicate these decisions to the rest of our company".*

In major subcontracting companies, senior management (beyond delivery manager) is not involved in specificities of projects, technologies, technical aspects or individual projects. It focuses on broader strategic and visionary decision-making for the company.

**Mobility into Middle Level Management**

*According to a respondent, " I have noticed that over the decades, mobility rate into managerial positions have significantly come down. When I joined the industry in the early 2000s, ratio of managers to workers was much higher. It was much easier to become a manager. One needed to maintain consistently high ratings and demonstrate good interpersonal skills. But I don't think it's so easy any more. Companies have figured out ways to have fewer managers for a much bigger number of junior employees".*

Major software companies prefer to keep employees at entry and junior levels because they are the least paid. Managers do not perform production work and they are much more expensive. Major software subcontractors like to cut down on managerial positions to keep costs down. This reduces opportunities for upward mobility.

*A respondent observed, "In most major subcontracting companies we can assume that all management positions would make up between ten to fifteen percent of the total employee strength. This includes middle and senior level management positions.*

Managerial positions are few. A small percentage of entry and junior level employees can become managers. Subcontracting companies are extremely selective in deciding whom to promote.

*According to a project manager, " In most major software companies I have worked, I have seen that after working for several years in the company one can get promoted into a project management role eventually. But it is common for a senior employee to be passed over for promotion into a managerial role. In software companies promotions are not automatically given based on seniority and years of work. You have to show you are a performer and a good fit for the role. Having consistently top ratings*

*throughout one's career is an important way to show that. But there are many other factors that go into it. Like showing you have leadership qualities, interpersonal skills, ability to network and so on. There is not fixed, predictable path or career ladder laid out there".*

Indian software subcontracting companies do not have seniority based promotion or upward mobility career paths. Mobility into middle level management is determined based on one's individual abilities. There is no standardization. Upward mobility paths are varied. Employees have to show they are a good fit for the role.

*According to Deepak-head of customer relationships for his company. "I always wanted to become a manager. In my experience, to be successful in these roles you have to take a lot of initiative that gets you noticed and talked about. In my case I focused on learning a range of skills. When something goes wrong or a process doesn't work in any area relevant to my employer, I can solve it. That's my strength. Which also means I know a lot of people in a range of areas- corporate lawyers, high engineering skill people, people in other subcontracting companies and so on. So I know whom to approach whenever a problem comes up and solve it efficiently. Ability to network is another strength. To be a successful networker you need to be the sort of person that attends networking events outside of your company- like industry conferences, workshops and seminars. Here you meet people, get to know what the competitors are doing, where the business is headed, what the future holds. Based on these insights you can take leadership initiatives in your department. You need to be an initiative taker. You need to come across as outgoing, expressive and articulate. Networks help you in your career. I got this job through a manager I met at a conference. We spent three days together as a*

*team. He liked my problem solving abilities and perspectives, therefore offered me a job at the end of the day".*

Most respondents stressed on networking, communication and problem solving skills for mobility into managerial roles. These abilities demonstrate leadership qualities. These skills require several years of "on the job" work experience to build. So seniority and years of experience spent working in the industry does play a role here. Managerial roles are rarely given to entry-level employees.

*After 12 years working in a major subcontracting company, Mayank was finally promoted to project manager, a position others would get within 8 years at that time. "When I came in, I was very tongue tied, shy and self-conscious of my limited communication, self-expression skills and ability to speak in English. These things are necessary to get noticed and promoted in the company. So I made up by being the "grunt worker". I had a mind numbing testing project. And I took up things or work in the project that others didn't want to do, working longer hours than any one else in my team or project. But I would still see others around me get promoted, or take credit for the work I did and move up as a result. Many others in my team left for better opportunities. But I slogged my way through silently, never complaining. Eventually I came to be seen as the most dependable, reliable person. At the same time I kept working on my English and communication skills. My client really trusted me. Once the project manager quit, I got my promotion and onsite opportunity that was long due. Even now I am connected to the client 24 hours 7 days a week, taking calls and answering emails for any issue that comes up after I come back home. Most days my sleep is disturbed with the US client calling me at the middle of the night. It's a support project and the client needs help*

*during their regular work hours in the US. This is how I show persistence and commitment to my client. It got me noticed. I believe that is why I got promoted into this managerial role".*

Mayank's upward mobility path shows that seniority based promotion within the same company can be one way to move into managerial roles. Candidates need to demonstrate outstanding dependability and commitment to the job. Being liked by the client is a big plus.

*According to Ash, a manager with major software company, "If you want to become a manager, client facing roles add brownie points to your CV. You get these roles after some years of experience. Going onsite and working directly with the client is an asset. To your potential employer it signals ability to communicate, lead, intervene and manage people relationships. "*

To be considered for managerial roles, candidates need to know how to work with clients. Ability to build close relationships with the client is considered an important part of a manager's job.

*Ganesh is considered an exceptionally high achiever in his career. Within 3 years of joining his first job at a mass recruiter he became a project manager. Its remarkable since the normal time frame then was 8-10 years to get there. "I was a consistent high achiever at work getting the highest performance appraisals throughout entry level. Then after 2.5 years at entry level I was made a team lead, because the previous team lead quit. And I was sent onsite to the US. So it was a stroke of luck. There I got along really well with the client manager. And I found out that there were some processes that they could do much more cheaply. I presented my own ideas to them and they found that*

*implementing my ideas would save them 40 percent of the costs. The client was pleased and gave me the project. So I was naturally made the manager for that project".*

Ganesh became a manager very quickly because he demonstrated the ability to work with clients and showed salesmanship abilities. Employees that are able to get new business demonstrate salesmanship. Subcontracting companies highly value and reward such employees.

*Janani is a successful customer relationship manager at major MNC. She joined after a few years of coding work in a subcontracting software company and then getting her MBA. She manages one large client with an army of technicians. Her job is to provide constant support to a product at the client end. She was able to move into this role because she was able to sell to this client to this client. It earned her the job.*

Some managers are able to move into these roles after several years of experience in the industry, followed by an MBA degree. Many respondents believe an MBA degree makes it easier to become a manager. At times MBA degrees from the IIMs or similar prestigious universities can be a significant advantage. MBA degrees from IIMs along with some years of experience can earn a direct entry into "Business Development" teams that manage sales and customer relationships.

*"Business development teams" are carefully selected for being savvy communicators. Together with Delivery Managers, they bring in new business and build contacts with potential clients. Scouting for new clients, they often go for foreign trips to meet and interact with foreign MNC executives. They do not do technical coding work but have thorough knowledge in this area. They are considered middle management".*

Many projects have these separate business development teams that take care of customer/client relationships. They report to the Delivery Managers. They are independent of the individual project management based hierarchy.

*Amar has worked in three companies over his seven-year career in the industry. He has risen up quickly into middle level management into the business development team. "Early on I realized that to be successful in this industry you have to always look for what is hot in the market and keep skilling in that. Even when things are going smoothly in your current job don't ever become comfortable or complacent with your career. I learnt early on that your employer is never going to be loyal to you. Whenever its necessary they will get rid you without any remorse, for them you are entirely dispensable. So don't bother being loyal to your company, instead be loyal to your job, your skill and abilities. Every couple of years I would go for job interviews "dip my toes into the water". So I know what's hot in the market and I am always up to date with the latest skill sets. This way I also get to know what's my worth in the market, whether I am getting paid my actual market value or not. If the market is paying 20 percent more I would definitely change my job. And if the job involves directly working with clients, I found it's always more paid. Now I am into client based sales work. I enjoy that very much. In the subcontracting world, "client is god" because that's where the money flows. Anyone close to the client or can influence the client will always be valued in the market".*

Rather than "sticking with" the same company for too long, respondents favored changing jobs periodically as a sound career mobility strategy. An individual employee's career path may not be limited to a single company. It's common for individual career

paths to span several companies. Employees know their jobs are insecure. They don't trust their employers will retain them in the long run. They change companies periodically hoping to negotiate a better "deal" in the next company. Continuing to work with the same company for many years i.e. loyalty to the company is not a part of the employee value system.

*Arijit was an exceptional hard worker through his career. In his first job he worked so hard the first six months, he fell sick and had to be hospitalized. His doctor told him to stop working that way and stop night shifts. But through his dedication he became the "master" at a new technology that few people were familiar with and got him noticed for promotion. "The first two years, I didn't care how much I was getting paid or what opportunities I was getting. I just wanted to achieve a level of perfection and competence at this area and technology I was assigned. At that time this technology area was just starting to emerge. Few people knew about it in my project. I ended by taking responsibility for all modules and learnt how to successfully customize them. At that time few people had that knowledge. People in my project started looking up to me as I took in even more responsibility. I ended up becoming the "go to person" for my project. I got the highest ratings throughout the period. I was promoted to be the team lead. Then I successfully bargained hard for an onsite opportunity and got it. Here I learnt the highly valued customer-facing role. And I earned a lot of money. Once I came back I wanted to get a higher managerial role. But I realized it wouldn't happen in my current company. They didn't have such openings for me. I quit, used my experience to move into another company as middle level manager. This MNC was just starting to set up its subcontracting operations in India".*

Many respondents stressed the importance of periodically changing their jobs to steadily move up in their careers. They also stressed on the ability to get noticed by other managers as an important strategy. Honing latest skills and technical abilities is a preferred way to get noticed.

*Arpit said, " When I worked in a major subcontracting company, we used to have these major networking parties every few months. Our company organized these networking parties. Here one could freely meet with the delivery manager and other "big shots" of our company. It is otherwise difficult to meet such higher ups. Many junior project employees tried to develop relationships with the delivery manager. A lot of us believe that having personal relationships with the delivery manager enables onsite positions and management promotions. Delivery managers are the "owners" of the project. They are the final authority in these decisions."*

All respondents in middle level management began their career at entry level. Many were hired through campus recruitment. Working in the industry for over several years, they were able to move up into middle level management eventually. Most of them didn't get additional MBA degrees to move up. On the job experience and skills were enough. It shows mobility into middle level management is possible through promotions up the corporate hierarchy.

**Mobility into Senior Level Management**

Mobility into senior management requires a specific type of communication and networking skill for visionary leadership. It is focused on growing the organization's business, future market and client base.

*According to a delivery manager, "I think at the senior management level, one needs a very specific type of communication and networking ability: decisive selling based on deep, strategic networking. Therefore identifying beforehand who can be a potential client, make networks with the right authority or point person in the potential client company beforehand, being ready when the client is about to ask for bids and quoting the lowest bid so you can get it are some of the strategic networking and decisive selling skills needed to be successful at my position. A lot of times these client relationships can be more decisive in "snagging" a contract than the lowest bid because say two rivals have comparably low bids, the contract will be awarded to the one that has better relationship with the client point person".*

In case of the software subcontracting business knowing the right "point persons" in the potential client company and influencing them is perhaps the most important component in strategic networking. Ability to "break into" the highly opaque client related information and secretive bidding process determines who gets to stay in business. Bids and contracts are won through this information. So it is very crucial.

But this information is opaque because point persons are not openly advertised or openly available to outsiders. Unless one has deep networks and familiarity with the client its very difficult to make such inroads. One needs "insider" networks and information to break into it.

Early studies on Indian software industry's expansion found that Indian diaspora *as former employees of many present day foreign subcontracting clients brought much of the software subcontracting business to India.* Silicon Valley based employees of Indian origin started many software-subcontracting companies. This is observed by Saxenian

(2006) that IBM's India operations was set up by Chief Technology Officer of IBM in the US. Similarly, before founding India's software subcontracting company Infosys, Narayana Murthy worked in the US Silicon Valley for many years. Using his professional connections in Silicon Valley he got his subcontracting clients.

*Paras recalled in early 2000s, his small software company had only one project from a US based telecom giant. "The owner was an Indian origin, long time former employee of this client for ten years. Working there for so long he had a thorough understanding of what kinds work could be done out of India so the company could save money. He used this knowledge and his influence and contacts to convince his employer to bundle up a set of tasks for a fixed sum that would get done for a fraction of the cost. And he was the personal guarantor making sure he would deliver. So he set up our small company, hired us to do the work and he was the owner of the project. It was a multi-million-dollar contract and he pocketed most of it I believe. He could make us work very cheaply in India. And he really made us work very hard as well. But it was a win-win situation for him and his company".*

Using their existing industry networks, former employees convinced their employers to become their clients. They were personal guarantors for initial subcontracting projects. Many body shops and subcontracting companies began this way.

Early writings between late 1990s and early 2000s on Indian software industry show these sorts of companies dominated the software outsourcing landscape. As these small companies started to scale up, eventually becoming mass recruiters, the initial founders and key people involved in bringing in business grew with their companies and businesses.

*Suresh, currently a VP with a major software company joined as one of the earliest employees. "When I joined the company had less than 500 employees. We were just starting and we didn't even have all these project-based hierarchies. I went to North America. I knew some people personally in one major MNC. They were interested to try outsourcing some of their software testing work. And I got a project, our first multi million-dollar project. And we suddenly expanded to 5000 employees because of that one project. Later we got many more projects from that single client. And now we have expanded to over 100,000 employees. Then we didn't have all these designations but I have been heading this division since then.*

All the senior management I interviewed joined before early 2000s, at the inception when the industry was starting to expand rapidly. They grew as business started coming in and eventually started heading units that grew under their "tutelage".

*Manisha graduated engineering and sensing she didn't like coding work, got an MBA from the premier IIMs. Using her alumni networks she found a job in the burgeoning software subcontracting business development team. "Those days the industry was in a massive expansion mode. I got deputed to the US so I could scout around for new business opportunities. Frankly there was a lot of interest in outsourcing and I was able to snag a big client for my company. That client gave us a multi million-dollar project and that became my project and I became the delivery manager. Most of my time was spent traveling between the US and India because I was the main liaison between my company and the client".*

Business development teams are independent of the project management hierarchy. Those that joined these teams when the industry was on an expansion mode

were very successful in bringing new business. They were able to move up into senior management.

