Working Paper Series

#2007-035

Approach for analysing capabilities in latecomer software companies

Rossitza Rousseva

United Nations University - Maastricht Economic and social Research and training centre on Innovation and Technology
Keizer Karelplein 19, 6211 TC Maastricht, The Netherlands
Tel: (31) (43) 388 4400, Fax: (31) (43) 388 4499, e-mail: info@merit.unu.edu, URL: http://www.merit.unu.edu

Approach for analysing capabilities in latecomer software companies

Rossitza Rousseva[1]

November 2007

Abstract

Software development activities have been identified as a 'window of opportunity' for latecomer companies. Notwithstanding the growth of software development activities in the latecomers, there are single exceptional cases of companies that have successfully launched their own products in global software markets. This suggests that most of the latecomer software companies possess limited technological capabilities. This point has not been explored systematically in the literature so far, as the critical literature review reveals. This paper develops an approach for analysing technological capabilities in latecomer software industries. It outlines the specifics in analysing technological capability in latecomer software companies and makes an account of the array of technical and organisational capabilities associated with development of software technological capability. The research also looks at the diverse learning paths pursued by the latecomer companies in developing software activities. The study improves our understanding about the complexity in developing software industries in latecomer context, and the challenges the latecomer companies face in entering and competing in international markets.

Keywords: technological capabilities, software industry, latecomers
JEL codes: O32, O33, O57

UNU-MERIT Working Papers
ISSN 1871-9872

Maastricht Economic and social Research and training centre on Innovation and Technology, UNU-MERIT

1 United Nations University Maastricht Research and training centre on Innovation and Technology (UNU-MERIT), The Netherlands and Science and Technology Policy Research (SPRU), UK Current contact details: UNU-MERIT, Keizer Karelplein 19, 6211 TC Maastricht, The Netherlands; e-mail: rousseva@merit.unu.edu, R.I.Rousseva@sussex.ac.uk

1. Introduction

In the last two decades a group of studies has been emphasising that the information technologies (ITs) open opportunities for leapfrogging by latecomer companies, e.g. companies that originate and are embedded in a context that is not advanced (Soete 1985; Steinmueller 2001). It has been observed that the availability of skilful human capital creates a base for development of IT industries by latecomer countries. The software industry is, in principle, a low-capital but knowledge and skill-intensive industry, and the international market for software is big and growing (OECD 2004; Steinmueller 2004). Due to their higher contribution to economic growth the development of software and other high-tech industries has a potential to foster economic development in latecomers (Kuznets 1957). For the above reasons, the discussion about developing indigenous software industries in the latecomer context has gained particular attention both in academic and policy literature for more than a decade (OECD 2004; Soete 1985; Steinmueller 2004; UNIDO 1988).

A number of latecomers have attempted to develop software industries in the last decade. Different countries have followed different paths: development of the latecomer software industries in some latecomers is foreign-led, in some it is indigenous-based; some latecomer software industries are predominantly outsourcing-driven, some develop their own software activities; some latecomer software industries are export-driven, others remain domestic-oriented. Research on these developments is burgeoning following the recent outbursts of software development activities undertaken by latecomers (Arora et al. 2001; Arora and Gambardella 2005; Carmel 2003; Commander 2005; Correa 1996; Heeks and Nicholson 2002; Tschang 2003). Studies focus on different variables reflecting differences in development of latecomer software industries in the range of products and services offered, market orientation, models of development (e.g. outsourcing-driven, development of own products and services, domestic-oriented vs. export-driven, etc.), revenue, growth, skills and abilities, etc.

Despite the extensive research on development of latecomer software industries, a major gap in our understanding exists. Notwithstanding the growth of software development activities in the latecomer context, few latecomers have managed to develop software products and services on their own and there are single exceptional cases of latecomer companies that have successfully launched own products and services in the global software markets (this point will be expanded in the discussion later on). This points out that latecomer software companies possess limited technological capabilities[2]. A recent research on latecomer software industries (Arora and Gambardella 2005) has pointed Israel and Ireland as examples of latecomers that have successfully entered the global software markets. However, this paper argues that despite being latecomers in entering the software industry, Israel and Ireland are advanced countries and thus the context is rather different from the context in the rest of the countries in Arora and Gambardella's study, which are latecomers, in particular India, Brazil and China. Therefore it is meaningful and important to distinguish between advanced and latecomer context.

Software production is by definition an innovation activity because it aims to produce new products or new ways of executing known tasks and functions (Torrisi 1998). To undertake software activities, companies need to possess capabilities to innovate. However, development of innovation capabilities is neither automatic nor certain. The literature on technological capability reveals that innovation capabilities are developed gradually and as a result of cumulative effort in passing through stages of gradually increasing technological sophistication (Figueiredo 2001). Applied to the software industry, this suggests that successful development of latecomer software activities requires

[2] The fact that few latecomer companies succeeded to enter the global markets holds not only for software activities but also for the whole range of activities of new technology based firms, which illustrates the difficulties in building technological capabilities in new technological areas, and also influence of other entry factors and developments.

accumulation of technological capabilities for software production in the latecomers[3]. Therefore, the development of technological capability has to become a focal point both for companies engaged in software development activities and countries that are aiming to develop software industries to harness their potential in fostering economic development.

Despite the extensive studies on latecomer software development activities the focus has seldom been placed explicitly and systematically on the issue of technological capability, as the critical literature review in this paper reveals. The limited number of studies that did so either have not been well-recognised or have some limitations. This paper disentangles the complexity in analysing technological capabilities in latecomer software companies and develops a framework for analysing capabilities in latecomer software companies with a special focus on technological capabilities.

The paper is structured as follows. The following section 2 makes an overview of the existing literature on capabilities in latecomer software companies. Section 3 presents the proposed approach for analysing capabilities in latecomer software companies. Section 4 discusses diverse learning paths pursued by the latecomer companies in developing software activities. The final section 5 draws conclusions and outlines directions for further research.

2. Existing literature on capabilities in latecomer software companies

Schware (1989; 1992), Correa (1996), and Heeks and Nicholson (2002) have identified capabilities as critical factors enabling latecomer software companies to enter international markets. Some of the recent studies investigating the remarkable expansion of indigenous software development activities in the recent decade in a number of latecomer countries, like India, China and Brazil (see for

[3] Studies about technological development in latecomers have repeatedly revealed that to be able to develop innovation capabilities the latecomers have to engage in a deliberate effort of technological capability building. In exceptional cases perhaps there are instances where a brilliant software solution can be developed by a 'lone inventor' but this is more likely to happen in advanced-context companies rather than latecomers. Studies in technological capabilities in latecomer software companies are limited to the moment and no such case has been identified yet.

example among many others, for all latecomer countries (Arora and Gambardella 2005; Carmel 2003; Commander 2005; Heeks and Nicholson 2002), for India (Arora et al. 2001; Athreye 2005; Desai 2005; Tschang 2001), for China (Saxenian and Quan 2005; Tschang and Xue 2005), for China vs. India (Contractor 2004; Tschang 2003), for Brazil (Behrens 2005; Botelho et al. 2005), also have mentioned capabilities as important drivers in the latecomer software industries development. However, most of these studies have focussed mainly on capabilities for software engineering without disentangling further the capability issue (the few exceptions are outlined below).

A recent book by Arora and Gambardella (2005) analyses the underpinnings of the successful development of the software industries in several latecomer countries, among them India, China and Brazil. Alongside the specific developments in the individual countries, the study outlines the driving forces in the development of latecomer software industries. Capabilities emerge as important drivers underlying the success of these latecomers, as emphasised in the individual countries' chapters (see (Athreye 2005), in particular; also (Botelho et al. 2005; Tschang and Xue 2005) and in the conclusions (Arora 2005).

Despite the recognition and highlighting of their importance, the analysis provides neither a detailed nor a unified framework for analysing capabilities. In the individual chapters, the analysis of capabilities is combined with many other factors affecting industry development, and it is the sources of the incubation of capabilities that are the focal points of the analysis, rather than the capabilities (with the notable exception of Athreye's contribution discussed below). A similar bias applies to the conclusions, which despite emphasising the importance of firms' capabilities focus on the sources of firms' capabilities rather than on capabilities themselves (Arora 2005). An explicit framework considering the specifics of technological capabilities building in the software industry and a connection with the literature in the field of technological developments in latecomer contexts are both absent from most of this study.