*After his degree from IITs, Dinesh joined an engineering skill company for technologically cutting edge work. Initially he worked in sales and once his company transitioned into lower end software subcontracting work, he moved from engineering skill work into subcontracting sales work. Because of this transition, many people quit. But the company heads preferred in-house employees with sales experience. So he got the job as a delivery manager. He admitted that because of his previous sales work, he knew several people in the senior management and they knew he was a good fit for the role even through he hadn't technically brought in a subcontracting client. "I guess I was in the right place at the right time and got the job. Things worked in my favor".*

All respondents in senior management didn't really make it through promotions up the project hierarchy. Along with relationship and networking skills they seemed to be there in the "right" time and the right place. I didn't come across any instance of senior managers that began at entry or junior level and worked their way up into senior management via the project hierarchy. It is not clear if such an upward mobility path possible in case of senior management level. But given the small size of my respondents I am not sure I am not sure I can draw this conclusion. I interviewed a total of seven senior management respondents across major software companies: two Vice Presidents, two directors, and three delivery managers.

**Summary**

Mobility into managerial roles in the next step in a software subcontracting employee's career and professional growth. In corporate hierarchy, managers are middle

and senior management. Mobility into middle level management roles is through incremental promotions. But these mobility opportunities are very few and selectively awarded. Mobility paths are not standardized. Ability to move into middle level management depends upon individualistic strategies. Mobility into senior management role is rare.

# CHAPTER 5

## Mobility into High Value, High Skill Software Work

*High Value Software Subsidiaries*

*A major high technology IBM's India based software development division helps with its in-house software production. This MNC developed many software products and packages in the market and its India division helps with the production process and technical upkeep of these products. Technical upkeep and support also involves providing existing product customization or creating upgrades and patches. Once sold, each product brings in millions of dollars in revenue.*

*Similarly Ericsson has situated some of its software development work (for its mobile telephony products) in India. The Indian team provides patches and upgrades to its existing products. A similar case is of Microsoft: its India based software operations are very small but this team provides design upgrades and improvements in its existing Windows operating system.*

Many technology MNCs set up subsidiaries in India to collaboratively develop software products with Indian software engineers. Well known names are: Microsoft, Intel, Google, EMC2, Ericsson and IBM. Their software products are considered "cutting edge". Each product fetches millions of dollars; this work is considered high value. Their high value products are generally meant for sale in the developed country markets. India's domestic market for high technology products is small or underdeveloped. It cannot afford these products. These technology companies operate on the technically sophisticated road.

In the Indian software industry jargon, these subsidiaries are called “product” companies; their business model is to develop high value technology products. This term distinguishes them from the “services” model of software subcontracting companies. Therefore “product” companies operate in a different market segment.

*Case of IBM*

*IBM is a well known US based high technology MNC. IBM has three main units in India:*

*1) IBM Research: IBM Research in India collaborates with foreign teams to come up with new ideas for the next generation of high technology products. It involves making new design prototypes and developing new types of technologies. It designs many kinds of prototypes. Many new ideas and prototypes may not be always commercially viable, but whenever one is deemed saleable, it is sent for production and customization. If the product does well in the market, it translates into many million dollars of profit. In India this unit is tiny, barely employing a few hundred people. On the software value chain this work is considered highest value. Very little design and research work happens in India. It is done at the parent company headquarters in the US.*

*2) IBM Software Labs: It was set up so Indian teams could collaborate with US based teams to develop cutting edge software products for commercial production and customization. Mongo DB, a popular and well-known database product is an outcome of this collaborate effort. Once the prototype was handed to the IBM’s software labs in India, teams worked on different parts of the product, developing features, adding different aspects, making sure it runs, customizing for different settings and contexts. This work is considered high value on the software value chain. Most of the high value work*

*done in India operates at this level. IBM's software development subsidiary employs several thousand people in India.*

*3) IBM Global Business Services: This is the subcontracting services unit of IBM. This unit takes software support and maintenance projects from external clients. This work is considered "post production"- involving maintenance and upkeep services. Tasks are routine and standardized. Many India based product subsidiaries have close subcontracting relationships with India based subcontractors like IBM- services. They "subcontract" out many parts of their modular and routine jobs to save money and time. On the software value chain, this work occupies the lowest position i.e. lowest in value. It employs over 100,000 employees in India.*

The patterns of structural variations visible across the three units of IBM somewhat maps on to the larger structural variations visible in India's software industry. A majority the industry (approximately 80 percent to 90 percent or more) employee strength is located at the lower end-subcontracting sector. It includes managers in the subcontracting sector. Higher end product development and technical customization accounts for the remaining employee strength. Research and development work in India is negligible.

*High "Engineering" Skill Work*

*Explaining the difference in skill and abilities needed to work in the high value product sector and the low value subcontracting sector, a senior respondent said, "These subcontracting companies are like mechanics that repair your car or a plumber that fixes your faucets. They need to know enough about how it works so they can diagnose and fix problems. Their skill sets need to be generic across many models and makes. However a*

*mechanic does not design a car like an automobile engineer does, just as a plumber is not expected to design your home water piping system. In the same manner, software-subcontracting employees are not trained to design and develop products. Top-notch engineers can perform that sort of work. So we are talking about two different kinds of technical requirements and abilities.*

To work in the high skill "product" development sector one needs highly specialized software engineering, design and development skills. Employees need to have in-depth knowledge in their area of expertise. It is different from the relatively "generic", superficial nature of software skills needed to work in the subcontracting sector. Much like the mechanics subcontracting employees "fix", "repair" and "maintain" existing software products. They do not develop them.

**Irreplaceable, Highly Valued Employees**

*Starting, entry-level salaries in these companies are three to five times higher than jobs in software subcontracting. And they may provide additional generous benefits like stock options. With some years of experience, employees may be paid several times more than employees in the subcontracting sector with similar years of experience.*

Most of these companies pay very high salaries to these employees. They are the highest paid employees in the software industry in India. Labor policies in these companies are starkly different from that of software subcontracting.

*When Mohit's division of 200 employees suddenly shut down, his company informed them almost a year in advance about the impending layoff so employees had enough time to find alternate arrangements. And they also paid a handsome severance package- 1.5 month's full pay for every year worked in the company. His biggest stress*

*was finding the next job. In India there are few companies working in his area. So his re-employment options were severely limited. The company even got in touch with other rivals in the same area and connected these employees for job opportunities. It ensured he got an alternate job at a rival company. Mohit got a package of almost 8 months extra pay as severance. Though overall stressful, Mohit was satisfied that the company did everything in its power to smoothen the transition.*

Clearly these high skill employees are highly valued. When such employees are " laid off", they are treated respectfully. Many have generous severance packages and they try to smoothen the exit, including assisting laid-off with finding the next job. Given the high pay and respectful treatment, these employees are considered the "elites" of India's software industry. It is because of the "niche" specialized nature of their skills.

*Dinesh has worked in digital signal processing for over 17 years as an engineer and now he is an architect. This is a very specific or niche area of work within mobile telephony. So he specializes in wireless communication technology used in mobile networks and achieved mastery over specificities of 2G, 3G and 4G mobile technology, GSM and WCDMA technologies. But most of his life he has coded in C++ building DSP (digital signal processing) architecture.*

Dinesh's skills are in a specific area within mobile telephony, his domain specialization. It has taken him close to 2 decades, including college to build these skills. Employees in this skill track spend a lot of their time at work learning different aspects of the same technology or area of their work. This is how they can develop new areas in a particular technology and design novel products. Most of this training happens "on the job".

*When he was hired from the IITs, in his first job Dinesh spent close to a year attending training sessions, conferences and workshops on company time. His company was developing a product in some years and he was hired in the early stages. He recalls having the time and liberty to attend academic lectures given locally in the city and spending hours discussing technology with several professors. This helped develop his understanding of the product he was tasked with developing. He recalls the thrill and exhilarating experience of building something "novel" and gaining expertise in an area.*

Employees hold a specific set of skills applicable to that company's needs because in-depth domain knowledge is crucial to their work. These skills and expertise are built over a period of time. Given the long time, intensive training and abilities needed, supply of such "niche" skill people is very low. They are hard to find. Domain knowledge in one specific technology is not easily migrated to another one. Nor are in-depth technical skills easily transferable. These employees are irreplaceable. Since they have high technical skills, they are also highly regarded as "experts" of the technical world or "tech wizards". They are the "creative professionals" of India's software industry.

**India's Creative Professionals**

*At the time of interview Smita had just finished helping her team of 15 members design and develop a multi-million dollar storage product for a finance company. The product, including all its architecture was entirely designed and developed in house by the India unit. It was worth several million dollars. For the whole company it was a big achievement. "It was a "shining project" that got us noticed by senior management including senior management in the parent country head quarters. All of us got extra*

*bonuses and raises. More senior members of the team got promoted to higher positions. I got into this project because of a mentor in senior management".*

A small team can bring in millions of dollars of revenue through product sales. In such teams, contribution of each employee to the company bottom line is very high; each employee's "value added" is high. Companies try hard to retain these employees. They treat them well. Given their small employee sizes, interaction between members of senior management and junior employees is very high. Together they collaborate to develop new products and ideas.

*Working for an MNC, Vandana helped developed a popular and well-known database product. "It was a multi country based team effort. And now this product is a pretty well known name in global the technical community. I was selected in this team because of my database related coding expertise and logical skills. This multi country design effort came to us once the initial design and architecture was laid out. My inputs helped correct many deficiencies initial prototype had. And I helped correct a lot of bugs and errors throughout. After this I got promoted to team lead position. And I also got a special bonus".*

Highly skilled employees are hired to create products that command high value in the market. At times the products individual employee teams develop become market successes. When an employee's efforts translate into major sales and profits for these companies, such employees are recognized and valued. Individual employees' prospects are tied to how well their product fares in the market.

*A respondent informed that their team has many patents over products they developed in house, either collaboratively with overseas teams or singly within their*

*team. "We do have a team leader guiding our patents work, everyone of us worked on the patents. All of us have ownership over the work. Of course the team leader gets most of the credit. And he got promoted to a managerial role in record time".*

Employees in these companies have to show initiative in developing high technology skills and demonstrate the ability to innovate. One way to demonstrating one's outstanding technical abilities is through product patents, because they are granted for innovation. Product patents count as "certification of one's technological innovativeness". Many employees strive to obtain as many patents as they can. Having many patents counts in getting promoted.

*A fast upward moving manager said, " Our company follows an open door policy. You can walk into a senior management's cabin and have technology discussions with them. They welcome that. And I like doing that. They are experts and I like discussing my ideas with them. And I get newer ideas from them, better perspectives for designing new products so we can get patents. And they can see my passion, dedication. And I get recognition".*

There is a culture of openness in these companies. Interaction between senior level management and junior employees is high. Relatively flat hierarchies and small sizes of these companies enable this interaction. Some respondents said that connecting and networking with upper level management helps get ideas to drive newer projects, newer ideas for the next patent and so on. Unlike the subcontracting sector, managers in high skill subsidiaries have expertise in their product area as specialists and experts in a particular technology. Supervisory and people management work is a limited part of their

job description. It enables a collaborative work culture. Companies encourage this type of open interaction.

**Creative Work, Autonomous Work Culture**

*Kumar got hired from the IITs into a company that designs automation for cars. Presently he is assisting the team in coming up with a prototype for a Japanese car company. In the last 5 years they have achieved little by the way of a concrete product. "We come up with some idea after lots of discussions. Then we develop something for some time and then it doesn't run well. So we have to demolish it. And start over again from another new alternative route. And we keep doing this again and again. A lot of times we don't now where we are going, what is the way forward. But each time we fail, there is something we learn from the process. If nothing else, we just learn what not to do. So we move forward by way of trial and error method. It's a very slow and uncertain process. But all of us learn a lot".*

Kumar work in the highly specialized research and development area. There is a lot of uncertainty in the processes and procedures because they are yet to be standardized. There are no fixed steps to do them i.e. modularization is difficult to achieve. Without process standardization managerial control over employees is limited. It is very different from the subcontracting processes where systems and procedures are clearly laid out. Employees have to follow these procedures and complete tasks based on fixed timelines. Standardization allows managers to exert their control.

"*In our teams, all of us team members mutually decide the product deadlines- when we can deliver, how feasible that will be, taking stock of what is needed. All of this we assess together as a team. If teams are not able to meet deadlines, they have the*

*flexibility to reschedule. Our jobs are not on the line if a deadline falls through. So the atmosphere is not stressed in that way. And monitoring and supervision is limited to ensuring that once team members make their own deadlines, they can meet them", according to a respondent.*

Since each individual employee is an expert in their area, they have to work alone and figure out solutions independently. They are not tied to managers or other team members' schedules. Whenever new products need to be developed, work tends to become highly individualized.

*Many respondents said that employees can come into office and go as they like flexibly. Unless there is in-person meeting scheduled, often times their work doesn't get affected if they are not present in the office physically. So employees often work from home 3-4 days a week. It is possible because they can log into the company network (via internet) and be in touch with all the teams worldwide. Everyone is issued a laptop that they can use personally. "When I work remotely from home, I don't spend hours commuting and that is a relief. So I have much more energy and time for my personal life and family needs".*

As a result employees are able to exercise a lot of control and ownership over their work much like the "craft workers". This gives individual employees a lot of personal independence at workplace. Deadlines and timelines are not "hard" because of the unpredictable nature of work. Employees' lives cannot be easily regimented and strictly bound by managers. The constant stress and anxiety that generic track workers face in meeting daily deadlines and deliverables is missing here. Work in these companies is often immune to Taylorist principles of Scientific Method".

*Most respondents mentioned that working 8 hours a day is usually enough. No one is expected to work on weekends. All that they need to do is ensure that the work allocated to them gets completed. Mandatory daily logging of hours is usually absent. Their workplace doesn't have time tracking or such forms of "regimented" monitoring. Employees can take breaks as they like. They can come in at work any time of the day they like and leave as they like. Their lives are not regimented to time bounds.*

Since employees have a lot of freedom from managerial control and they are also difficult to replace experts, companies willingly extend greater sense of work-life balance as an effective retention tool. So work culture in these companies is much more employee friendly than elsewhere in the software industry

**Meritocratic Hiring Process**

*Candidates have to clear written entrance exams to get hired. These entrance tests are technically rigorous and difficult to clear. Candidates are tested for outstanding mathematical and logical skills. Specialist knowledge in a particular technology and platform, ability to use these skills to design software is essential.*

*Entry Level Hiring*

*Explaining the difference between entry level hiring in high "engineering skill" work and low skill subcontracting work, Deya a HR manager with major subcontracting company said, "In my subcontracting company we need "resources" to do specific kind of tasks, not highly skilled engineers with excellent technical competencies. Therefore we generally don't hire from the top tier colleges like IIT/NITs as they expect very technically challenging work and three times the pay we can offer. If we ever hire "engineers" from IITs/NITs, it's for very special, innovation centric, technical work that*

*our "resources" cannot do. So it's a very different career track from the regular (subcontracting one). Such employees will have separate appraisals, salary structure and expectations. That's why we hire from lower tier engineering colleges".*

These high skill subsidiaries are very selective about the college. At entry level they only hire from top tier IITs and NITs and similar elite engineering colleges. This hiring happens through campus placements. These companies operate in a completely different labor market.

*A senior manager at a major MNC product subsidiary explained, "At the entry level freshly out of college, candidates are yet to demonstrate their professional abilities. Other than the written technical test, we have no other way of knowing how good of a fit or able the recruit is technically. But when the recruit is from an elite college like IITs and got a degree in computer science or software engineering, we know they have a certain minimum standard of training and ability. That is why at entry-level campus placements we only hire from the IITs".*

At the time of entry level hiring, companies in this skill track expect new recruits to have pre existing training and significant coding, programming and advanced computer language skills. They are very particular about the engineering degrees of their entry-level recruits: only those with computer science or software related engineering degrees are eligible to apply. Some of them only hire those with Masters' degrees in software engineering or computer sciences related fields. Technical projects done in college have an edge as good projects indicate higher technical competence. This is unlike the low skill-subcontracting sector where companies have no preference for pre-existing coding skills or technical abilities.

*Entry-level recruits are generally expected to start working on live projects and learn on the job. They may be given a few weeks of "training" to familiarize with the project. This training is a highly individualized affair specific to the project into which the recruit is allocated i.e. every project has a different type of entry-level recruit training.*

High skill companies insist on strong coding skills at entry level hiring because fresh recruits are expected to start working right away. They are given very little time to train. Without strong software engineering and development skills it is impossible for them to do their job. Training is therefore strikingly different from the training that generic skill companies provide to their recruits: a highly standardized set of skills, imparted via several months of intensive crash course method and intended to mass-produce workers with homogenous skill sets.

At the entry level, these companies are closed to lower tier engineering college graduates. Graduating from top tier colleges like IITs can be instrumental in moving into the high "engineering" skill work. These companies are exclusive to highly meritorious students because IITs are extremely meritocratic.

*The Joint Entrance Exam (JEE) for the IITs is considered the most competitive entrance exam in the world; of the 1.3 million candidates only 10,000 are finally selected for admission.*

*High Merit, High Privilege*

*The IITs have a strong urban bias with over 50 percent of their incoming cohorts from barely 11 cities. Those from rural areas are the most disadvantaged with barely 12 percent making it into any of the IITs. These elite institutes are also places of class*

*privilege. The internal IIT audit found that 81 percent of the incoming cohorts were from extremely affluent families, earning over 450,000 rupees per annum (USD 65,000 per annum). And the most successful rates are from families where parents have either engineering or medical backgrounds.*

The IITs are generally out of reach for Indian poor and working classes. It is so rare for members of these classes to make it into the IITs that whenever that ever happens, these entrants become instant celebrities and darlings of the English media catering to India's middle class.

*In 2016 two students from a daily wage laborer's family made it through IITs and that became really big national news. They became instant celebrities of India's English Media and its top professional class audience. Their entry made news because the poor face impossible odds in making it into the IITs. It is rare for them to enter the IITs.*

The fact that these sorts of incidents become national news affirms the impossible odds the poor and working classes have to surmount to enter colleges like the IITs. The celebratory rhetoric admits that these elite colleges are out of bounds for the poor and less educated families.

Unlike the prohibitively expensive private engineering colleges (in low skill subcontracting work), IITs are state subsidized. They provide affordable education. But they are out of reach because entering the IITs is often contingent upon very expensive and exclusive test preparatory courses. These courses are administered by "coaching centers". They are a multibillion-dollar industry in India:

*IIT entrance exams called Joint Entrance Exams (JEE) have spawned IIT coaching towns like Kota- a north Indian small town with a concentration of these*

*preparatory centers. It attracts hundreds of thousands of aspirants from across the country come. They come and stay in this small town for a couple of years or more preparing for the entrance exams. Tuition fees in some of the more famed coaching centers here is more than what most Indian upper middle class families make in one year. Then there are additional boarding and lodging expenses that need to be paid.*

*Families may need to bear this training expense for years before their aspirant makes it into one of the IITs. It is common for aspirants to start preparing years ahead to crack the difficult entrance exam. Test preparations generally start in the 9$^{th}$ standard and can extend for years after high school graduation.*

Anecdotal evidence reveals some of these famed coaching centers have very high rates of success. Making it into the exclusive coaching centers can be a ticket to an IIT admission or a similar highly ranked, meritocratic engineering college. IIT admissions need significant investments in time and resources unaffordable to many.

Some respondents that made it into the IITs pointed out the importance of cultural networks in the early phases of their career: classmates in school, teachers and neighborhood acquaintances. Many respondents ascribed their own success of getting into the top tier colleges to solid foundations built in their grade schools, but also a social circle of academically successful peers.

*Rajat is a highly successful software industry employee with a 20-year career. He graduated from the top tier IITs and worked with some of the best-known prestigious companies in the industry. His parents are both well-educated professionals. He went to the best private school in his city. He attributes his academic success to his high school teachers that built his fundamentals. In his school he interacted with a peer group that*

*was very academically successful. He got to know about the IIT entrance exam from his peer group. All three of them went to IITs and eventually had very good careers. They shared notes, did group study and helped each other make it into the IITs. He felt that this mutual mentoring and enabling each other kept all of them motivated.*

In Rajat's case the middle class professional social network helped hone his career goals and means to achieve that. His first job was through IIT placement on campus. And throughout his stay at IIT he received significant mentoring from his classmates, alumni and professors. Their guidance in his life was crucial and shaped his career decisions:

*"My first job was a IIT campus placement. I had another job offer but decided to take this one up because I had heard good things about it from my networks. My professor who was also my advisor in my program recommended this company to me. He had a personal contact here, one of his friends who quit IIT and joined it at a senior level. Then some of my seniors also confirmed that this company was very good in terms of the high skill nature of work and good work culture. Through them I found out the technical details of work that this company did. This advice was crucial to me because I wanted a certain type of work culture and engineering skill intensive work. But early in your career how would you know that the company you are joining would be a good fit for your interests and career goals? At the time of hiring it is very uncertain and unclear. But through my social networks I knew what I was getting into right at the outset so I could be confident in making the right choice".*

Getting admission into highly ranked IITs is an important requirement in mobility into high skill software work. Hiring from IITs signals commitment to meritocracy, IITs

are highly meritocratic institutions. But gaining admissions into IITs is usually off limits for most Indians. Members of urban, elite professional class families make it into the IITs. Thus high skill software work is entirely filled by India's topmost privileged class.

**Limited High Skill Work Opportunities in India**

Most "product development" jobs are located in the developed world, generally US Silicon Valley. Employment options in high skill work in India are limited. After several years of building expertise, many professionals migrate to the US or other developed countries so they can continue with their career growth. Often this migration can within the same company.

*A senior director level respondent explained. "Generally in companies like ours' there is only limited scope for high skill professionals. At senior levels, the purely technical track where one can design and develop prototypes, take ownership over an entire product by developing it in-house is almost absent in India. Though it is common in the US where our company is headquartered.*

Many respondents expressed frustration that in India it is difficult to retain the most talented engineers and developers because of limited opportunities for career growth. Many of their friends and acquaintances moved out this way.

*A director in a product MNC said, " Most of the work that happens in our Indian subsidiary is assisting in product development. The foundational level design is all done at the headquarters and we fill in the rest of the technological requirements. We are never able to be the drivers and owners of any product. This is the case with most subsidiaries. I think they just don't trust us enough. But I personally knew the CTO in our headquarters in the US. I sat with him many times, convinced him to send conceptual*

*level work to India. Finally he agreed and I was able to get several million-dollar projects. And I became the technical director for this division".*

Retaining high skill professionals is harder when their companies in India close down and they are not left with more employment options. Then they prefer to migrate into the west.

*After working for four years in a cutting edge products division of an MNC, Mohit suddenly found out that his divisions was closing down. Senior management had decided to refocus company priorities onto other technology areas because the company felt the current technology didn't have much future marketability. And the product his division was building, all the hard work they put in was suddenly lost. Their division with 200 high skill engineers was about to layoff everyone in the next one year. The company didn't have any space for their skills because it had completely moved out of that technology.*

These sorts of sudden job losses are common in the high skill sector. Currently developing research products and technologies can suddenly get shelved. Entire units can suddenly shut down. This volatility arises out of "technological dynamism": Rapidly changing technologies create continuous upheaval. It emerges out of companies needing to continually invest in new technology research for futuristic product sales. Yet investing in new areas is risky because there is no guarantee the product or technology they are developing will even sell in the market.

In the US Silicon Valley this sort of unpredictability is normal. Within a matter of few years, a technology may come, bring in many jobs with it and simply die out quickly, replaced by another one. Benner (2002) found that in the Silicon Valley within 15 years

the topmost 75 high-end software corporations eased to exist- either shut down or merged with another company because of rapid technological change.

This sort of constant change and uncertainty in the high skill, high value product technology market translates into an unpredictable job market. It creates volatility, upheaval and job insecurity for the employees. In the US Silicon Valley, there are many companies working in the high skill product market. Employees can quickly transition into the next company when they are laid off. But in India it can be difficult because of the small size of the product development sector. Such employees may have little choice but to migrate out into the west.

Those that choose to remain in India after getting laid off may end up joining the software-subcontracting sector if they cannot find alternate opportunities in their area. It is relatively easy to migrate their skills there. Employment opportunities for such high skills professionals in the subcontracting sector are many.

*Though laid off engineering skill employees are also free to migrate into the software subcontracting. Respondents informed that high skill professionals that lose their jobs, at times might end up in the low skill-subcontracting sector. But it's considered a severe downward mobility, a "demotion" in career. They have to work at much lower pay than they are accustomed to, deal with a stressful, controlled work culture and most importantly, they cannot use their in-depth technical knowledge and expertise. It is extremely frustrating and limiting at the same time.*

Joining subcontracting work is a last desperate last resort for these high skill professionals. It's an undesirable option. Perhaps they may not want to admit that they moved from high skill work into subcontracting work; it might be too painful for them.

Limited opportunities in the high skill sector allow high skill subsidiaries to be extremely choosy in their recruitment and hiring process. It reduces mobility opportunities for most people wanting to get into high skill software development work.

**Summary**

The high skill software sector in India produces high value, cutting edge products for developed world markets. It is led by MNC subsidiaries to collaborate with India based high skill software engineers and developers in developing these products. Their employees work in specialized, niche technology areas. They need to show high innovation ability, autonomy and creativity at work. Their skills are in-depth, built over many years. Their hard to acquire skills makes them irreplaceable. These companies make every effort to retain their employees. They reward their employees highly. Managerial control over them is low. These jobs are some of the best jobs in India's software industry. Mobility into this area is generally restricted to highly meritocratic elite engineering colleges like Indian Institute of Technology (IIT). Admission into these colleges is generally possible for the most privileged, urban professional class of India. Given the highly specialized, niche skills needed in this sector, it is very difficult for subcontracting software employees to move into these companies even with years of experience.

# CONCLUSION

In chapters one and two I show India's privileged, professional class fills software industry jobs in India. Many are offspring of India's "old middle class"- employees of India's state sector with secure, middle class lifestyle and high status. With the relatively high pay scales in the software industry they can maintain high social status.

But in chapter three I show that their jobs are precarious. The Indian state relaxed labor laws for the software industry. Hiring and firing is common and unions are prohibited. These jobs are located at the low end of the software value chain; needing limited software skills and technical expertise. The majority of software work done in India is an outcome of global software-subcontracting arrangements. Indian subcontracting companies receive the most standardized, modular set of support and maintenance jobs from major US MNCs. Thus Indian software industry specializes in lower end, low skill work.

Before they began to subcontract out to India, foreign MNCs often hired high school graduates with coding skills to do these jobs in-house. This was possible given the limited skill requirements needed to do this work. In India however, MNCs could get these jobs done at a fraction of the labor costs. Thus global capital's labor savings is the main logic driving India's software industry.

These jobs are well paid by Indian standards. Many aspire to get into the software industry. Software corporations decided to hire overqualified people with high levels of cultural capital to fill these jobs. In chapter two I showed most companies prefer engineering degree holders, with English language proficiency and Bourdieusian "habitus", i.e. cosmopolitan identities.

The privileged class and upper caste members are able to acquire this cultural capital and habitus requirements. They use their class and caste privilege- social capital, cultural capital and affluence -to enter the industry. For instance, through college campus placements in expensive private engineering colleges, many get recruited into subcontracting software jobs. Members of working classes and disadvantaged, "dalit" or untouchable castes face considerable challenges in entering the software industry. Entry paths into the industry favors social privilege.

System administration and maintenance, technical support, testing and documentation are the main tasks subcontracted to India. This work is considered postproduction, and is located at the lower end of software value chain. These tasks are standardized and modular. Major subcontracting companies like TCS, Infosys, Wipro, IBM (Global Business Services) employing 100,000-300,000 employees each, "mass produce" these skills through short term, intensive trainee programs. After campus recruitment, trainees are sent to these programs where they acquire skills with relative ease. Software companies do not test coding or technical skills at the time of entry-level recruitment. Engineering degrees are needed to enter these software jobs as a formal "business requirement".

But ease of skill transfer makes employees replaceable. Software companies are legally allowed to hire and fire employees based on short-term market signals. In chapter three I show that to remain employable and "hold on to their jobs", employees are forced to continually train in new, rapidly changing technologies. The software industry is characterized by "technological dynamism" i.e. existing technologies rapidly become obsolete, getting replaced with newer forms of technologies. To keep up with their

training needs, employees need to keep abreast of the changing technology market and invest several months in intensive technology reskilling every few years.

But continuously reskilling and learning new technologies creates fatigue and burnout over time and the industry continually replaces senior employees with younger and cheaper labor. Employees that fail to upgrade their technical skills periodically lose their market value and become unemployable in the long run.

Working with limited labor rights, employees are powerless against managerial authoritarianism and unreasonable workplace demands. Working weekends, working extra hours (at times 12-14 hours a day) and working late nights are common features of the work culture. It creates unpredictability and compromises work-life balance. Unprotected by the state and labor unions, employees have limited bargaining power and agency.