The exceptional nature of the study by Athreye (2005) deserves further attention. Although it does not provide an analytical framework nor does it explore the issue of technological capabilities building systematically, it does capture and portray the underlying idea of technological capabilities building. Exploring the development of the Indian software industry and the success factors contributing to its development, Athreye (2005) observes that the evolutionary development of capabilities underpins the Indian success. The study reveals that Indian companies had entered the international markets by providing basic programming skills, but over time they had developed capabilities for software process management, and in a few cases expertise in specialised domains. Athreye concludes by emphasising that the Indian model is a specific example; its success lies on the winning combination of developing different variants of the outsourced service model and evolving organisational capabilities for software process control and large-scale labour management (Athreye, 2005: 36). In this sense, it can be perceived as a specific and exceptional case of development of a latecomer software industry.

A range of paths, including outsourcing or developing own products and services for domestic or international markets, lay before latecomers, as already outlined. Athreye's focus on outsourcing software products and services is just one of these paths. In the case of outsourcing for multinational enterprises (MNEs), capabilities building would be heavily influenced by learning spillovers from MNEs. Different paths may well require and call upon different capabilities, which latecomer companies need to master. For example, outsourcing might require a set of skills, which are limited and significantly narrower than the set of skills required for companies to produce their own products and services. In this sense, the question about technological capabilities in the latecomer software companies still remains to a great extent open.

Tschang (2001) provides a comprehensive software development model. He employs a typology of software development activities in the software development lifecycle that corresponds to

successive/different phases of the product lifecycle. The model outlines five major software development activities (below in italic type) paralleling four product development phases:

1) new product development phase:
    1.1. conceptualisation: requirements analysis and design;
    1.2. (initial) software engineering: system analysis and software engineering, coding and programming, and testing;
2) installation phase:
    2.1. customisation;
3) after sales phase:
    3.1. maintenance: operations and servicing;
4) expiration phase:
    4. 1. product code updating/versioning/improvement.

This is a valuable contribution, as it indicates the intersection between the software development lifecycle and the software product lifecycle. However, the model does not disentangle the issue of the technological sophistication required for software development activities it identifies nor does it investigate further the links of these activities to outsourcing, although it does identify differences in value added associated with different activities. Further, as the author acknowledges, further work is needed to break this model down into products and services, to determine different individual activities and skills needed for each type of activity, to distinguish different types of activities and firms, etc.

Tschang identifies a set of skills that the latecomer software companies need to develop (Tschang, 2001, pp. 19-20). They are classified in two major groups, product development skills and business development skills. Four categories of skills are identified in the product development skills: 1) basic technical skills such as coding and programming languages; 2) system skills including project management, requirement analysis and systems analysis; 3) advanced or high technical skills including mathematical abilities and other fundamental (scientific) knowledge used in science and innovative product development; 4) innovative technical skills, which are creative, interdisciplinary and other skills needed for new product innovation. Under business development skills the author identifies two groups of skills: a) entrepreneurial skills including various management and networking skills, e.g.

sourcing of venture capital, managing a start up, forming alliances, etc. b) other conceptual skills including new products requirements analysis, knowledge of market and customer needs, and innovative and creative abilities.

This is a valuable pioneering contribution with respect to analysing the capabilities that latecomer software companies need to muster. Nevertheless, it has two major limitations. First, despite the fact that the list of technical skills provides a relevant account of the technical capabilities involved in software production, it is not specific in identifying advanced and innovation capabilities.

This paper adopts a different approach to identifying technical capabilities for software development as compared to Tschang (2001). The outlined array of technical capabilities reflects both the specifics of the software production, and the writings in technological capability building and business literature. In this sense, this study attempts to provide more practically-oriented advice and a better representation of the array of capabilities that latecomer software companies have to muster in order to develop technological capability. Second, the list of business skills provided by Tschang (2001) is generic and does not take into account the specificity of the technological development in the latecomer context and the specific organisational capabilities mentioned by several studies of latecomer company development. Furthermore, both in technical and organisational aspects Tschang (2001) outlines skills rather than capabilities and this by itself has conceptual limitations. To provide a better account and to deliver more valuable advice to latecomer software companies there is a need to elaborate the list both with respect to the technical and organisational capabilities, which is done in this research.

In a later study (Tschang, 2003), the author focuses explicitly on capabilities of the latecomer software industries, looking at the case of the Indian and Chinese industry. He provides a list of items, namely individual technical skills, process maturity, management capability, technology, revenue model and product marketing capability. Despite being relevant and informative about the state of

development of a latecomer software industry this list looks at individual technical capabilities and at technology but does not disentangle the technical capabilities needed for software production (for example, capability for software engineering, design, etc.). It also ignores a substantial number of important capabilities, like for example capabilities to monitor technological development and identify potential niches, capabilities for strategic thinking, linkage capabilities, capabilities to establish dynamic organisational learning environment, etc. Further, from technological capability point of view this model provides a mix of capabilities (see above) and performance indicators (e.g. revenue model), without being exhaustive and without being clear how these are derived (although most of them are indeed relevant).

In discussing the development of the Rumanian software industry Grundey and Heeks (1998) employ a theoretical framework based on the concept of technological capabilities and provides a taxonomy of the software technological capability. The study makes a valuable contribution, as it outlines different software production activities representing different phases in climbing the technological ladder to perform more sophisticated software production. It outlines activities underlying production and non-production software capabilities and it makes a comprehensive analysis of the process of moving up from the simple software production to software redesign and reaching skilled software production (Grundey and Heeks, 1998, p. 11). It classifies software activities in seven levels: level one and two include non-production activities, and level three onwards outline the production activities. Level three represents basic production of making copies of an existing software product; level four includes adaptation without production (e.g. create a situation-specific application from a package); level five is simple software production (e.g. creating a new set of interfaces for users, creating a program to move date between applications, creating a small utility program, modifying an existing program to meet user needs); level six involves software redesign (e.g. redesigning a program to meet local user needs, redesigning a program to meet regional/global user needs, and minor process

change (i.e. modifying the software production process); level seven, the highest level in the classification, represents the skilled software production: local product innovation (e.g. developing a new program to meet local user needs), international product innovation (i.e. developing a new program to meet regional/global user needs), major process change (i.e. redesigning the software production process), and process innovation (i.e. designing a completely new software production process).

This model offers a comprehensive account of the wide variety of software activities. However, it includes both non-production and production activities. As the aim of the analyses of technological capabilities is to capture the level of technological sophistication of the software production in the companies, the analyses will be focussed on the levels five onwards in the classification. This part of the classification (e.g. level five to seven) is incorporated in the framework of this research. Further, despite being valuable and operational Grundey and Heek's approach focuses in essence only on a range of software creating activities constituting what we can define as a 'software development technological ladder' but does not analyse the wide arrays of capabilities underlying these activities. In this sense, it does not explore the capabilities, which allow the companies to execute the outlined activities and to build technological capability. However, to have much practical value analyses should scrutinise the constituent capabilities underlying the development of technological capability. Further, in Grundey and Heek's study the theoretical framework seems decoupled from the empirical section, which explores predominantly the institutional foundations (and their transformation) and briefly touches upon the development of software activities in the latecomer software industry in Romania, and thus does not provide a clear approach of how to apply the proposed framework.

The critical review of the existing studies about capability building in latecomer software industries pinpoints a major gap in our understanding about capability building in the latecomer software industries. The predominant part of the studies do not investigate the capability issue in the

light of technological capability building. The limited number of studies that have placed the analysis within the framework of technological capability building or have explicitly focussed on investigating capabilities in the latecomer software industries have limitations, which have been discussed above.

This paper investigates capabilities with a special focus on technological capability. In the following section it discusses the specifics in analysing technological capability in latecomer software industries, followed by a section in which it analyses a wide array of technical and organisational capabilities associated with development of technological capability in latecomer software companies.

3. Investigating technological capabilities in latecomer software companies: a proposed framework

To have the capacity to investigate the software technological capability the analysis has to incorporate the main ideas in the field of technological capability building[4]. The framework has to investigate both the level of technological sophistication of innovation capabilities, which the latecomer companies have managed to develop, and the underlying constituent capabilities. It also has to take into account the specifics of technological development in the latecomer context and the specifics in analysing the development of the latecomer software development activities. Further, it has to scrutinise the learning efforts and the upgrading trajectories they unfold. Combining all these aspects

[4] The current enquiry adopts the main insights advanced by the literature on technological capability building. Based on the main propositions in the field and the critical review of the literature, the requirements for a framework of capabilities building are that it needs to consider that: 1) technological development in latecomer has its specifics, which has to be taken into account; 2) every sector reveals sector-specific features and trajectories of technological development, which have to be taken into consideration; 3) technological capability comprises a wide array of capabilities; 4) innovation capabilities develop gradually by passing through subsequent stages of increase in technological sophistication of the accumulated capabilities; 5) analyses have to investigate both the level of technological sophistication of the routine and innovation capabilities, which the latecomer companies have managed to develop, and the underlying constituent capabilities; 6) alongside technical capabilities the technological capability includes capabilities that are organisational not technical in nature; 7) the organisational capabilities shapes the dynamics of the learning and development of technological capability; 8) technological capability building reveals systemic features; interdependencies occur among the development of the constituencies; 9) learning and technological capability building is a process of passing through different stages of technological maturity, i.e. upgrading; 10) the learning efforts in the company are the major driver for technological upgrade; studies have to scrutinise the company-specific learning mechanisms and capabilities that drive the technological upgrade.

the analysis will have the capacity to grasp the complexity in developing latecomer software industries and to deliver much practical value.

This study develops a framework for analysing technological capabilities in latecomer software industries by taking all the above points into account. It takes into account the contributions of the previous studies about capabilities in latecomer software industries (Athreye 2005; Grundey and Heeks 1998; Tschang 2001; 2003). Studies have highlighted that organisational settings in the companies shape learning and development of technological capability (Kim 1997; Kim and Nelson 2000; Leonard 1995; Marcelle 2004). Therefore it is important for the latecomers to develop enabling organisational arrangements. The studies have stressed various organisational aspects and this research combines them to produce a list of organisational capabilities that the latecomers should muster. An important conceptual issue is still open in the literature and it concerns the interface between organisational capabilities and the technological capability. Another paper (Rousseva 2008b) makes a critical enquiry in that direction. For the purpose of this paper we will accept that organisational underpinnings are critical drivers shaping the development of technological capabilities.

The framework first disentangles the specifics in analysing technological capabilities in latecomer software industries. Following that the framework attempts to identify the major capabilities, which the latecomer software companies have to muster to achieve the highest level of technological capability. Then it discusses the diverse learning paths pursued by the latecomer companies in developing software activities.

Exploring the technological capabilities in latecomer software industries presents a challenge. So far, studies analysing the process of technological capabilities have been predominantly focused on the industrial sector, studying development of the electronics industry (Gee and Kuo 1998; Hobday 1995; Kim 1997; Mytelka and Ernst 1998), textiles (Gee and Kuo 1998; Lall 1987), pulp and paper industry (Figueiredo 2001), steel industry (Dutrenit 2000; Lall 1987), telecommunications (Marcelle

2004), and so forth.

As the predominant part of the studies have been directed at exploring technological capabilities in industrial sectors, the analytical framework developed in the field so far reflects the specifics of the industrial sector as contrasted with the service sector and, additionally, specific features of industrial activities such as photolithography in the context of electronic integrated circuits. A study investigating technological capabilities in latecomer software industries needs to take into account the specifics of the software industry, which are discussed below.

3.1. Specifics in analysing technological capabilities in latecomer software industries

This research identifies three major features, which have to be considered in analysing latecomer software development activities.

First, concerning the degree of innovativeness inbuilt in software technological capability. As noted earlier software production in inherently an innovation activity (Torrisi 1998). Therefore, when analysing the technological capabilities in the software industry, the distinction between production capabilities and innovation capabilities becomes blurred. To produce software solutions latecomer companies must possess some capabilities to innovate.

In order to assess the abilities of the latecomer companies to innovate the analyses should take into account that the *degree of innovativeness* in the latecomers' software production may vary and therefore should scrutinise the degree of innovativeness inbuilt in the products and services offered by the latecomer companies. The significance of novelty, which governs the *extent* of innovation capability needed, varies among different software projects. This point is particularly important in studying the latecomer software industry. It requires the research to disentangle the software innovation process and to classify the software production in terms of its degree of innovativeness.

Before disentangling the degree of innovativeness in the software production, it is worth

clarifying our treatment of innovativeness. Usually the innovativeness is measured with a reference to the novelty to the world market. Following this logic, only products that are successful in the international markets involve high degree of innovativeness. However, innovativeness is also associated with the commercial value it produces. It is usually held that commercialisation in the international markets has a potential to reap higher profits than commercialisation in the domestic market. But this may not always hold. For example, a company may attain higher commercial value by creating an innovative product meeting the needs of a large number of customers in the domestic market (for example payroll and tax record keeping system reflecting the local regulations) in comparison with a company that develops a niche product for a limited number of customers in the international markets. Bearing this duality in mind we will treat innovativeness as novelty to the world, as the aim of the analyses of technological capability is to assess the technological sophistication, which the latecomer companies have achieved.

Different types of software production activities involve different level of innovative efforts and respectively result in software products and services with different degree of innovativeness inbuilt in them. Software services like re-coding legacy applications into more modern computer languages, data migration, or resolving specific incompatibilities between similar systems, and so forth, involve a relatively small innovative component. On the other hand, producing sophisticated software customised services and software packages usually involve higher degree of innovativeness. For example, to successfully launch an ERP or CRM system, or e-commerce solution, a latecomer software company needs to deploy sophisticated knowledge and expertise, and to offer a solution, which is comparable to the frontier technological developments in that particular domain.

In analysing the software activities undertaken by the latecomer companies it is important to scrutinise the degree of innovativeness inbuilt in them and the degree of sophistication of innovation capabilities deployed. When studying the degree of innovative efforts associated with producing

particular software products or services, it is meaningful to distinguish between minor, moderate and major innovation, which are respectively associated with capabilities for minor, moderate and major innovation. This can be viewed as a 'software development technological ladder'.

This distinction follows the classification of software technological capability developed by Grundey and Heeks (1998) and corresponds to the activities classified in level five to level seven in their classification. Simple software production, i.e. software activities like creating new set of interfaces, data migration, creating small utility program and/or modifying an existing program to meet user needs involve a small innovative component and signify existence of capabilities for minor innovation. Software redesign activities like redesigning a program to meet local user needs and redesigning a program to meet regional/global user needs (i.e. customisation and/or localisation), and minor process change (i.e modifying the software production process) indicate capabilities for moderate innovation. Skilled software production activities like local product innovation (i.e. developing a new program to meet local user needs), international product innovation (i.e. developing a new program to meet regional/global user needs), major process change (i.e. redesigning the software production process), and process innovation (i.e. designing a completely new software production process) suggest capabilities for major innovation.

The proportion of activities like re-coding, data migration, resolving incompatibility, and so forth, can be expected to account for a significant share in the software services offered by latecomer companies. On the other hand, the presence of major innovative activities, like creation of packages or sophisticated customised services, despite their small share in latecomer software developments, signals the existence of potentially significant innovation capabilities in the latecomers. For example, if many latecomer software development activities are directed at offering services in the domestic markets and there is also a growing share of performing outsourcing services for international markets, this would indicate capabilities for minor and eventually moderate innovation; India has specialised in

offering software services in the international markets, and if we apply the classification of the degree of innovativeness to the range of software development activities which the Indian software companies offer according to the literature (Arora et al. 2001; Athreye 2005; Desai 2005) it will reveal the existence of capabilities for minor, moderate innovation and in some though a limited number of cases capabilities for major innovation. China and Brazil have developed software products and services for the domestic markets (Behrens 2005; Botelho et al. 2005; Saxenian and Quan 2005; Tschang and Xue 2005), which suggest capabilities for moderate and major innovation. The above is a general overview of the accumulated technological capabilities in latecomer software industries based on the existing studies but further in-depth, case study and comparison-based analyses are needed to reveal the achievements and problems in development of technological capabilities. These should be done by comparing different companies in one country and comparing companies in different countries to capture both company- and context-specific issues.