At the same time, these jobs are monotonous, repetitive and mind numbing. Employees learn to apply software skills over a range of technologies and platforms. It is difficult to develop in depth coding or software development skills. They are not able to develop specialization in any technology. This limits mobility from lower end on the value chain into high on the value chain software work.

In chapter four I show upward mobility paths in software subcontracting generally lead to managerial positions. Managers have more power, autonomy and higher pay. They perform supervisory, "people management" work. But upward mobility opportunities into managerial jobs are limited; barely fifteen to twenty percent employees become managers. Few employees can move up into higher quality managerial jobs, and the majority of India's software employees remain stuck at the entry level.

In chapter five I show major MNCs in the technology sector set up Indian subsidiaries for collaborative software design and development--considered high up on the software value chain. These jobs allow employees to significantly express their creativity in designing new products and developing in depth specialization in technology.

Compared to the software-subcontracting sector, high-end software subsidiaries are a small proportion of Indian software industry. Opportunities in the high-end software sector are limited. These jobs need outstanding mathematical and logical skills, in-depth specialized knowledge in technology and ability to build high value software products. Learning to apply a range of technologies in repair and maintenance of existing software products "on the job", subcontracting workers rarely acquire the technically sophisticated specialized skills needed to enter the high value sector. It is difficult to move into higher-level work on the value chain.

Summing up, India's software industry shows a pattern of jobs where India's privileged find work in dead-end, poor quality jobs with limited labor rights. These jobs are dead end in the sense they provide limited upward mobility opportunities. Only a few employees can get promoted into managerial roles. These jobs are precarious in the sense employees can be fired anytime. The work environment is unpredictable as employees bear market risks. Continuous technology learning and training is essential to remain employed.

Most employees remain stuck in entry-level type technology work almost resembling "intellectual assembly lines", in precarious working conditions and

continuous pressures to reskill, in what is sometimes called the "technology treadmill". Thus most software employees are simultaneously privileged and precariat.

A much smaller portion of the software industry, comprising of research, development and design work, has a slightly different structure, generally an outcome of foreign offshoring. A few MNCs set up offshore subsidiaries in India to utilize India's top technology talent for developing software products. Through subsidiaries MNCs can retain ownership and full control over work done out of India.

To get the technical skills and outstanding mathematical and logical abilities needed for this work, they selectively hire from the most meritocratic engineering institutions. Hiring is mainly from top tier engineering colleges. Entry into these colleges is very competitive, entirely filled by children from privileged families.

Design and development work is highly creative work that needs niche technology skills and expertise. Employees often work autonomously with limited managerial control. Employees in subsidiaries work collaboratively as equals with teams in the MNCs' home countries. Employees are difficult to replace, and MNCs provide excellent working conditions and benefits to retain them, giving relatively high job security. Work conditions are some of the best in the industry. Employees in this segment have specialization and expertise in niche technological areas. Managerial control over employees is limited.

Subsidiaries and subcontracting arrangements are two very different patterns of global offshoring in Indian software industry. Subsidiaries are owned and operated by foreign MNCs. But very few MNCs have subsidiaries in India, limiting employment prospects in this segment of the software industry. Subcontracting arrangements are

fixed-term contract-based work, which MNCs get done through external agencies. Subcontracting arrangements dominate the industry jobs.

For software industry employees, the two different offshoring patterns translate into distinctly different opportunity structures and work cultures. Work in higher end of the value chain offshored through subsidiaries tends to create a few relatively secure, "good jobs". Global subcontracting arrangements create most of the Indian software jobs; but these are "poor quality", insecure and precarious. These employees are the "privileged precariat" of India's software industry.

Subcontracting arrangements are primarily driven by labor cost reduction. To remain in business, software companies indulge in "cost competition" with rivals. Through employee competitiveness and individual achievement they try to push employees to be more productive and reduce labor costs. Salaries tied to employee performance are competitively determined and variable. Poor performance may lead to salary deductions. Promotions are competitively awarded. Software companies do not follow seniority-based promotions.

The upward mobility and reward structure has uncertain, unpredictable outcomes. To get incentives and rewards, employees need to "outshine their peer" and competitively demonstrate a "portfolio" of technical skills and interpersonal abilities such as maintaining managers' favor. A select few can successfully navigate the uncertainty, and are rewarded with income "windfalls" through "onsite" deputations. A few years of onsite positions is sufficient for property and real estate purchases. Employees make these investments for long-term financial security and retirement planning; software jobs don't have provisions for retirement or employment security.

But the uncertainty around promotions and rewards breeds employee resentment. Continually trying to "outshine" their peer for fewer rewards fosters toxic competition. Employees are not incentivized to stick to the same employer. Hoping to find better prospects elsewhere, employees periodically change their jobs. A software employee's work history usually spans a succession of workplaces. This opportunity structure makes up a majority of India's software industry jobs and careers.

The character of software jobs is different from the state sector jobs held by most software employees' parents. Their parents had jobs with security, permanent tenure and predictable "cradle to grave" career paths. Employees continued with the same employer until retirement. The security and stability in these jobs were an outcome of the state's guaranteed labor protections and collectively bargained terms. But working in software jobs without state protections and collective bargaining rights, the "privileged" class offspring of former state sector employees, face lifetime careers of insecure and precarious conditions.

# BIBLIOGRAPHY


Agarwala, Rina. 2008. “Reshaping the Social Contract: Emerging Relations Between the State and Informal Sector Workers in India.” *Theory and Society* 37 (4): 375-408.

Alamgir, J. 2005. “Computer Software.” Pp. 358-364 in *Globalization: Encyclopedia of Trade, Labor, and Politics*, edited by A. Vaidya. Santa Barbara, CA: ABC-CLIO. Retrieved June 10, 2016 http://ezproxy.library.wisc.edu/login?url=http://search.credoreference.com.ezproxy.library.wisc.edu/content/entry/abcgetlp/computer_software/0

Ally Shireen. 2009. *From Servants to Workers: South African Domestic Workers and the Democratic State. Ithaca: Cornell University Press.*

Almeida, Paul. 1996. "Knowledge Sourcing by Foreign Multinationals: Patent Citation Analysis in the US Semiconductor Industry." *Strategic Management Journal* 17: 155-165.

Amable, Bruno. 2003. *The Diversity of Modern Capitalism*. Oxford: Oxford University Press.

Amin, Ash and Nigel Thrift. 1992. “Neo-Marshallian Nodes in a Global Economy.” *International Journal of Urban and Regional Research* 16 (4): 571-87.

Amsden, Alice. 1989. *Asia's Next Giant : South Korea and Late Industrialization.* New York : Oxford University Press.

Anand, Nupur. 2013. “Himanshu & Varun Aggarwal's'Aspiring Minds' Have Found Success in the Placement Business.” *Economic Times*, February 13.

Aneesh, A. 2006. *Virtual Migration: The Programming of Globalization.* Durham, NC: Duke University Press.

-------. 2009. “Global Labor: Algocratic Modes of Organization.”. *Sociological Theory 27 (4): 347-370.*

Antilov, Hans. 1999. “The New Rich and Cultural Tensions in Rural Indonesia.” Pp. 189-208 Pp. 252-275 in *Culture and Privilege in Capitalist Asia*, edited by Michael Pinches. London: Routledge.

Antras Pol and Elhanan Helpman.2008. “Contractual Frictions in Global Sourcing”. In *The Organization of Firms in a Global Economy*, edited by Elhanan Helpman, Dahlia Marin and Thierry Verdier. Cambridge, MA: Harvard University Press.

Arora, A., V.S. Arunachalam, J. Asundi and R. Fernandes. 2001. “The Indian Software Services Industry”. *Research Policy* 30 (8): 1267–87.

----------2004. “The Globalization of the Software Industry: Perspectives and Opportunities for Developed and Developing Countries.” Pp. 1-32, in *Innovation Policy and the Economy, edited by* Adam B. Jaffe, Josh Lerner and Scott Stern. Boston MA: Massachusetts Institute of Technology Press.

-------and Alfonso Gambardella. 2006. “Bridging the Gap: Conclusions.” Pp. 275-304 in *From Underdogs to Tigers: The Rise and Growth of the Software industry in Brazil, China, India, Ireland and Israel, edited by* Arora Ashish and Alfonso Gambardella. New York: Oxford University Press.

--------, Forman Chris, and Yoon Jiwoong. 2008. “Software.” Pp. 53-99 in *Global Industries: U.S. Firms Competing in a New World*, edited by J.T. Macher and D.C. Mowery. Washington D.C.: Innovation National Academies Press.

-------- and Surendrakumar, Bagde. 2011. Private Investment in Human Capital and Industrial Development: The Case of the Indian Software Industry." *Dimensions of Economic Theory and Policy Essays for Anjan Mukherji*, edited by Krishnendu Ghosh Dastidar, Hiranya Mukhopadhyay, and Uday Bhanu Sinha. New Delhi : Oxford University Press.

Arabandi, Bhavani. 2011."Globalization, Flexibility and New Workplace Culture in the United States and India." *Sociology Compass* 5 (7): 525-539.

Asundi Jayateertha. 2001. *Issues in Software Development- Outsourcing, Design and Organization.* Unpublished Dissertation. University of Pennsylvania, Pittsburg, 2001.

Athreya Suma S. 2006. "The Indian Software Industry". Pp. 7-40 in *From Underdogs to Tigers: The Rise and Growth of the Software industry in Brazil, China, India, Ireland and Israel*, edited by Arora Ashish and Alfonso Gambardella. 2006. New York: Oxford University Press.

Attwell, Paul and Sukhdeo Thorat. 2010. "Caste is not Past: The Persistence of Discrimination in India's Formal Labor Market." Pp. 243-255 in *Discrimination in An Unequal World,* edited By Miguel Angel Centeno and Katherine S. Newman

Bair, Jennifer. 2005. "Global Capitalism and Commodity Chains: Looking Back, Going Forward." *Competition and Change 9 (2):153-180.*

Bairy, Ramesh T.S. 2010. *Being Brahmin Being Modern: Exploring the Lives of Caste Today.* New Delhi: Routledge.

Baldry, Chris, Peter Bain, Phil Taylor, Jeff Hyman, Dora Scholarios, Abigail Marks, Aileen Watson, Kay Gilbert, Gregor Gall and Dirk Bunzel 2007. *The Meaning of Work in the New Economy.* Hampshire: Palgrave Macmillan.

Bardhan, Pranab. 1999. *The Political Economy of Development in India.* New Delhi: Oxford University Press.

---------2012. *Awakening Giants, Feet of Clay: Assessing the Economic Rise of China and India*. Princeton, NJ: Princeton University Press.

Basant, Rakesh and Uma Rani. 2004. "Labour Market Deepening in India's IT: An Exploratory Analysis." *Economic and Political Weekly* 39(50): 5317-5326.

Basu, Mihika. 2013. "IIT JEE Success Skewed In Favour of Urban, High-Income Students." *Financial Express,* August 8.

Baviskar, Amita and Raka Ray. 2011. "Introduction." Pp. 1-27 in *Elite and Everyman: The Cultural Politics of Indian Middle Classes,* edited by Raka Ray and Amita Baviskar. New Delhi: Routledge.

Becker Uwe. 2013. "Measuring Change of Capitalist Varieties: Reflections on Method, Illustrations from the BRICs." *New Political Economy* 18(4):503-532.

Beirne Martin, Harvie Ramsay, Androniki Panteli. 1998. "Developments in Computing Work: Control and Contradiction in the Software Labor Process." Pp. 147- in *Workplaces of the Future* edited by Paul Thompson and Chris Warhurst. London: Macmillan Business.

Bell, Daniel. 1973. *Coming of the Post Industrial Society*: *A Venture in Social Forecasting*. New York:

Basic Books.

Benner Chris. 2002. *Work in the New Economy: Flexible Labor Markets in Silicon Valley.* London: Wiley Blackwell.

Bergquist, Charles. 1986. Labor in Latin America : Comparative Essays on Chile, Argentina, Venezuela, and Colombia. Stanford, CA: Stanford University Press.

Betteile, Andre. 2001. "The Social Character of the Indian Middle Class". Pp. 73-85 in *Middle Class Values in India-Europe*, edited by Ahmed, Imtiaz, and H. Riefeld. New Delhi: Social Science Press.

Bhagwati, Jagdish. 1993. *India in Transition: Freeing the Economy*. Oxford: Oxford University Press.

Bhattacharya, Saumya. 2016. "Hottest Skills That Can Get You Hired In 2016." *Economic Times*, January 13.

Bhowmik, Sharit K. 2009. "India: Labor Sociology Searching for a Direction." *Work and Occupations* 36: 126-144.

Bhatnagar, Subhash and Robert Schware.2000.*Information and Communication Technology in Development Cases from India.* New Delhi: Sage Publications.

Biao, Xiang. 2007. *Global Body Shopping: An Indian Labor System in the Information Technology Industry*. Princeton, NJ: Princeton University Press,

Biswas, Soutik. 2017. "Is India's Middle Class Actually Poor?" *BBC News* November 15. Retrieved on April 1, 2019. https://www.bbc.com/news/world-asia-india-41264072

Biradavolu, Monica Rao. 2008. *Indian Enterpreneurs in the Silicon Valley. The Making for a Transnational Techno-Capitalist Class.* New York : Cambria Press.

Birdsall, Nancy. 2015. "Does the Rise of the Middle Class Lock in Good Government in the Developing World?" *Center for Global Development*. Retrieved on April 10, 2019. http://www.cgdev.org/publication/does-rise-middle-class-lock-good-government-developing-world

Blair-Loy, Mary.2004."Work Devotion and Work Time." Pp.282-316 in *Fighting for Time: Shifting Boundaries of Work and Social Life*, edited by Cynthia Fuchs Epstein and Arne L. Kallenberg. New York: Russell Sage Foundation.

Block, Fred. 2008. "Swimming Against the Current: The Rise of a Hidden Developmental State in the United States." *Politics & Society* 36 (2) : 169-206.

Bourdieu. P. 1984. *Distinction: A Social Critique of the Judgment of Taste.* Cambridge, MA: Harvard University Press.

--------and Loïc J.D. Wacquant.1992. *An Invitation to Reflexive Sociology.* Chicago, IL: University of Chicago Press.

Braverman, Harry. *Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century.* New York: New York University Press.

Breen, 2005. "Foundations of Neo-Weberian Class Analysis." Pp. 31-50 in *Approaches to Class Analysis*, edited by Erik O. Wright. Cambridge: Cambridge University Press.