Second, reflecting the breadth of capabilities. The discussion about the degree of innovativeness inbuilt in software technological capability in the previous point reflects also on the breadth of the technological capability. Higher technological sophistication usually entails deeper (i.e. more sophisticated) capabilities and a wider range of (i.e. broader) capabilities. This illustrates the idea that technological capabilities develop by deepening and broadening of capabilities and involve moving upwards the technological ladder (Hobday 1995). It reveals that capabilities develop in a sequential manner and this makes it important for studies to analyse that sequencing and the development of a deeper and broader set of capabilities.

Provided that the nature of innovation is similar across sectors, then the capabilities literature suggests that major innovation require execution of a greater variety and also more complex software development activities as compared to capabilities for moderate and minor innovation. For example, creating a product innovation involves broader and far more complex capabilities than the software

redesign or the simple software production. Similarly, capabilities for moderate innovation entail bigger variety and more complex software activities as compared to capabilities for minor innovation. Thus for example, software redesign (e.g. redesigning a program to meet local or global user needs or minor process change) require greater capabilities than the simple software production of creating new interfaces, small utility program or program for data migration, modifying existing program, etc. Therefore, higher degree of innovativeness entails broader and more complex capabilities, i.e. the breadth of capabilities increases with the increase in the innovativeness.

High technological sophistication entails sophisticated capabilities. The following section in this paper explores a wide variety of capabilities, which the latecomer software companies need to muster to reach the highest level of technological maturity, which is to launch own products in the international markets[5]. In this sense it attempts to outline the major capabilities, which the latecomer software companies have to muster to achieve the highest level of technological capability. Therefore, the framework portrays the major capabilities involved in the software development technological ladder, from minor innovation to major innovation but further empirical analyses have to be conducted to reveal the links between the degree of innovativeness and the underlying constituent capabilities. It is worth underlying that development of this framework for analysing technological capabilities in latecomer software companies is derived from premise-based logic and analogy, but this needs to be tested in a specific contexts to explore the links between the degree of innovativeness and the underlying constituent capabilities, and to see whether it holds true for the industry in every developmental context.

In identifying the wide arrays of capabilities, which the latecomer software companies need to

[5] Strictly speaking the highest level of technological maturity is the process innovation, e.g. designing a completely new software production process. However, reaching this level is a daunting task even for advanced-context companies; for latecomers is almost unattainable. Therefore, for the purpose of this analysis and to make the study more operational it can be assumed that the highest level of technological maturity, which the latecomer software companies might be aiming to achieve is the launching of their own products or services in the international markets.

build the study draws on insights from studies on software engineering, development of latecomer software industries, technological capability building literature, etc.

There is one other aspect worth mentioning in discussing the level of technological sophistication of software production and the sophistication of the underlying capabilities. There might be cases where 'low complexity' innovation represents a major innovation. For example, development of the 'bubble sort,' a relatively early and rather problematic 'innovation' widely used in teaching computer science, represent a 'low complexity' innovation which is in essence a 'major innovation' because it is a conceptual breakthrough with regard to existing state of the art. In the context of assessing technological capabilities in latecomer software companies a conceptual breakthrough (e.g. algorithmic invention) by itself does not constitute an innovation because it is not embedded in a commercial product. Nevertheless, any presence of technical expertise of that kind signals the existence of high level of technical potential that if harnessed successfully can generate innovative software products and services.

Third, in a globally competitive context, the variation in technological sophistication of software development activities and the implications for the organisation of software development is relevant. Software production is carried out in a cycle of software development activities, as described by software engineering literature, and involves a range of software development activities. It is appealing to explore further two issues: the variation in technological sophistication of software development activities and the implications for the organisation of software development.

The activities involved in software development lifecycle vary in terms of their degree of technological sophistication. Some activities like for example analysis of customer needs, system analysis, design, etc. require sophisticated technological knowledge, while others like coding, testing, maintenance are less sophisticated.

The table 1. in the Appendix below classifies major types of software development activities in

the software lifecycle according to their technological sophistication.

Software development activities along the lifecycle can be partitioned and split into different software projects to be executed by different software development teams. Sophisticated software development activities, like requirements analysis and design, can be executed by advanced-context companies, while less sophisticated software development activities like coding, testing and maintenance can be outsourced to latecomer companies to leverage on cost advantage in the latecomers. Recent years saw an upsurge in the outsourcing of software development activities to the latecomer companies. The increasing acceptance of this new business model of organisation of software development makes relevant the examining of its complexity (e.g. technological sophistication of the outsourced activities, etc.) and its prevalence. This is a particularly important point when analysing development of latecomer software activities.

A number of latecomers have developed outsourcing-driven software development activities. The question which arises is what kind of software development activities these involve, whether they involve only software development activities with lower technological sophistication or they also include other more sophisticated software development activities. For example, the predominant part of software development activities outsourced to latecomer companies might involve coding, testing and redesign because requirements for these activities can be concisely and precisely described and incorporated into business contracts. At the same time, studies have revealed that the Indian companies have managed to move up the technological ladder and increase the technological sophistication of their production by shifting from coding and testing to domain expertise (Athreye, 2005). Therefore, in analysing the development of latecomer software industries, studies need to investigate both the range and the type of software development activities undertaken by the latecomers and to scrutinise their technological sophistication.

3.2. Disentangling the constituent technical and organisational capabilities

To be able to produce software products and launch them successfully in the international markets, a latecomer company needs to develop an array of capabilities. Table 2 in the Appendix summarises the technical and organisational capabilities underpinning the accumulation of technological capability, which the latecomer software companies need to develop. The subsequent sub-sections examine the two sets of capabilities, technical and organisational, and within these sub-sections each of the listed capabilities is briefly discussed.

3.2.1. Technical capabilities in latecomer software companies

A number of technical capabilities are involved in developing software technological capability, many of which have been identified by existing studies on capability development in latecomer software industries and in the software engineering and business literature.

Technical capabilities can be divided in basic, intermediate and advanced categories. The underlying idea behind this categorisation is the existence of thresholds in knowledge accumulation and capability development. This point is particularly important for the latecomer software companies, as they may face difficulties in accumulating not only advanced capabilities, but also intermediate and even basic capabilities, and this will affect directly their possibilities to develop indigenous software activities on their own and sustain them over time without being trapped in low and narrow capability trap.

Basic technical capabilities

The capabilities, which constitute the minimal core technical capabilities, which any software company needs to possess in order to undertake software development activities, are classified as basic technical capabilities. The first basic technical capability, the capability for software programming, is

associated with the phase of actual writing of the software. Knowledge about software programming languages and techniques, and platforms creates the base for creating software code. To develop mastery, however, latecomers need to develop deep understanding about software programming. They need to build expertise about algorithms, software programming languages, and so forth. Building expertise in scripting languages for WWW data (for example Java, XML, JavaScript, HTML, CSS, XSL, XSLT, C++, C and Object-C) requires developing understanding about their comparative utility for specific application contexts. Further, mastering one of these languages requires developing deeper understanding about it. For example, to develop excellent programming skills the latecomers need not only to learn the syntax of the Java programming language, but also object-oriented programming with the Java programming language; creating graphical user interfaces (GUI), exceptions, file input/output (I/O), threads and networking.