Brooks, Frederick P. Jr. 1995. *The Mythical Man-Month Essays on Software Engineering.* Boston, MA:

Addison Wesley Longman, Inc.

Brosius, Christiane. 2012. *India's Middle Class: New Forms of Urban Leisure, Consumption and Prosperity*. New York: Routledge.

Burawoy, Michael. 1979. *Manufacturing Consent: Changes in the Labor Process Under Monopoly Capitalism.* Chicago, IL : University of Chicago Press.

Burt, Ronald S. 2001. "The Social Capital of Structural Holes." Pp. in *New Directions in Economic Sociology*, edited by Mauro F. Gullen, Randall Collins, Paula England and Marshall Meyer. New York: Russel Sage Foundation.

Cardoso, Gustavo. 2005. "Societies in Transition to the Network Society." Pp. 23-70 in *The Network Society: From Knowledge to Policy*, edited by Manuel Castells and Gustavo Cardoso. Washington DC: John Hopkins Center for Transatlantic Relations.

Castells, Manuel. 2005. "The Network Society: from Knowledge to Policy." Pp. 3-22 in *The Network Society: From Knowledge to Policy*, edited by Manuel Castells and Gustavo Cardoso. Washington DC: John Hopkins Center for Transatlantic Relations.

Centeno, Miguel Angel. 2010. "Discrimination in an Unequal World." Pp 3-22 in *Discrimination in An Unequal World*, edited By Miguel Angel Centeno and Katherine S. Newman. New York: Oxford University Press.

Chakrabarty, Roshni. 2016. "Only 7 Percent Engineering graduates Employable: What's Wrong With India's Engineers?" *India Today* July 13. http://indiatoday.intoday.in/education/story/engineering-employment-problems/1/713827.html

Chakravarty, Sukhamoy . 1987. *Development Planning: The Indian Experience.* Oxford: Clarendon Press.

Chalmers Johnson. 1982. *MITI and the Japanese Miracle: The Growth of Industrial Policy, 1925-1975.* Stanford, CA: Stanford University Press.

Chan, Talk W. and John H. Goldthorpe. 2007. "Class and Status: The Conceptual Distinction and its Empirical Relevance." *American Sociological Review* 72: 512-532.

Chandler, Alfred D. Jr. 1994. "The Competitive Performance of U.S. Industrial Enterprises Since the Second World War." *Business History Review* 68 : 1-72.

Chandrasekhar, C.P. and Jayati Ghosh. 2002. *The Market That Failed: Neoliberal Economic Reforms In India*. New Delhi: Left Word Books.

--------2012. "From Dirigisme to Neoliberalism: Aspects of the Political Economy of the Transition in India." *Development and Society* 39 (1): 29-59.

Cheng, Mariah M. And Arne L. Kalleberg. 2001. "How Permanent Was Permanent Employment ? Patterns of Organizational Mobility in Japan, 1916-1975." Pp. 171-186 in *Working in Restructured Workplaces: Challenges and New Directions for the Sociology of Work*, edited by Daniel B. Cornfield, Karen Campbell and Holly J. McCammon. Thousand Oaks, CA: Sage Publications.

Chibber, Vivek. 2003. *Locked in Place: State-Building and Capitalist Industrialization in India, 1940-1970*. Princeton, N.J. : Princeton University Press.

---------2008. "On the Decline of Class Analysis in South Asian Studies." Pp. 24-49 in *Whatever Happened to Class? Reflections from South Asia*, edited by Ronald J. Herring and Rina Agarwala. New York: Routledge.

Christopherson, Susan and Jennifer Clark. 2007. "*Remaking Regional Economies: Power, Labor and Firm Strategies In the Knowledge Economy*." New York: Routledge

Chun, Jennifer. 2009. Organizing at the Margins: The Symbolic Politics of Labor in South Korea and the United States. Ithaca: Cornell University Press.

CNN (2016), "Indians Flock to Temple in Hyderabad Believed to Help in Getting U.S. Visas". *CNN Travel*, February 26. Retrieved on April 1, 2019. http://www.cnn.com/travel/article/india-temple-visas-hindu/index.html

Collier, Stephen J. and Aihwa Ong. "Global Assemblages, Anthropological Problems". Pp. 3-28 in *Global Assemblages: Technology, Politics, and Ethics as Anthropological Problems*, edited by Aihwa Ong and Stephen J. Collier. Blackwell Publishing Limited.

Collins, Jane. 2003. *Threads : Gender, Labor, and Power in the Global Apparel Industry.* Chicago, IL: University of Chicago Press.

----------2006. "Redefining the Boundaries of Work: Apparel Workers and Community Unionism in the Global Economy." *Identities: Global studies in Culture and Power* 13(1): 9-31.

Collinson, David L. and Margaret Collinson. 2006. The Power of Time: Leadership, Management, and Gender." Pp. 219-246 in *Fighting for Time: Shifting Boundaries of Work and Social Life*, edited by Cynthia Fuchs Epstein and Arne L. Kallenberg. New York: Russell Sage Foundation.

Cornfield Daniel B, Karen Campbell and Holly J. McCammon. 2001. "Working in Restructured Workplaces: An Introduction." Pp xi-xxii in *Working in Restructured Workplaces: Challenges and New Directions for the Sociology of Work*, edited by Daniel B. Cornfield, Karen Campbell and Holly J. McCammon. Thousand Oaks, CA: Sage Publications.

Crouch, Colin. 2005. "Models of Capitalism." *New Political Economy*, 10(4): 439-456.

Damarin, Amanda Kidd. 2006. "Rethinking Occupational Structure: The Case of Web Site Production Work". *Work and Occupations* 33 (4) : 429-463.

Dayasindhu, N. 2002. "Embeddedness, Knowledge Transfer, Industry Clusters and Global Competitiveness: A Case Study of the Indian Software Industry." *Technovation* 22 (9): 551-560.

Davis, Diane .2010. "The Socio-spatial Reconfiguration of Middle Classes and Their Impact on Politics and Development in the local South: Preliminary Ideas for Future Research." *Political Power and Social Theory* 21:241-267.

D'Costa, Anthony P. 2003. "Uneven and Combined Development: Understanding India's Software Exports." *World Development* 31(1): 211-226.

---------2003. "Catching Up and Falling Behind." Pp. 44-64 in *Asia.Com: Asia Encounters the Internet*, edited by K. C. Ho, Randy Kluver and Kenneth C.C. Yan. New York: Routledge Curzon.

D'Cruz, Premilla, and Ernesto Noronha. 2006. "Being Professional Organizational Control in Indian Call Centers." *Social Science Computer Review* 24(3): 342-361.

Desan, Mathieu Hikaru. 2013. "Bourdieu, Marx, and Capital: A Critique of the Extension Model." *Sociological Theory* 31 (4): 318-342

Deshpande, Ashwini and Katherine Newman. 2007. "Where the Path Leads: The Role of Caste in Post-University Employment Expectations." *Economic and Political Weekly* 42 (41) : 4133-4140.

Deshpande Satish. 2003. *Contemporary India: A Sociological View.* New Delhi: Penguin India.

Deyo, Frederick C. 2001. "Labor and Post Fordist Restructuring in East and Southeast Asia." Pp. 409-424 in in *Working in Restructured Workplaces: Challenges and New Directions for the Sociology of Work,* edited by Daniel B. Cornfield, Karen Campbell and Holly J. McCammon. Thousand Oaks, CA: Sage Publications.

D'Mello, Marisa and Sundeep Sahay. 2007. ""I am Kind of a Nomad Where I Have to Go Places and Places''. . . Understanding Mobility, Place and Identity in Global Software Work From India." Information and Organization 17 (2007) 162–192.

---------- 2008. "Betwixt and Between? Exploring Mobilities in a Global Workplace in India." Pp. 76-100 in *In an Outpost of the Global Economy: Work and Workers in India's Information Technology Industry*, edited by Carol Upadhya and AR Vasavi. New Delhi: Sage Publications.

Eischen, Kyle. 2004. *The Limits of Rationalization: Work Practice, Industry Structure, and Spatial Organization in the Software Industry.* Unpublished Ph.D Dissertation. University of California-Santa Cruz.

Eden, Lorraine. 1991. "Bringing the Firm Back-in: Multinationals in International Political Economy." *Millennium: Journal of International Studies* 20(2) : 197-224.

Elchardus, Mark and Jessy Siongers. 2007. "Ethnocentrism, Taste and Symbolic Boundaries." Poetics 35 : 215–238.

Epstein, Cynthia Fuchs and Arne L. Kalleberg. 2004. "Time and Work: Changes and Challenges.". Pp. 1-22 in *Fighting for Time: Shifting Boundaries of Work and Social Life*, edited by Cynthia Fuchs Epstein and Arne L. Kallenberg. New York: Russell Sage Foundation.

Esping-Andersen, Gøsta. 1992. Post Industrial Class Structures: An Analytical Framework. *Estudio/Working Paper 1992/38*. Retrieved on June 3rd 2015.

------1999. *Social Foundations of Postindustrial Economies*. Oxford: Oxford University Press.

Evans, Peter B. 1995. *Embedded Autonomy: States and Industrial Transformation.* Princeton, NJ: Princeton University Press, 1995.

Feagin, Joe R., and Michael Peter Smith. 1998. "Cities and the New International Division of Labor: An Overview." Pp. 25-58 in *The New Urban Paradigm: Critical Perspectives on the City*, edited by Joe R. Feagin. Lanham, MD : Rowman & Littlefield Publishers.

Fenwick, Rudy and Mark Tausig. 2004. "The Health and Family-Social Consequences of Shift Work and Schedule Control: 1977 and 1997." Pp. 77-110 in *Fighting for Time: Shifting Boundaries of Work and Social Life*, edited by Cynthia Fuchs Epstein and Arne L. Kallenberg. New York: Russell Sage Foundation.

Fernandes, Leela. 2006. *India's New Middle Class: Democratic Politics in an Era of Reform.* Minneapolis, MN: University of Minnesota Press.

-------- and Patrick Heller. 2008. "Hegemonic Aspiration. New Middle Class Politics and India's Democracy in Comparative Perspective". Pp. 146-154 in *Whatever Happened to Class? Reflections from South Asia*, edited by Ronald J. Herring and Rina Agarwala. New York: Routledge.

Fernandez-Kelly, Karina, Penny Bamber and Gary Gereffi. 2010. "The Offshore Services Global Value

Chain". Duke Center on Globalization, Governance and Competitiveness. Economic Upgrading And Workforce Development. Retrieved on November 10, 2014.

http://www.cggc.duke.edu/pdfs/CGGCCORFO_The_Offshore_Services_Global_Value_Chain_March_1_2010.pdf

Flecker Jörg, Ulrike Papouschek and Stavros P. Gavroglou. 2006. "New Forms of Work Organisation and Flexibility in the Knowledge-Based Society". Pp. 47-64 in *Transformation of Work in a Global Knowledge Economy: Toward a Conceptual Framework*, edited by Ursula Huws. WORKS Project. Retrieved on July 30, 2013. http://www.ftu-namur.org/fichiers/WORKS-Conceptual_framework.pdf

Flemmen, Magne. 2012. "The Structure of the Upper Class: A Social Space Approach." *Sociology* 46 (6): 1039–1058

Florida, Richard. 2004. *The Rise of the Creative Class : And How It's Transforming Work, Leisure, Community and Everyday Life.* New York: Basic Books.

--------2005. *The Flight of the Creative Class : The New Global Competition for Talent.* New York: Harper Business.

--------2012. *Rise of the Creative Class, Revisited : And How It's Transforming Work, Leisure, Community and Everyday Life.* New York: Basic Books.

Frederick, Marla . 2010. "Rags to Riches: Religion, Media, and the Performance of Wealth in a Neoliberal Age." Pp. 221-237 in *Ethnographies of New Liberalism* edited by, Carol J. Greenhouse. Philadelphia, PA: University of Pennsylvania Press.

Freeman, Carla. 2000. *High-Tech and High Heels in the Global Economy: Women, Work, and Pink-Collar Identities in the Caribbean*. Durham, NC: Duke University Press.

Fröbel, Folker, Jürgen Heinrichs, and Otto Kreye. 1981. *The New International Division of Labour*. Cambridge: Cambridge University Press.

Fuller, Christopher J. 2006. "Engineering Colleges, 'Exposure' and Information Technology: Professionals in Tamil Nadu. *Economic and Political Weekly* 41 (3) : 258-288.

--------2007. "Information Technology Professionals and the New-Rich Middle Class in Chennai (Madras)." *Modern Asian Studies* 41(1): 121-150.

-------and Haripriya Narasimhan. 2008. "Empowerment and Constraint: Women, Work, and the Family in the Software Industry in Chennai." *Pp. 190-2010 In an Outpost of the Global Economy: Work and Workers in India's Information Technology Industry*, edited by Carol Upadhya and AR Vasavi. New Delhi: Sage Publications.

-------2014. *Tamil Brahmans : The Making of a Middle-Class Caste.* Chicago, IL: University of Chicago Press.

Gabriel, Richard P. 1996. *Patterns of Software Tales from the Software Community.* New York: Oxford University Press.

Ganguly- Scrase and Timothy Scrase. 2008. *Globalisation and the Middle Classes in India: The Social and Cultural Impact of Neoliberal Reforms.* New York: Routledge.

Gereffi, Gary.1993. *Global Production Systems and the Third World Development. Madison, WI: Global Studies Research Program, University of Wisconsin-Madison.*

-------1994. Global Commodity Chains: New Forms of Coordination and Control." *Pp. 95-122 in Commodity Chains and Global Capitalism* edited by Gary Gereffi and Miguel Korzeniewicz. Westport, CT: Praeger.

--------, John Humphrey and Timothy Sturgeon. 2005. "The Governance of Global Value Chains." *Review of International Political Economy* 12 (1): 78-104.

------ 2008. *The New Offshoring of Jobs and Global Development: An Overview of the Contemporary Global Labor Market*. Geneva, Switzerland: International Labor Organization.

--------and Karina Fernandez-Stark. 2010. The Offshore Services Industry: A Global Value Chain Approach. Durham: Center on Globalization Governance and Competitiveness, Duke University. Retrieved on November 15, 2014. http://www.cggc.duke.edu/pdfs/CGGC- CORFO_The_Offshore_Services_Global_Value_Chain_March_1_2010.pdf.

Giddens, Anthony. 1990. *The Consequences of Modernity*. Stanford, CA: Stanford University Press.

--------1991. *Modernity and Self Identity.* New York : Routledge.

--------2000. Runaway World: How Globalization is Reshaping Our Lives. New York: Routledge.

Gordon Ian R. and Philip McCann. 2007. "Industrial Clusters: Complexes, Agglomeration and/or Social Networks?" *Urban Studies* 37 (3) : 513–532.

Gorman Elizabeth H. and Rebecca L. Sandefur. 2011. "Golden Age," Quiescence, and Revival: How the Sociology of Professions Became the Study of Knowledge-Based Work." *Work and Occupations* 38 (3): 275-302.

Gouldner, Alvin W. 1979. *The Future of Intellectuals and the Rise of the New Class*. *New York: Seabury Press.*

Gourevitch, Peter and James Shinn. 2007. *Political Power and Corporate Control: The New Global Politics of Corporate Governance*. Princeton, NJ: Princeton University Press.

Gupta, Dipankar. 2000. *Mistaken Modernity: India Between Worlds*. India: HarperCollins Publishers.

Granovetter, Mark S. 1973. " The Strength of Weak Ties." *American Journal of Sociology* 78 (6): 1360-1380.

Grossman, Gene M. and Elhanan Helpman. 1999. *Innovations and Growth in the Global Economy*. Cambridge, MA: Massachusetts Institute of Technology Press.

Hall Peter A. and David Soskice. 2001. "An Introduction to Varieties of Capitalism." Pp. 1-44 in *Varieties of Capitalism: Institutional Foundations of Comparative Advantage edited by* Peter Hall and David Soskice. Oxford: Oxford University Press.

Handy, Charles B. 1994. "*The Empty Raincoat: Making Sense of the Future.*" London: Hutchinson.

Hanson Gordon H., Raymond J. Mataloni, Jr., Matthew J. Slaughter, Robert Z. Lawrence and James Levinsohn. 2001. "Expansion Strategies of U.S. Multinational Firms." *Brookings Trade Forum* 2001 (1): 245-294.

---------, Raymond J. Mataloni Jr. and Matthew J. Slaughter. 2005. "Vertical Production Networks in

Multinational Firms." *The Review of Economics and Statistics* 87 (4): 664-678.

Harding, E.U. 1989. "India: After IBM's Exit, an Industry Arose." *Software Magazine* 9 (14): 48-54.

Hariram. 2016. "Indian IT Companies: Changing Contours of HR Policyscape." IT/ITES Employees Center, February 19.

Hartley, Morgan and Chris Walker. 2012. "The Culture Shock of India's Call Centers." Forbes Dec 16. Retrieved on April 10th, 2019. https://www.forbes.com/sites/morganhartley/2012/12/16/the-culture-shock-of-indias-call-centers/#576aa1af72f5

Harvey, David. 2005. *A Brief History of Neo Liberalism.* New York: Oxford University Press.

Harvey, William S. 2008. "Strong and Weak Ties? British and Indian Expatriate Scientists Finding Jobs in Boston." *Global Networks* 8 (4): 453-473

Heeks, Richard. 1996. *India's Software Industry: State Policy, Liberalisation and Industrial Development.* New Delhi: Sage Publications.

Herrigel, Gary and Volker Wittke. 2005. "Varieties of Vertical Disintegration: The Global Trend toward Heterogeneous Supply Relations and the Reproduction of Difference in US and German Manufacturing." Pp. 277-311 in *Changing Capitalisms? Internationalization, Institutional Change, and Systems of Economic Organization*, edited by Glenn Morgan, Richard Whitley, and Eli Moen. Oxford : Oxford University Press.

--------and Jonathan Zeitlin. 2010. "Inter-firm Relations in Global Manufacturing: Disintegrated Production and its Globalization." Pp. 527-564 in *The Oxford Handbook of Comparative Institutional Analysis* edited by Glenn Morgan, John Campbell, Colin Crouch, Peer Hull Kristensen, Ove Kai Pedersen, and Richard Whitley. Oxford: Oxford University Press.

Heryanto, Ariel. 1999. "The Years of Living Luxuriously: Identity Politics of Indonesia's New Rich." Pp. 160-188 in Culture and Privilege in Capitalist Asia, edited by Michael Pinches. London: Routledge.

Hite Amy Bellone and J Timmons Roberts. 2008. "Development and Globalization: Recurring Themes. Pp. 1-16 in *The Globalization and Development Reader*, edited by J Timmons Roberts and Amy Bellone Hite. London: Blackwell Publishing Limited.

Hirsch Paul M. and Charles E. Naquin. 2001. "The Changing Sociology of Work and Reshaping of Careers." Pp. 427-436 in *Working in Restructured Workplaces: Challenges and New Directions for the Sociology of Work*, edited by Sage Publications: Thousand Oaks, California. 2001.

Holmes, Douglas R. and George E. Marcus. 2007. "Cultures of Expertise and the Management of Globalization: Toward the Re-Functioning of Ethnography". Pp. 235-252 in in *Global Assemblages: Technology, Politics, and Ethics as Anthropological Problems*, edited by Aihwa Ong and Stephen J. Collier. Blackwell Publishing Limited.

Holston James. 2008. *Insurgent Citizenship: Disjunctions of Democracy and Modernity in Brazil.* Princeton, NJ: Princeton University Press.

Hoogenboom Marcel, Willem Trommel & Duco Bannink. 2006. "Knowledge Societies (Plural): The Rise of New Knowledge Types and the Global Division of Labour." Pp. 165-178 in *Transformation of Work in a Global Knowledge Economy: Toward a Conceptual Framework,* edited by Ursula Huws. WORKS Project. Retrieved on July 30, 2013. http://www.ftu-namur.org/fichiers/WORKS-Conceptual_framework.pdf

Hung Shiu-Wan. 2009. "Development and Innovation in the IT industries of India and China." *Technology*

*in Society* 31: 29–41.

Huws, Ursula and Monique Ramioul. 2006. "Globalisation and the Restructuring of Value Chains". Pp. 15-30 in *Transformation of Work in a Global Knowledge Economy: Toward a Conceptual Framework*, edited by Ursula Huws. WORKS Project. Retrieved on July 30, 2013. http://www.ftu-namur.org/fichiers/WORKS-Conceptual_framework.pdf

Illvarasan Vigneswara P. 2008. "Software Work in India: A Labor Process View". Pp. 162-189, in *In an Outpost of the Global Economy: Work and Workers in India's Information Technology Industry*, edited by Carol Upadhya and A.R. Vasavi. New Delhi: Sage Publications.

Jacobs, Jerry A. and Kathleen Gerson. 2004. Understanding Changes in American Working Time: A Synthesis. Pp. 25-45 in *Fighting for Time: Shifting Boundaries of Work and Social Life*, edited by Cynthia Fuchs Epstein and Arne L. Kallenberg. New York: Russell Sage Foundation.

Jodhka, Surinder and Katherine Newman. 2007. "The Language of Globalization: Meritocracy, Productivity and Persistence of Caste-Based Stereotypes Among Indian Employers." *Economic and Political Weekly 42 (41):* 4125-4132.

Jameson Fredric.1991. *Post Modernity Or the Cultural Logic of Late Capitalism*. Durham, NC: Duke University Press.

Jeffrey, Craig. 2010. *Timepass: Youth, Class, and the Politics of Waiting in India*. Stanford, CA: Stanford University Press.

Jorgensen, Dale W. And Khuong M. Vu. 2005. "Information, Technology and the World Economy." Pp. 71-124 in *The Network Society: From Knowledge to Policy*, edited by Manuel Castells and Gustavo Cadoso. Washington DC: John Hopkins Center for Transatlantic Relations.

Kalleberg, Arne L. 2001. "The Advent of the Flexible Workplace: Implications for Theory and Research." Pp. 437-454 in *Working in Restructured Workplaces: Challenges and New Directions for the Sociology of Work* edited by Daniel B. Cornfield, Karen Campbell and Holly J. McCammon. Thousand Oaks, CA: Sage Publications.

------- 2003. "Flexible Firms and Labor Market Segmentation: Effects of Workplace Restructuring on Jobs and Workers." *Work and Occupations* 30 (2): 154-175.

Kapur, Devesh. 2002. "The Causes and Consequences of India's IT Boom." *India Review* 1 (2) : 91-110.

-------- 2010. Middle Class in India: A Social Formation or Political Actor. Pp. 143-170 in *Political Power and Social Theory*, edited by Julian Go. Bradford, GBR: Emerald Group Publishing Ltd.

-------2010. *Diaspora, Development, and Democracy: The Domestic Impact of International Migration From India*. Princeton, NJ: Princeton University Press.

-------- Neelanjan Sircar and Milan Vaishnav. 2018. "The Importance of Being Middle Class in India." Pp. 39-58 in *The New Middle Class in India and Brazil: Green Perspectives?*, edited by Dawid Danilo Bartelt and Axel Harneit-Sievers. New Delhi: Academic Foundation

Khambapati Uma S. 2002. The Software Industry and Development: The Case of India. *Progress in Development Studies* 2 (1) : 23–45.

Khanna, Tarun, and Krishna G. Palepu. 2004. "Globalization and Convergence in Corporate Governance: Evidence from Infosys and the Indian software industry." *Journal of International Business Studies* 35 (6) : 484-507.

Kochar, Ramesh. 2015. "A Global Middle Class Is More Promise than Reality: From 2001 to 2011, Nearly 700 Million Step Out of Poverty, but Most Only Barely." *Pew Research Center*, July 18. Retrieved on July 10, 2016. https://www.pewresearch.org/global/2015/07/08/a-global-middle-class-is-more-promise-than-reality/

Kohli, Atul. 2004. *State-Directed Development: Political Power and Industrialization in the Global Periphery*. Cambridge: Cambridge University Press.

-------- 2009. "Politics of Economic Growth in India, 1980-2005, Part I: The 1980s'." Pp. 140-163 in *Democracy and Development in India, From Socialism to Pro-Business*, edited by Atul Kohli. New Delhi: Oxford University Press.

Korpi, Walter. 2006. "Power Resources And Employer-Centered Approaches In Explanations Of Welfare States And Varieties Of Capitalism Protagonists, Consenters, and Antagonists." *World Politics*. 58: 167-206.

Krishna, Anirudh, Vijay Brihmadesam. 2006. "What Does It Take to Become a Software Professional?" *Economic and Political Weekly* 41 (30) : 3307-3314.

------- 2014. "Examining the Structure of Opportunity and Social Mobility in India: Who Becomes an Engineer?" *Development and Change*, 45 (1): 1–28.

--------, and D. Bajpai. 2015. "Layers in Globalising Society and the New Middle Class in India: Trends, Distribution and Prospects." *Economic and Political Weekly* 50 (5): 69–77.

Krishnan, Sandhya and Neeraj Hatekar. 2017. Rise of the New Middle Class in India and it's Changing Structure." *Economic and Political Weekly* 52 (22): 40-48.

Kumar, Nagesh and K.J. Joseph. 2005. "Export of Software and Business Process Outsourcing from Developing Countries: Lessons from the Indian Experience." *Asia-Pacific Trade and Investment Review* 1 (1) : 91-110.

Kumar, Vivek. 2004. "Understanding the Politics of Reservation: A Perspective from Below." *Economic and Political Weekly* 40 (9) : 803-806.

Kunda, Gideon. 1992. *Engineering Culture: Control and Commitment in a High-Tech Corporation.* Philadelphia: Temple University Press.

Lakha, Salim. 1994. "The New International Division of Labour and the Indian Computer Software Industry." *Modern Asian Studies* 28 (2) : 381-408.

------- 1999. "The State, Globalization and Indian Middle Class Identity." Pp. 252-275 in *Culture and Privilege in Capitalist Asia,* edited by Michael Pinches. London: Routledge.

Lamont, Michele. 1992. *Money, Morals, and Manners : The Culture of the French and American Upper-Middle Class.* Chicago, IL: University of Chicago Press.

--------- and Virag Molnar. 2002. The Study of Boundaries in Social Sciences. *Annual Review of Sociology* 28 : 167-195.

Lane, Carrie M. 2011. *Company of One: Insecurity, Independence, and the New World of White-Collar Unemployment.* Ithaca, NY: ILR Press.

Lareau, Annette. 2011. *Unequal Childhoods: Class, Race, and Family Life*. Berkeley, CA: University of

California Press.

Larson, Magali, S. 1979. *The Rise of Professionalism : A Sociological Analysis.* Berkeley, CA: University of California Press.

Lee, Ching Kwan. 1998. *Gender and the South China Miracle : Two Worlds of Factory Women.* Berkeley, CA: University of California Press.

----------2007. *Against the Law: Labor Protests in China's Rustbelt and Sunbelt.* Los Angeles, CA: University of California Press.

Levin, Peter. 2004. "Gender, Work, and Time: Gender at Work and at Play in Futures Trading." Pp. 249-281 in *Fighting for Time: Shifting Boundaries of Work and Social Life*, edited by Cynthia Fuchs Epstein and Arne L. Kallenberg. New York: Russell Sage Foundation.

Levy Frank Murnane and Richard J. 2005. *The New Division of Labor: How Computers are Creating the Next Job Market.* Princeton, NJ: Princeton University Press.

Liechty, Mark. 2003. *Suitably Modern: Making Middle-class Culture in a New Consumer Society*. Princeton, NJ: Princeton University Press.

Lin, Nan, Walter M. Ensel, and John C. Vaughn. 1981. "Social Resources and Strength of Ties: Structural Factors in Occupational Status Attainment." *American Sociological Review* 46 (4): 393-405.

Lizardo, Omar. 2006. "How Cultural Tastes Shape Social Networks." *American Sociological Review* 71 (5): 778-807.

Lopez, Steven Henry. 2010. "Workers, Managers, and Customers: Triangles of Power in Work Communities." *Work and Occupations* 37 (3): 251-271.

Macher, Jeffrey T., and David C. Mowery. 2004. "Vertical Specialization and Industry Structure in High Technology Industries." *Advances in Strategic Management* 21: 317-355.

Majumdar, Sumit K. 2010. "Globalization and Relative Compensation in India's Information Technology Sector." *Information Technologies and International Development* 6 (1) : 21-33.

Marks, Abigail and Chris Baldry. 2009. "Stuck in the Middle With Who? The Class Identity of Knowledge Workers." *Work, Employment and Society* 23 (1) : 1-24.