The second basic capability for software development is testing and high quality assurance. The latecomer companies need to deliver high quality products and services, if they are to be successful. To produce high quality products and services latecomers need to build capabilities for software quality assurance in which reliability is the central issue. Developing capabilities for software quality assurance is a focus of attention among practitioners and academics, and has resulted in creation of quality assurance guidelines, reflected both in ISO certification scheme and the Capability Maturity Model Integration (CMMI) assessment scheme. Whether certified under one of these schemes, or not, latecomer software companies need to apply software quality assurance techniques to ensure the quality and reliability of the software they deliver. Rigorous testing, de-bugging and defect elimination are critical steps in that direction.

The basic technical capabilities are relatively easy to obtain as they contain high codified element and most of latecomer companies entering the software development activities pass successfully this stage. The basic technical capabilities alone are insufficient for a software company to

establish itself, and they need to be augmented further with intermediate technical capabilities.

Intermediate technical capabilities

The intermediate technical capabilities and the capabilities for project and software process management in particular, represent an important threshold in capability development for the latecomers to be able to adequately perform technical tasks and to become competitive. Development of the intermediate capabilities enables the latecomer companies to manage systematically the software development activities and thus lays foundations for accumulation of further more advanced expertise.

Among the most important intermediate capabilities are the capabilities for project management. Every company has to develop abilities to effectively organise and manage its projects. Companies have to develop skills to split the software development activities in a project into different tasks and to gather teams to execute them. Good project management requires gathering teams with adequate expertise and ensuring efficient workflow. Project management becomes more complex nowadays due to the increasing complexity in the organisation of the software development. As already mentioned, the modern model of organisation, where software development is distributed among several firms, some of which latecomers, is becoming more common at present. This new trend makes project management capabilities critical for the latecomers. Project management consists of both technical and organisational aspects. The organisational aspect will be discussed in the following section, which makes an overview of the organisational capabilities.

Apart from project management capabilities, the latecomer software companies have to develop capabilities for software process control, which are identified as the second intermediate technical capabilities. Every software company should tailor its software production to achieve efficiency gains. Academics and practitioners have dedicated considerable efforts in developing models for assessing the effectiveness of the software production processes, and Capability Maturity Model Integration (CMMI)

and ISO are the most influential among them. To achieve efficiency gains a software producing company should go beyond the stage of craft and ad-hoc software production and should make deliberate efforts to streamline its production process.

Although there is a high intangible component associated with it, software production permits measurability and tailoring, and achieving better efficiency. During testing and de-bugging phase mistakes should be reported and instructions put in place to ensure that these are avoided in future. This suggests that software companies have to monitor constantly their production activities and evaluate their efficiency. Further, the software code written for one project can be stored for eventual future use.

Software production allows for re-usability of the software code, which creates opportunities for considerable efficiency gains. Developing capabilities for software process control is not automatic in general and is far from straightforward for latecomer companies. Therefore, latecomer software companies have to put deliberate efforts to develop capabilities to control and manage their software production processes. An increasing number of latecomer software companies are undergoing ISO and CMMI certification to reveal their technological potential. Studies (Arora et al. 2001; Athreye 2005) have emphasised that certification has been extensively used by the Indian companies to strengthen their competitive stand. Nevertheless, certifying and developing capabilities in project and software process management remains a challenging task for latecomers.

The next (optional) intermediate capabilities are the capabilities for operating in various environments. The latecomer software companies may need also to build capabilities in various operating environments, to be able to respond to various clients' needs and requirements. Although Windows operating system is the dominant computer platform, other operating systems like Linux, Unix and Mac are also used. Therefore, the latecomer software companies need to be aware that opting for learning software design, programming and networking only on the dominant platform instead of building expertise in all existing platforms may involve costs or lost business. This type of business

loss is not necessarily fatal and can be perceived as an incentive for companies to broaden their expertise by learning to undertake software activities also in other platforms, to be able to better respond to customers' requirements. Development of capabilities for operating in LINUX environment is with increasing importance, as the free libre and open standard software (FLOSS) solutions are becoming more popular recently. Alongside some companies, FLOSS have been adopted by some public administrations and public education systems, which signals a growing presence of FLOSS, which may open new opportunities for latecomer software companies.

Creating a modern software product or service very often requires capabilities for networking applications and these are identified as the next (optional) intermediate capabilities. The networking applications include any kind of software solution, which operates in a network environment. Wide ranges of software solutions nowadays involve networking. For example, the information system in a company represents a network (that is intranet). Simultaneously, the companies are using software solutions operating in Internet (that is web-based solutions), while executing their every day operations for document sharing, coordination, communication, payment, and so forth. Therefore, the software company, which develops a new software solution for a client, needs to take into account the access by multiple users to the network resources such as files and to ensure security over the access as well as dealing with the need for 'file locking' to prevent simultaneous editing by different users of the same data. To be able to develop reliable networking applications the latecomer software companies need to build expertise in security engineering. They need to develop understanding about the network operating systems, security protocols, techniques for specifying and implementing a security policy, and so forth. This may present a cumbersome task for the latecomers, as network security is among the most dynamic fields, which has been rapidly developing recently.

Advanced technical capabilities

The last threshold in development of capabilities for software production involves development of advanced capabilities. Basic and intermediate capabilities provide a base for undertaking software development activities, but development of advanced capabilities reveals a qualitatively different stage in capability development, as it involves a mastery over capabilities with high technological sophistication.

The first advanced capability is the capability for software design. It is associated with the phase of inception of software development, when software companies need to design how the software will look and perform based on clients' requirements. Software design is sophisticated software activity involving a number of interrelated steps of analysing the software domain, identifying the software specifications, analysing thoroughly customers' needs, designing the software architecture, system analysis, etc. Software design activities are complex, as they involve a number of sophisticated activities and determine the software characteristics. Due to their complex nature capabilities for software design are difficult to muster, as it might be a daunting task for latecomer software companies to develop an in-depth expertise in all these areas.

It is also vital for latecomers to establish specialised expertise and depth of understanding in a particular domain, which may be identified as the second advanced technical capability. Developing a deep understanding and expertise about frontier technologies is particularly challenging for latecomer companies as they are embedded in a latecomer context, and therefore, developing expertise by focussing on a particular domain appears to be a more realistic strategy for the latecomer companies to enable their attempts for technological upgrading. For instance, good knowledge and understanding of finance and banking system is required, if a software company is to create finance or banking software solution. The task becomes even more daunting, if a latecomer company seeks to develop finance or banking solution for international markets, where operations are far more complex and sophisticated;

and therefore, the latecomer company needs to put deliberate efforts to develop deeper understanding about its specifics.

A second example is the development of ERP systems. To be able to build an ERP system, a latecomer software company needs to develop an understanding about not only the structure of the ERP systems but also knowledge and understanding of corporate practices in a particular market and a particular industry in which the system is to be employed. Some though a limited number of latecomer companies have already managed to establish domain expertise, like for example Brazilian and Indian companies have developed expertise in the financial domain (Athreye 2005; Botelho et al. 2005).

The capability to diversify the products and services (the third advanced technical capability) is optional and perhaps rather difficult to achieve by latecomer software companies. The software companies are specialised suppliers according to Pavitt's (1984) taxonomy, that is companies providing specialised products. Being specialised suppliers does not preclude diversification. Companies can diversify their products and services within their specialised niche. Diversifying the range of products and services, which a company offers, creates an opportunity for companies to reap greater benefit from their knowledge base by applying it in different contexts. Once the latecomer companies have developed expertise in a particular area, they may decide to broaden the range of products and services they offer, based on the experience and knowledge they have, or market opportunities they perceive. The degree of diversification may vary. Diversifying by introducing new products or services, which require the latecomer company to enter a new domain, which is completely separate from the domain in which the company has accumulated expertise so far is associated with high risk for any company, especially latecomers. On the other hand, diversifying by introducing new products or services, which are close to the existing range, and to the already accumulated expertise in a particular field, is far less risky endeavour. For example, developing a modular web-based platform for automated billing, invoicing and customer management for the one spectrum of services, for example Voice-over-IP

involves lower technological efforts and leads to lower degree of diversification in a company which has previously developed a modular web-based platform for automated billing, invoicing and customer managements for a spectrum of services like Internet, triple play, WiMAX and regular voice communication services. Introducing systems able to address all of these applications will require significantly higher technological efforts and will result in a higher degree of diversification for a company than simply developing an automated billing application for one application. Such diversification may, however, be necessary to provide a competitive offering to those being offered by rivals. This sort of diversification is a difficult task for latecomers, given that they need to maintain the depth of expertise in a particular domain and, at the same time, to broaden the range of the products and services within that domain and, perhaps, also to broaden the expertise in different domains.