Marschall, D. 2012. *The Company We Keep: Occupational Community in the High-Tech Network Society.* Philadelphia : Temple University Press.

Marshall, Don D. "Understanding Late-Twentieth-Century Capitalism: Reassessing the Globalization Theme." *Government and Opposition* 31 (2) :193-215.

Mathews, Eldo. 2015. Who Gets Into India's IITs?" *A Blog From the Center for International Higher Education, Inside Higher Ed,* April 14.

Mazumdar, S. 2008. "Crony Capitalism and India: Before and After Liberalization", *ISID Working Paper* 2008 (04). Retrieved on April 30, 2013 http://isidev.nic.in/pdf/WP0804.PDF

Mazzarella, William. 2003. *Shoveling Smoke: Advertising and Globalization in Contemporary India*. Duke Durham, NC: Duke University Press.

McBreen, Pete. 2002. *Software Craftsmanship: The New Imperative*. Addison-Wesley Professional.

McKay, Steven. 2006. *Satanic Mills or Silicon Islands? : The Politics of High-Tech Production in the*

*Philippines*. Ithaca, NY: Cornell University Press.

Meyer, Christian and Nancy Birdsall. 2012. "New Estimates of India's Middle Class: Technical Note." *Center for Global Development.* Retrieved on April 3, 2019. https://www.cgdev.org/sites/default/files/archive/doc/2013_MiddleClassIndia_TechnicalNote_CGDNote.pdf

Milkman, Ruth. 2006. *L.A. Story : Immigrant Workers and the Future of the U.S. Labor Movement*. New York: Russell Sage Foundation.

Mills, C. Wright. 1951. *White Collar: The American Middle Classes.* New York: Oxford University Press.

------ 1956. *The Power Elite.* . New York: Oxford University Press.

Ministry of Human Resource Development, Government of India. 2014. *Annual Report 2013-14.* Retrieved on April 13, 2017. https://mhrd.gov.in/sites/upload_files/mhrd/files/document-reports/AR2013-14.pdf

Mishra, Sudhanshu. 2013. "Revealed: Inside Kota's Rs 300 Crore Coaching Industry, Where 1.5 Lakh Students Brave Cut-throat Competition to Crack IIT." *Daily Mail*, April 23.

Moen, Phyllis Jack Lam, Samantha Ammons and Erin L. Kelly. 2013. "Time Work by Overworked Professionals: Strategies in Response to the Stress of Higher Status." *Work and Occupations* 2013 40 (2): 79-114.

Mukherjee, Anahita. 2014. "Degrees of Bias." *Times of India*, September 24.

Mukherjee-Reed, Ananya. 2001. *Perspectives on the Indian Corporate Economy: Exploring the Paradox of Profits.* London: Palgrave-McMillan.

Mukherjee, Sanjukta. "Gendered Geographies of Alienation in India's New High-Tech Workplace." Pp. 50-75 in *In an Outpost of the Global Economy: Work and Workers in India's Information Technology Industry*, edited by Carol Upadhya and AR Vasavi. New Delhi: Sage Publications.

Murphy, Jonathan. 2011. "Indian Call Centre Workers: Vanguard of a Global Middle Class?" *Work, Employment & Society* 25 (3): pp. 417-433.

Nanda, Prashant K. 2015. "Glass Ceiling Cracking the IIT Code: Family Resistance and Ignorance About Options Are Just Two of the Hurdles Standing Between Girls and IITs." *Live Mint*, October 28.

NASSCOM. The IT-BPM Sector in India - Strategic Review 2011,
--------The IT-BPM Sector in India - Strategic Review 2012.
--------The IT-BPM Sector in India - Strategic Review 2013.
--------NASSCOM Annual Report 2012.
--------NASSCOM Annual Report 2013.
--------NASSCOM Annual Report 2014.

Newman Katherine S. and Ashwini Deshpande. 2010. "Roadblocks at the High End: The Role of Caste in Post-University Employment." Pp. 187-213 in *Discrimination in An Unequal World*, edited by Miguel Angel Centeno and Katherine S. Newman. New York: Oxford University Press.

Nisbett, Nicholas. 2009. *Growing Up in the Knowledge Society: Living the IT Dream in Bangalore*. New Delhi : Routledge.

Noon, Mike. 2010. "The Shackled Runner: Time to Rethink Positive Discrimination?" *Work, Employment and Society* 24 (4): 728-739.

Ockey, Jim. 1999. “Creating the Thai Middle Class.” Pp. 231-251 in *Culture and Privilege in Capitalist Asia,* edited by Michael Pinches. London: Routledge.

O'Hearn, Denis. 2001. *The Atlantic Economy: Britain, the US and Ireland*. New York: Manchester University Press.

Ong, Aihwa. 2006. *Neoliberalism as Exception: Mutations in Citizenship and Sovereignty*. Durham NC: Duke University Press.

-------2010. *Spirits of Resistance and Capitalist Discipline Factory Women in Malaysia. Albany, NY:* State University of New York Press.

Orru, Marco. 1997. “Dirigiste Capitalism in France and South Korea.” Pp. 368-382 in *The Economic Organization of East Asian Capitalism*, edited by Marco Orru, Nicole Woolsey Biggart, and Gary G. Hamilton. Thousand Oaks, CA: Sage Publications.

Ó Riain, Seán. 2000. "The Flexible Developmental State: Globalization, Information Technology and the ‘Celtic Tiger’." *Politics and Society* 28 (2) : 157-193.

---------- 2002. “High-Tech Communities: Better Work or Just More Work?” *Contexts* 1 (4): 36-41.

----------2004. *The Politics of High Tech Growth: Developmental Network States in the Global Economy*. Cambridge: Cambridge University Press.

----------2008. "Competing State Projects in the Contemporary Irish Political Economy." Pp. 165-185 in *Contesting the State: Lessons from the Irish Case*, edited by Maura Adshead, Peadar Kirby and Michelle Millar. Manchester: Manchester University Press.

–----------2010. “The Missing Customer and the Ever-Present Market : Software Developers and the Service Economy.” *Work and Occupations* 37 (3): 320-348.

----------2011. “Human Capital Formation Regimes: States, Markets and Human Capital in an Era of Globalisation” Pp. 588-617 in *Handbook of Human Capital* ,edited by Alan Burton Jones and J.C. Spender. Oxford: Oxford University Press.

Pager, Devah. 2010. “Measuring Discrimination.” Pp. 45-62 in *Discrimination in An Unequal World,* edited by Miguel Angel Centeno and Katherine S. Newman. New York: Oxford University Press.

Rhacel, Parrenas S. 2015. *Servants of Globalization.* Stanford, CA: Stanford University Press.

Royster, Dierdre A. 2005. *Race and the Invisible Hand : How White Networks Exclude Black Men From Blue-collar Jobs.* University of California Press.

Pachuki, Mark A., Sabrina Pendergrass and Michele Lamont. 2007. Boundary Processes: Recent Theoretical Developments and New Contributions. *Poetics* 35: 331-351.

Pandey, Gyanendra. 2009. "Can There Be a Subaltern Middle Class? Notes on African American and Dalit History." *Public Culture* 21 (2) : 321-342.

Parthasarathy, Balaji. 2000. *Globalization and Agglomeration in Newly Industrializing Countries: The State and the Information Technology Industry in Bangalore, India.* Unpublished Dissertation, University of California, Berkley.

--------- and Yuko Aoyama. 2006. “From Software Services to R&D Services: Local Entrepreneurship In the Software Industry in Bangalore, India.” *Environment and Planning* 38 (7) :1269-1285.

--------2009. “Information Technology.” Pp. 469-474 in *International Encyclopedia of Human Geography,* edited by Rob Kitchin and Nigel Thrift. London: Elsevier.

-------and Ricardo Lange. 2010. “Organizational Impacts of Information Technology.” Pp. 289-298 in *The Handbook of Technology Management*, edited by Hossein Bidgoli. Hoboken, NJ: John Wiley.

Patel, Reena. 2010. Working the Night Shift. Stanford, CA: Stanford University Press.

Patibandla, Murali, and Bent Petersen. 2002. "Role of Transnational Corporations in the Evolution of a High-Tech Industry: The Case of India's Software Industry." *World Development* 30 (9): 1561-1577.

Peck, Jamie A. 1989. “Reconceptualizing the Local Labour Market : Space, Segmentation and the State.” *Progress in Human Geography* 13: 42-61.

-------and Nik Theodore. 2007. “Variegated Capitalism.” *Progress in Human Geography 31 (6) :731-772.*

Perlow, Leslie. 2001. “Time to Coordinate: Toward an Understanding of Work-Time Standards and Norms In a Multi-country Study of Software Engineers.” *Work and Occupations* 28 (1): 91-111.

Perrons, Diane, Colette Fagan, Linda McDowell, Kath Ray and Kevin Ward. 2005. “Work, Life and Time in the New Economy: An Introduction.” *Time and Society* 14 (1): 51-64.

Pinches, Michael. 1999. “Cultural Relations, Class and the New Rich of Asia.” Pp 1-56 in *Culture and Privilege in Capitalist Asia,* edited by Michael Pinches. London: Routledge.

Piore, Michael J and Charles F. Sabel. 1984. *The Second Industrial Divide: Possibilities for Prosperity.* New York: Basic Books.

Poggendorf-Kakar, Katharina. 2001. “Middle Class Formation and the Cultural Construction of Gender in Urban India” Pp. in *Middle Class Values in India and Western Europe,* edited by Imtiaz Ahmad and Helmut Reifeld New Delhi: Social Science Press.

Portes, Alejandro. 2010. *Economic Sociology: A Systematic Inquiry*. Princeton, NJ: Princeton University Press.

Prasad, Monica. "International Capital on “Silicon Plateau: Work and Control in India's Computer Industry." *Social Forces* 77 (2) : 429-452.

Presser, Harriet. 2004. “Employment in a 24/7 Economy: Challenges for the Family.” Pp. 46-76 in *Fighting for Time: Shifting Boundaries of Work and Social Life*, edited b

PTI. 2016. “IT Highest-Paying Sector in India: Study.” *Times of India*, February 15.

Radhakrishnan, Smitha. 2011. *Appropriately Indian: Gender and Culture in a New Transnational Class*. Durham, NC: Duke University Press.

Raider, Holly J. And Ronald S. Burt. 1996. “Boundaryless Careers and Social Capital.” Pp. 187-200 in *The Boundaryless Career: The New Employment Principle of a New Organizational Era*, edited by Michael B. Arthur and Denise M. Rousseau. New York: Oxford University Press.

Rajagopal, Arvind. Thinking About the New Indian Middle Class: Gender, Advertising and Politics in an Age of Globalization. Pp. 57-100 in *Signposts: Gender Issues in Post-Independence India*, edited by Rajeswari Sunder Rajan. New Delhi: Kali for Women Press.

Ramachandraiah, C. 2003. “Information Technology and Social Development.” *Economic and Political*

*Weekly* 38 (12/13): 1192-1197.

Ramamurthy, Priti. 2010. "Hegemonic Developments: The New Indian Middle Class, Gendered Subalterns, and Diasporic Returnees in the Event of Neoliberalism." *Journal of Women in Culture and Society,* 36 (1) : 127-152.

Ray, Raka and Seemin Qayum. 2009. *Cultures of Servitude : Modernity, Domesticity, and Class in India*. Stanford. CA: Stanford University Press.

-------2010. "The "Middle Class": Sociological Category or Proper Noun?" Pp. 313-322 in *Political Power and Social Theory,* edited by Julian Go. Bradford, GBR: Emerald Group Publishing Ltd.

Reed, Michael. 1996. "Expert Power and Control in Late Modernity: An Empirical Review and Theoretical Synthesis. *Organization Studies* 14 (4): 573-597.

Remesh, Babu P. 2008. "Work Organisation, Control and 'Empowerment': Managing the Contradictions of Call Centre Work." Pp. 235-262 in *In an Outpost of the Global Economy: Work and Workers in India's Information Technology Industry*, edited by Carol Upadhya and AR Vasavi. New Delhi: Sage Publications.

Ridgeway CL. 2014. "Why Status Matters for Inequality." *American Sociological Review* 79 (1) 1–16.

Chakrabarty (2016), "Only 7 Percent Engineering Graduates Employable: What's Wrong With India's Engineers?" Times of India July 13. http://indiatoday.intoday.in/education/story/engineering-employment-problems/1/713827.html

Rubin, Beth A. And Charles J. Brody. 2005. "Contradictions of Commitment in the New Economy: Insecurity, Time and Technology. *Social Science Research* 34: 843-861.

Rubin Beth A. And Brian T. Smith. 2001. "Reemployment in the Restructured Economy: Surviving Change, Displacement and the Gales of Creative Destruction." Pp. 323-341 in *Working in Restructured Workplaces: Challenges and New Directions for the Sociology of Work*, edited by Daniel B. Cornfield, Karen Campbell and Holly J. McCammon. Thousand Oaks, CA: Sage Publications.

Rudolph, Lloyd I., and Susanne Hoeber Rudolph. 1987. *In Pursuit of Lakshmi: The Political Economy of the Indian State*, Chicago: University of Chicago Press.

Rudrappa, Sharmila. 2009. "Cyber-Coolies and Techno-Braceros: Race and Commodification of Indian Information Technology Guest Workers in the United States." *University of San Francisco Law Review* 44 (2): 353-372. Retrieved on August 1, 2013. https://repository.usfca.edu/cgi/viewcontent.cgi?article=1297&context=usflawreview

Sahay, Sundeep, Brian Nicholson, and Shenai Krishna. 2003. *Global IT Outsourcing: Software Development Across Borders.* Cambridge: Cambridge University Press.

Sassen, Saskia. 1998. *Globalization and Its Discontents: Essays on the New Mobility of People and Money.* New York: New York Press.

-------2012. *Cities in a World Economy*. Los Angeles, CA: Sage/Pine Forge Press.

Saraswati, Jyoti. 2012. *Dot.Compradors: Power and Policy in the Development of the Indian Software Industry.* London: Pluto Press.

Saxenian, AnnaLee. 1996. *Regional Advantage: Culture and Competition in Silicon Valley and Route 12.* Cambridge, MS: Harvard University Press.

------2006. *The New Argonauts: Regional Advantage in a Global Economy*. Cambridge, MS : Harvard University Press, 2006.