However, a strategy of diversification is difficult to achieve even by an advanced company and therefore a strategy for specialisation remains a more realistic strategy for latecomers. It remains hard for latecomer companies to succeed in diversification, due to the limited access to frontier technological knowledge and knowledge of the application domain in other countries they have, being embedded in a latecomer environment. In fact, capabilities for diversification are the only among the outlined ones which remain as an optional theoretical but a bit difficult to defend as a realistic and feasible proposal for latecomer software companies. On the other hand, it depends pretty much on the nature and degree of diversification. For example, many latecomer companies operating only in domestic markets diversify the range of activities they offer. For example, it is common for latecomer software companies developing software services and solutions for domestic markets to offer also system integration and even telecommunication services. Therefore, the decision of whether and in what direction to diversify remains in the discretion of the individual company based on its expertise, technological and market positions, and perceived opportunities.

Capabilities to monitor technical change and to identify niches for potential developments are

identified as the fourth and last advanced technical capabilities. They draw heavily on the accumulation of technical expertise and also require some marketing expertise. To become competitive and sustain its competitive position any company (and particularly a company in a high-tech sector) needs to keep abreast with the new technological developments. By deepening their technological expertise and strengthening the sophistication of their technological capabilities over time the latecomer companies should eventually reach a point of being capable of identifying niches for potential new developments and developing them on their own. To achieve that, the companies need to stretch their capabilities beyond mere monitoring of technological development. They need to develop in-depth knowledge and understanding about technological development to be able to identify potential niches for new developments, assess their technological and market viability, and develop them on their own. Building the necessary technical and organisational expertise to master the existing frontier technological developments remains a challenge to the latecomers, and some of them may remain far behind in managing the existing frontier developments. Developing capabilities to identify potential niches for new developments require further deliberate, focused and persistent efforts to expand the technical expertise and organisational abilities, to reach a level of technological mastery enabling companies to foresee trends of development and to identify niches for potential new applications. Being embedded in a context, which does not expose them to the latest technological developments, and being away from lead users, latecomers may find this to be a particularly difficult task.

3.2.2. Organisational capabilities

Recent studies in technological capability building (Kim 1997; Leonard 1995; Marcelle 2004) have disentangled the interface between the organisational settings and the accumulation of technological capabilities. The results have highlighted that the organisational settings in the company shape the development of technological capability and therefore further research has to scrutinise the organisational capabilities in the latecomers. Following the findings of these recent studies the

proposed approach outlines a set of organisational capabilities that have been repeatedly cited as underpinning the development of technological capability and therefore are to be developed by the latecomer companies. This approach has some idiosyncratic elements; although it uses ideas from management literature, it adapts them to the specifics of the latecomer context. It produces a list of organisational capabilities that in some cases may seem simple or obvious from a standpoint of an advanced-context company but for a latecomer company they not be seen as needed or, even if perceived as important, may be difficult to develop, as studies have repeatedly revealed (Ernst et al. 1998; Kim 1997; Leonard 1995; Marcelle 2004).

The approach disentangles the organisational capabilities reaching a high level of disaggregation. Further, this approach places emphasis even on basic organisational capabilities, like organising, communication and control, as the latecomer companies may lack or possess limited capacity even in such basic management capabilities. Although the organisational capabilities are applied to the case of the software industry, they can be easily adapted to any other latecomer industries.

The approach offered here employs six categories of organisational capabilities (table 2 in the Appendix).

The first of the organisational capabilities, the ability to establish organisational culture facilitating learning, can be considered as the most fundamental capability, as it underlies the development of all technical and organisational capabilities. Studies have repeatedly revealed that development of technological capability is conditioned by the existence of organisational culture facilitating learning in the latecomers (Kim 1997; Marcelle 2004). The ability to learn is shaped by company's attitude towards new ideas, and the openness, flexibility and aptitude to accommodate them. It requires establishment of an interactive and open environment in the organisation, which makes it possible to be receptive to new ideas and new developments. The latecomer companies therefore need to establish an environment for accumulation of information, exchange of ideas, and knowledge generation and re-combination.

To achieve this type of organisational culture, the whole organisation needs to be 'tuned' into a learning mode, which requires a degree of openness, exploration and adaptation. Supporting

organisational arrangements are to be put in place to create an open and flexible environment and to enable learning. The organisation should establish flexible structures allowing active information and communication exchange and generation of new ideas. Learning inevitably involves change. Adopting new ideas and developments often necessitates undergoing some sort of change. In this sense, abilities to learn are closely coupled with change management skills. Case studies of successful technological development often confirm that it was indeed the change management capabilities that underpin dynamic technological and organisational learning (Kim 1997; Marcelle 2004).

The abilities to establish organisational culture facilitating learning constitute essential elements of the capabilities for effective management. The former have been outlined separately, as they represent the major driver of the process of technological capabilities building. The latter, the second of the organisational capabilities, is no less important. Every single aspect of the organisational functioning is affected and indeed shaped by the company's management, which makes company's management a critical factor. The company should be managed in a consistent way to achieve effective results. A step further from effective management, i.e. doing the right things, is to manage things efficiently, i.e. doing the things right. The latecomer companies should focus first on developing capabilities for effective management and later on embark on achieving efficiency gains.

Organisational processes, like communication, decision-making, coordination, control, and so forth, are contingent on management decisions and reflect company's strategy. Managerial decisions underlie the vision of what and where the organisation should be in the future and the strategy for pursuing it. To be effective, the management activities should encompass all levels and types of company's activities to create a coherent working environment. A company with effective management involves its members in active communication with each other, participatory decision-making and well-coordinated activities; and team building and working in a team are essential elements in the organisational culture.

Essential organisational processes that must be managed effectively are the organisational design, coordination, communication, organisational integration, change management, etc. For the latecomer companies it may well be a difficult task to ensure the effective functioning of all of these organisational processes, as it requires elaborate organisational skills that they may not possess. For example, the latecomers may find difficulties in setting an organisational design that is most suitable for the task executed by the company. They also may face difficulties in arranging effective coordination across the organisational units and functions. Similarly, rigid communication structures and arrangements may be in place and the latecomers may be unable to dismantle them, which in result will impede the exchange of information and knowledge in the organisation and will affect the upgrade process. Organisational integration is also a complex task, which requires achieving integration among all organisational functions to accrue synergy effects, which may be beyond the capacities of some latecomer companies.

Project management is an essential element in effective management. Project management involves identifying the activities to be undertaken within the project, setting deadlines and creating workflow plans, assigning responsibilities, monitoring the work progress and delivering quality outcomes within the deadlines. To take best advantage of market opportunities, latecomers should develop project management skills to manage both small and large-scale projects. Managing large-scale projects poses greater challenges for latecomers. The large-scale projects may require resources well beyond those under the company's control, as normally the latecomer companies are small-scale and possess limited resources and sometimes even limited expertise. If a latecomer company faces the opportunity of executing a large-scale project, it has to be able to mobilise the necessary human resources and the additional expertise it may need. It also has to be able to coordinate the proliferation of tasks that large-scale projects produce, a process different in degree if not in type from the process of managing smaller-scale projects. Large projects require the efficient location of knowledge and other

necessary resources as well as rapid response and excellence in coordination. Developing capabilities to manage both small and large projects appears one of the critical drivers underlining the success of the Indian software industry (Athreye 2005). Being capable of managing large projects the latecomers are in a possession of a large pool of resources, which they are able to deploy and utilise. When these are coupled with relevant technical expertise and other organisational skills, the latecomer software companies will be in a position to compete.