Schellenberg, Kathryn. 2001. "Taking It or Leaving It: Instability and Turnover in a High-Tech Firm." Pp. 211-228 in *Working in Restructured Workplaces: Challenges and New Directions for the Sociology of Work*, edited by Daniel B. Cornfield, Karen Campbell and Holly J. McCammon. Thousand Oaks, CA: Sage Publications.

Schmidt, Vivien A. 2007. "*Changes in Comparative Political Economy: Taking Labor Out, Bringing the State Back in, Putting the Firm Front and Center*." Montreal, Canada. Unpublished. Retrieved on April 20, 2013. http://aei.pitt.edu/8022/

Schrank Andrew .2009. "Understanding Latin American Political Economy: Varieties of Capitalism or Fiscal Sociology?" *Economy and Society*, 38 (1) : 53-61.

Schware, Robert. 1992. "Software Industry Entry Strategies for Developing Countries: A 'Walking on Two Legs' Proposition." *World Development* 20 (2): 143– 64.

Seidman, Gay. 1994. Manufacturing Militance : Workers' Movements in Brazil and South Africa, 1970-1985. Berkeley, CA: University of California Press.

--------2007. *Beyond the Boycott : Labor Rights, Human Rights, and Transnational Activism.* New York: Russel Sage Foundation.

Selzer, Amy Fracker and Patrick Heller. "The Spatial Dynamics of Middle-Class Formation in PostAparthied South Africa: Enclavization and Fragmentation in Johannesburg." *Political Power and Social Theory* 21 : 171-208.

Sharma, Vivek. 2005. "World Class Universities." *Economic and Political Weekly* 40 (24) : 2370-2456.

Sharone, Ofer. 2004. "Engineering Overwork: A Bell Curve Management at a High Tech Firm." Pp. 191-218 in *Fighting for Time: Shifting Boundaries of Work and Social Life*, edited by Cynthia Fuchs Epstein and Arne L. Kallenberg. New York: Russell Sage Foundation.

Shatkin, Gavin. 2010. "Middle Class or Propertied Class? Class Politics and urban Redevelopment in Contemporary Asia." Political Power and Social Theory 21 : 269-279.

Shehzad Nadeem. 2011. *Dead Ringers : How Outsourcing is Changing the Way Indians Understand Themselves.* Princeton, NJ. Princeton University Press.

Slaughter, Anne-Marie. 2015. "A Toxic Work World." *New York Times Sunday Review*, September 18. Retrieved on September 20. http://www.nytimes.com/2015/09/20/opinion/sunday/a-toxic-work-world.html?emc=edit_ae_20150920

Smith, Vicki. 2012. ""You Get the Economy You Choose": The Political and Social Construction of the New Economy." *Work and Occupations* 39 (4): 148-156.

Sridharan E. 2004. "Evolving Towards Innovation? The Recent Evolution and Future Trajectory of the Indian Software Industry." Pp. 27-50 in *India in the Global Software Industry: Innovation, Firm Strategies and Development*, edited by Anthony P. D'Costa and Eswaran Sridharan. London: Palgrave Macmillan.

Srinivas, M.N. 1989. *The Cohesive Role of Sanskritization and Other Essays.* New Delhi: Oxford University Press.

Srinivas, Smriti. 2001. *Landscapes of Urban Memory: The Sacred and the Civic in India's High-Tech City.* Minneapolis, MN: University of Minnesota Press.

Standing, Guy. 2011. *Precariat: The New Dangerous Class.* UK: Bloomsbury Academic.

Subramanian, A. 2015. "Recovering Caste Privilege: The Politics of Meritocracy at the Indian Institutes of Technology." *New Subaltern Politics,* 76-99.

Subramanian, C. R. 1992. *India and the Computer.* New Delhi: Oxford University Press.

Talukdar, Jaita, and Annulla Linders. 2013. "Gender, Class Aspirations, and Emerging Fields of Body Work in Urban India." *Qualitative Sociology* 36 (1) : 101-123.

Taylor, Fredrick. 1911. *The Principles of Scientific Management.* New York: Harper and Brothers.

Tilman Altenburg, Hubert Schmitz, Andreas Stamm. 2008. "Breakthrough? China's and India's Transition from Production to Innovation." *World Development* 36 (2): 325-344

Upadhya Carol. 2004. "A New Transnational Capitalist Class? Capital Flows, Business Networks and Entrepreneurs in the Indian Software Industry." *Economic and Political Weekly* 39 (48): 5141-5151.

-------and A.R. Vasavi. 2006. *Work, Culture and Sociality In the Indian IT Industry: A Sociological Study.* Final Report submitted to Indo-Dutch Program for Alternatives in Development. Retrieved on August 10, 2013. https://www.researchgate.net/publication/278002564_Work_Culture_and_Sociality_in_the_Indian_Information_Technology_IT_Industry_A_Sociological_Study

--------2008a. "Management of Culture and Management Through Culture in Indian Software Outsourcing Industry." Pp. 101-135 *In an Outpost of the Global Economy: Work and Workers in India's Information Technology Industry*, edited by Carol Upadhya and AR Vasavi. New Delhi: Sage Publications.

--------2008b. "Rewriting the Code." Pp. 55-87 in *Patterns of Middle Class Consumption*, edited by Christophe Jaffrelot and Peter van der Veer. London: Sage Publications.

--------*2009.* "Controlling Offshore Knowledge Workers: Power and Agency in India's Software Outsourcing Industry." *New Technology, Work and Employment* 24 : 1-18.

--------2010. "Taking the High Road? Labor in the Software Outsourcing Industry." Pp. 101-135 in *Labour in Global Production Networks in India*, edited by Anne Posthuma and Dev Nathan, New Delhi: Oxford University Press:.

-------2011. "Software and the New Middle Class in India." Pp. 167-192 in *Elite and Everyman: The Cultural Politics of Indian Middle Classes*, edited by Raka Ray and Amita Baviskar. New Delhi: Routledge.

Vaid, Divya. 2012. "The Caste-Class Association in India: An Empirical Analysis." *Asian Survey* 52 (2) : 391-422.

Vallas, Steven P. 2001. "Symbolic Boundaries and the New Division of Labor: Engineers, Workers and the Restructuring of Factory Life." *Research in Social Stratification and Mobility* 18: 3-37.

Vallejo, Jody. 2012. *Barrios to Burbs: The Making of the Mexican American Middle Class*. Stanford University Press: Stanford California.

Varma, Pavan K. 2007. *The Great Indian Middle Class*. New Delhi: Penguin Books India

Vasavi, A.R. 2008. " 'Serviced from India': The Making of India's Global Youth Workforce." Pp. 211-234 in *In an Outpost of the Global Economy: Work and Workers in India's Information Technology Industry*, edited by Carol Upadhya and AR Vasavi. New Delhi: Sage Publications.

Veenstra. 2005. "Can Taste Illumine Class? Cultural Knowledge Forms of Inequality." *The Canadian Journal of Sociology* 30 (3): 247-279.

Wacquant, Loic. 1991. "Making Class: The Middle Class(es)." Pp. 39-64 in *Social Theory and Social Structure. Bringing Class Back In: Contemporary and Historical Perspectives*, edited by Scott G. McNall, Rhonda F. Levine and Rick Fantasia. Boulder, CO: Westview Press Inc.

Wallace, Michael, and Azamat Junisbai. 2003. "Finding Class Consciousness In the New Economy." *Research in Social Stratification and Mobility* 20 : 385-421.

Weber, Max. 1978. "*Economy and Society*." Edited by Roth Guenther and Claus Wittich. Berkeley: University of California Press.

-------- 1982. "Selections from Economy and Society, Volumes 1 and 2, and General Economic History." Pp. 60-86 in *Classes, Power and Conflict. Classical and Contemporary Debates*, edited by Anthony Giddens and David Held. Berkeley, CA: University of California Press.

--------- 1930. *Protestant Ethic and the Spirit of Capitalism.* Allen and Unwin.

Wade, Robert. 1990. *Economic Theory and the Role of Government in East Asian Industrialization.* Princeton: Princeton University Press).

Wang, Jun and Stephen Siu Yu Lau. 2009. "Gentrification and Shanghai's New Middle-Class: Another Reflection on the Cultural Consumption Thesis." *Cities* 26: 57-66.

Weeden, Kim A. 2015. "Why Do Some Occupations Pay More than Others? Social Closure and Earnings Inequality in the United States." *American Journal of Sociology* 108 (1): 55-101.

Weininger Elliot B. 2005. "Foundations of Pierre Bourdieu's Class Analysis." Pp. 82-118 in *Approaches to Class Analysis*, edited by Erik O. Wright. Cambridge: Cambridge University Press.

Weisskopf, Thomas E. 2010. Reflections on Globalization, Discrimination and Affirmative Action. In *Discrimination in An Unequal World, e*dited By Centeno Miguel Angel and Katherine S. Newman. New York: Oxford University Press.

Wright, Erik. O. 1982. "Class Boundaries and Contradictory Class Locations." Pp. 112-129 in *Classes, Power and Conflict: Classical and Contemporary Debates*, edited by Anthony Giddens and David Held. Berkeley, CA:University of California Press.

--------1989. "Rethinking, Once Again the Concept of Class Structures." Pp. 269-317 in *The Debate on Classes* edited by Erik Olin Wright. *New York: Verso.*

------- 2005. "Foundations of a Neo-Marxist Class Analysis." Pp. 4-30 in *Approaches to Class Analysis*, edited by Erik O. Wright. Cambridge: Cambridge University Press.

# APPENDIX A

## Interview Questions for Software Employees

## I Introductory Questions

- Which company do you work for and what is your job designation?
- How many years have you worked in the software industry?
- How many software industry jobs have you held so far?

## II Professional History

## Career Path and Goals

- Describe your jobs so far. Why did you join each of the jobs you took up? Why did you leave your former jobs?
- How did you get to this current job and company? How long are you planning to stay here? What will be your next move?
- What does it mean to be a "successful" person in the industry? What needs to be done to get there? Where do you plan to see yourself in the next 10 years? What is your ultimate career dream? Do you think you can achieve that dream? What steps have you taken to get there?
- Is there a dream company in which you want to work in? Why is it a dream place to work or a dream job? Do you feel you can get there? How? What is enabling/preventing you in getting there?
- Are there areas in which you would like to improve your skills relevant to your profession? What ways are open to you? What needs to be done in order to gain those skills? Are there constraints that prevent you from gaining them?

## III Career Goals and Strategies

### Education and Early Career

- What is your degree in? Why did you get it? Which college or university? Why did you join that college or university? How did you get in there- how hard or easy was it to get into that institution? Did you join coaching classes and were they expensive?
- Did it help you get this current job or the current career track you are in?
- When you went for your degree or just graduated college, what kind of a career track did you foresee for yourself? What were the considerations that made you get there?
- When you were in high school or before you went for your degree did you want to be in the software industry? What were your dreams then? If not the software industry what other career would you have chosen? Where did you foresee yourself in the next 10 or 15 years at that point?
- Going back can you say you have been able to stick to your early life goal? Why or why not? What changes did you have to make and why?

## III Ascribed Status

### Class and Family Background

- Which school did you go to? Was it English medium? How expensive was it? Was it conducive to your current career?
- What is your parents' education? What were your parents' occupations? Were they government servants? Are you in touch with your relatives? How educated are they? What kinds of careers do they/did they have?
- What class do you identify with while growing up and currently? Why? What defines middle class to you? What do you think are the essential aspects for maintaining middle class status?
- What are the next items/consumption goods that you aspire for? How do you plan to buy or invest in those? How and when do you plan to do that? (time line)
- Which region/city do you come from? How was belonging to this place conducive or detriment in your current career?

## Social Networks

- While growing up who are the main people who have provided guidance, advice in your professional growth and education? Have you taken advice from your parents with your career- which college to go to, what degree you wanted to get and which companies you would like to join? Who were the people who played an important role in your career decision?
- In what way did you parents support you in your career, degree, demands of professional life etc.? Are there areas in which you wish there was more support they could give you?
- Can you name a few people in your family or friends network who were role models or inspiration to you especially like you look up to them in matters of your own career choice, education? Did you ever take guidance or advice from them?
- Looking at your peers-successful friends, co-workers and acquaintances, do you believe their individual merit alone accounts for their success? Can you name what those aspects of their success are? Do you think connections help in doing well in the software industry?
- Are you in touch with your for classmates? Do you have a network? How do you maintain contact/remain in touch with them? Is it useful professionally to you? How? (Similarly for past colleagues).

## Caste and Casteism

- Are you comfortable identifying your caste?
- Are you a part of any caste based groups or networks?
- (If scheduled caste/tribe) Have you experienced discrimination based on caste?
- Do you feel caste plays a role in the software industry? What do you feel about reservations- in education? What do you feel about reservations in software industry jobs and private sector employment? Do you know of anyone who has benefited out of reservations?
- Do you know if many people from scheduled caste or tribe are able to join the industry? How easy or hard is it for scheduled castes or tribes to get into the industry?

## IV Labor Process

### Insecurity and Precariousness

- Do you feel secure about your job? What are the main issues you face with job security? Compare across employers and 2008 and 2001 recession time and the current 2014 layoffs in software in India. What about old age security, how do you deal with it?
- If there is a worldwide recession do you think how do you think the software industry will get affected in India? Do you ever fear your job might be on the line? Have you ever been laid off?
- What do you feel about the current layoffs going on in the software industry because of restructuring?

### Work Culture

- What does your average workday look like- describe? How many hours do you work generally and does it change in a given year or period? If yes why so?
- Do you think you are adequately able to look after your family with your current job? Which areas are lagging behind and which areas are fine?
- How do you do skill development in your company to keep abreast of technology or train freshers? Have you done any and were they useful? How much time do you need for training?
- Did you undergo cultural training? What did they teach you there? Was it useful?
- What perks, what facilities and benefits attracted and retained you in this company?
- How are promotions grated in your current and previous workplaces? What role do they play in your life? How do you get them?

### Work Relationships, Monitoring and Supervision

- How does your workplace ensure employees are coming on time and staying the whole time?
- How do you ensure employees are doing their job correctly, they are following instructions? What kinds of reporting structures are in place? Are you ok with that or do you think they are too restrictive or too lax? Compare across other employers

- How do you communicate and coordinate your work with your subordinates and supervisors? On what areas do you collaborate? How often does that happen? Compare across other employers.
- What are the decisions and considerations generally made for getting new hires? What do you think your company looks for? Any thoughts about how it might be similar or different in your jobs/employers?