Abilities for prompt delivery are also important elements in effective management capabilities. Meeting deadlines is crucial, as failure to do so result in increasing project costs and customer dissatisfaction. Delivery on time is crucially important in international markets. In latecomer countries, where the market power of suppliers may be high or where there are high transactions costs of changing suppliers, clients might be more lenient toward delays and this creates a challenge for latecomer companies to learn to execute projects meeting strictly the deadlines. To be able to do that, they need to develop abilities and skills for project management, tracking the work progress throughout the project, clarifying project requirements at the very beginning, effective communication between parties throughout the project, and so forth.

Establishing effective management practices can appear a difficult task for any company, but it is particularly difficult for latecomers, due to their limited business experience. Latecomers need to develop new skills but they also may need to discard existing ineffective management practices. All these require effective change management capabilities. Change management activities affect company's activities in all levels: tasks performed by individuals, various organisational processes (communication, decision making, coordination, control, and so forth), overall strategic vision, organisational culture, and so forth. In order to perform effective change management the organisations need to constantly monitor their activities and the signals by the external environment, and to undertake re-adjustments in the organisation. All above illustrates the complexity, which the latecomers face in

developing effective management abilities, enabling them to master the organisational and technical dynamics underpinning the development of new technologies.

In addition to the above capabilities a third category of capabilities appears to be important - the learning ability of the company. Alongside the establishment of organisational culture facilitating learning as the most fundamental organisational capability, it is also important how the learning effort operates. Studies have emphasised that alongside placing deliberate efforts to upgrade, the latecomer companies need to develop capabilities for expeditious (Kim 1997) and integrative learning (Marcelle 2004) to achieve successful technological upgrade. Learning is the major driving force in the process of upgrade but the appropriate learning effort does not emerge automatically. Studies have emphasised that in order to enhance the accumulation of technological capability the learning effort should be expeditious and integrative rather than isolated and effortless (Kim 1997; Marcelle 2004). The latecomers should develop capabilities to learn in an expeditious manner, which may involve crisis construction to rapidly acquire and absorb and re-combine new knowledge and to unlearn deficient elements. Only focussed, purposeful and expeditious learning effort can generate the required dynamics for successful and swift technological upgrade. In addition to this, to successfully upgrade the latecomer companies need to tap a variety of sources of information and knowledge. Information and knowledge from the global innovation system as well as information from suppliers and/or users should be closely integrated with the internal learning effort (Marcelle 2004). Furthermore, to achieve utmost results of these integrative learning efforts the latecomer companies need to create a right balance between the different learning sources (ibid). In this sense, the ability for expeditious and integrative learning is a fundamental organisational capability, as it induces and shapes the dynamics underlying the development of technological capability. Learning concerns both technical and organisational aspects of upgrade. It involves a search, acquisition, absorption, re-combination and application of technical knowledge. Active acquisition of the latest technological information and knowledge and its

rapid diffusion and re-combination has been the driving force of the successful technological development in East Asia (Bell and Pavitt 1993; Ernst et al. 1998; Ernst and Kim 2002; Hobday 1995; 2000; Kim 1997). Alongside acquisition of technical knowledge the learning efforts should also involve understanding and development of organisational arrangements to support the technological upgrade. For example, crisis construction has been actively used by successful latecomers to rapidly attain technological knowledge and also to re-engineer the existing business processes and develop new organisational arrangements (Kim 1997). This comes to reveal the critical importance of the expeditious and integrative learning effort for inducing the necessary dynamism for technological upgrade.

The fourth organisational capability is the linkage capabilities, which are critical enablers for upgrade in latecomer companies. Establishing links and relationships with clients, suppliers and other parties have been identified as vital capabilities for latecomer companies, as these provide channels for obtaining information and feedback on technological dynamism (Bell and Pavitt 1993; Ernst et al. 1998; Hobday 1995; Kim 1997). User-producer interactions have been outlined as one of the major drivers for generating innovation (von Hippel 1988), as the users possess an in-depth understanding about the work processes, the performance of the existing technologies and potential niches for further developments. Close interactions with clients can generate ideas for modification and improvement in the existing products and services. In the case of latecomers, interactions with clients are even more important, as the latecomers, being embedded in a latecomer context, have limited information about new developments and are away from lead users (Hobday 1995). Therefore, establishing and maintaining links and relationships with clients, foreign ones in particular, are important channels for obtaining information about new technological developments, feedback and identifying potential niches for further developments. Likewise, links with suppliers and other agents in the innovation networks (universities, research institutes, consultants, and so forth) contribute significantly to the

accumulation of information about new developments and the latecomer companies need to be able to tap also the potential of these sources of information. Partnerships with other companies or consultants have proved to be particularly influential for latecomer companies, as they also provide access to valuable pool of knowledge and resources and contacts.

Apart from being valuable sources of knowledge and information, linkages are influential channel for finding contracts. It has been emphasised that the diaspora of Indians working in the US has appeared a critical driver in establishing contacts, getting contracts from the US companies and the subsequent boom in software development activities undertaken by Indian companies. Thus, latecomers should place deliberate efforts to establish links and contacts with foreign companies.

Alongside development of the previous organisational capabilities, latecomer companies need to develop marketing capabilities, the fifth organisational capability. Latecomer software companies need to be capable of identifying potential clients, approaching them, promoting their in-house capabilities for software developments, and maintaining relationships with their clients upon completion of the project for further developments, and so forth. The abilities to market the in-house skills and expertise are crucial in ensuring the latecomers' success. Only successful commercial application can harness already developed technical and organisational capabilities, and allow further expansion.

Developing marketing capabilities is often difficult for any company, but it is very much the case with the latecomers. The latecomer companies may possess (sometimes very) limited knowledge about the structure and functioning of the international markets, which prevent them from being able to identify the right approach for entering a particular market, positioning themselves in the market and identifying the right customers. Further, even if successful in all of the above and having identified the right clients, the latecomers may fail in approaching and establishing contacts with them for a variety of reasons. In the domestic market, the latecomers have access to local customers with whom their

share the same cultural and business background, which makes establishment and maintaining contacts easier. This does not hold for the international customers, where the latecomers need to build relationship, taking into account international business ethics and management, and dealing with inter-cultural differences such as different norms or even beliefs. Thus, developing skills in international business management becomes one of the prerequisites for latecomers' success.

The sixth and last of the critical organisational capabilities is strategic thinking. In their business activities the latecomer companies need to apply strategic vision of where the company is going in a longer term. The ability to think in a strategic way underpins latecomers' success. Ernst (1998) identifies it as one of the five technological capabilities which latecomers need to develop. The managers need to have a clear vision what the company is aiming to accomplish in the future so they prepare and undertake the intermediate steps towards the final goal. In setting their goals the latecomer companies are to take into account their standing with respect to the development of the world industry. Questions like 'where do the products and services offered by the company position with respect to development of the world industry', 'in what direction and how the company can upgrade', and so forth, must be answered to achieve a sustainable position in the international market.

It would be a mistake to assume that the latecomers should aim to position themselves as competing against the world players, as these may be quite unrealistic. As Arora and Gambardella's (2005) study suggests the success in development of latecomer software industries lies in finding a niche in which the latecomers have a competitive advantage. To be able to identify an appropriate niche the latecomers have to have a deep understanding about the technological development of the world software industry and its trends, and the position, which the latecomers hold with respect to technological and organisational expertise and market strategy. Such assessment should adopt a dynamic perspective to re-assess changes over time. Once they have the vision set, the latecomers need to pursue their aim by undertaking the steps leading towards it. These would involve strategic actions

like establishing partnerships, building new expertise and expanding existing knowledge, and the like.

4. Diverse learning paths pursued by latecomer companies in developing software activities

High expectations have been raised about the benefits that the latecomers can reap by developing software industries. A number of latecomers have started developing software industries in the last decade. Despite the attention that has been given to the development of latecomer software activities in the academic and policy literature, an important question remains open. What are the possibilities before the latecomers in developing indigenous software industries: is there one 'best' path to develop software capabilities in the latecomers; are there many possible paths leading to the desired result and it is the analysis of the context that allows one to distinguish better from worse; or are there many paths – some lead upward and some lead off the cliff but all are obscured by clouds of uncertainty so one has to march forward with energy and enthusiasm and accept that one will either become God or a lemming.

The last proposition seems to be less likely – certain instructions are available to the latecomers based on the understanding about the requirements for developing software industries (e.g. building skills base, expertise in software engineering, etc.) and also derived by the experience of others be they advanced-context companies or latecomers (e.g. different paths for software development activities, policy facilitation, etc.). Therefore, notwithstanding the uncertainties associated with every economic initiative, the development of latecomer software industries is not an undertaking driven by mere guesswork and clouded by absolutely uncertainty.

The experience in development of latecomer software industries also reveals that the first proposition is also not viable: there is no one best path; different latecomers have embarked on different paths. Some latecomers have engaged actively in outsourcing for the international markets. Following the remarkable success of India, a number of latecomers have committed to outsourcing.

Another group of latecomers have directed their efforts towards providing customised services for the domestic markets. Few latecomers have focussed on developing software products for the domestic markets. China and Brazil are typical examples in that direction. Very few latecomers have targeted development of software products for the international markets and very few have succeeded.

Apparently there are there many possible paths for development of latecomer software activities and it is the analysis of the context that allows one to distinguish better from worse. The core question is what level of technological sophistication these involve and whether the latecomer companies make deliberate effort to attain higher technological sophistication over time, i.e. to move upwards the software technological ladder. Therefore, the analyses of latecomer software development activities have to be directed at exploring the technological capabilities of the latecomers and their efforts to upgrade.

There are a couple of important points to address in analysing outsourcing software industries. It can be expected that the latecomer companies which undertake outsourcing activities have developed capabilities in software programming, quality assurance and delivery on time, and possibly in software process control. However, there might be other capabilities that underpin the successful development of a software industry in a particular latecomer context and create a competitive advantage and these must be taken into account. For example, Athreye (2005) revealed that in the case of India it was the combination of both new business models and technical capabilities that created a strong competitive position for the Indian companies.

There is one particularly important point with respect to outsourcing-driven software development activities. As already underlined, the outsourcing-driven software development activities may vary with respect to the degree of technological sophistication they involve. Some outsourcing projects may require only simple programming, while others may necessitate deployment of more sophisticated software programming capabilities and even customisation or localisation of existing

software products. Therefore, it is important for studies to scrutinise the degree of technological sophistication of the deployed capabilities and executed activities in the outsourcing-driven software development projects by the latecomer companies.

The major challenge before software companies that undertake outsourcing activities is whether they will be able to sustain their business over time, if their advantage is only factor price-based. Therefore, studies need to assess the degree of technological sophistication of the outsourcing-driven software development activities undertaken by the latecomers and whether the latecomer companies have been able to deepen and broaden their capabilities, which may allow them to upgrade over time and avoid getting trapped in low and narrow capabilities trap.

Few latecomer companies have managed to develop own software products and services and in most of the cases these are commercialised in the domestic market. The software development activities in Brazil and China are typical examples in that direction. As development of own products and services is associated with a wide range of capabilities, it can be assumed that the latecomer software companies that offer their own products and services have managed to develop a wider range of capabilities as compared to the outsourcing-driven firms. The question which arises is what kind of capabilities these companies have managed to muster and whether these will enable them to make a shift to the international markets at some point.

A considerable number of latecomer software development activities involve customisation or localisation. As customisation is a process of tailoring a software product to specific customer needs, it may involve a wide range of activities from simple modifying to thorough redesign. Therefore, it involves a wide range of capabilities that go beyond programming, testing and process control and require capabilities for analysis of customer needs, re-design, possibly system analysis, etc. Customisation or localisation activities are carried out in almost every country, as there is a wide-spread need for adaptation to local needs or local users' requirements. Customisation or localisation

can involve localisation to local language (e.g. Chinese, Cyrilic, etc.) or specific local user requirements (e.g. adaptation of packaged software to the requirements of a particular customer). Often the localisation is undertaken by local software companies, which implies that a substantial part of the latecomer software development activities involve localisation and customisation and thus it is important for analyses to investigate the capabilities deployed.

There are very few examples of indigenous latecomer software companies that have succeeded to enter the global software markets with their own products and these represent single exceptional cases. It has been already underlined above that the examples that have been cited in the existing literature represent advanced- rather than latecomer-context cases. In this sense, the literature so far has not identified cases of indigenous latecomer software companies that have introduced successfully own software products in the global markets.

A recent research has detected that a limited number of Bulgarian software companies have been successful in launching their own products and services in the global markets and has emphasised that the software industry in Central and Eastern Europe (CEE) is entering a new phase, which has not been captured in previous studies (Rousseva 2008a). This is the first documented case in the literature of a small group of latecomer software companies that have successfully managed to develop own software products and enter the global markets.

The main issue arising out of the above analysis is what kind of capabilities the latecomer companies have managed to muster and to what extent they have developed them, and how these relate to the 'ideal' model of capability development. The 'ideal' model of capability accumulation, which the latecomers need to reach in order to achieve the highest level of technological sophistication, requires that all capabilities are developed to a high extent (fig 1 in the Appendix). This ideal model can be used as a template for assessing the level of capability accumulation in companies. It is likely that companies will be able to develop easier basic capabilities (signifying low level of capability

development) and gradually will start to deepen and broaden their capabilities, to reach a state of medium capability development and eventually reach the high (ideal) capability accumulation (fig 1 in the Appendix). The different paths of software development activities discussed above may well lead to different patterns of capability accumulation both in terms of the range of capabilities which the latecomer companies develop and in terms of the depth of capability accumulation. Therefore, studies need to scrutinise the diverse paths of development of latecomer software industries and investigate differences in the patterns of capability accumulation and underlying learning trajectories.

5. Conclusions and directions for further research

This paper contributes to the existing body of literature by developing a 'capabilities yardstick' for assessing the technological capabilities in latecomer software companies. The proposed approach disentangles the specifics in analysing technological capability in latecomer software companies into an array of technical and organisational capabilities which the latecomer software companies need to muster to reach a high level of technological sophistication. This framework aims to capture as adequately as possible the capabilities, which the latecomer software companies need to muster to develop software technological capabilities taking into account both the specifics of the software industry and technological development in the latecomer context. The aim of the paper is to lay the foundations, outline the issues, draw the major capabilities and set the perspective in which to investigate them, and to provide a ground for empirical analyses, which may bring some additional issues.

The main propositions advanced by this paper highlight that development of indigenous software industries is a daunting task for the latecomers, as developing technological capability requires development of a number of technical and organisational capabilities. This amplifies the need to scrutinise the capabilities, which the latecomer software companies have been able to develop in

executing their present software development activities and to analyse whether the latecomer software companies are managing to deepen their expertise and develop a wide array of capabilities, which would allow them to move upwards the 'software development technological ladder' and eventually launch their own products and services in the international markets at some point.

By disentangling the complexity in development of latecomer software industries and the wide variety of technical and organisational capabilities required for accumulation of technological capability, the paper improves our understanding about the complexity in development of latecomer software industries and suggests that some of the existing academic and policy views need further elaboration if they are to be practically useful in supporting catching up activities. While discussing the possibilities for development of latecomer indigenous software industries, the studies need to take into account the wide array of technical and organisational capabilities, which the latecomer companies need to build, and the complexity of technological development in the latecomer context, and to assess the achievements and challenges in the capabilities building process. Public policies aiming to support development of latecomer indigenous software industries need to go beyond building infrastructures (telecommunication, technological infrastructure, education, and so forth), and should focus on capabilities building in the latecomer companies. They should facilitate, empower, or stimulate the accumulation of technical capabilities but also (more so) organisational capabilities, as the latter are critical for harnessing the potential in the company and at the same time may be difficult to develop in the latecomer context.

The proposed approach opens avenues for investigation of capabilities and for comparison between different levels and patterns in their accumulation. One direction of research is to analyse the accumulated technological capabilities in latecomer software industries, which follow different paths of development, as outlined in the previous section. For example, it would be appealing to explore the accumulation of capabilities in a latecomer software industry, which is actively involved in outsourcing

(like India, for example), and to compare these with the accumulation of capabilities in a latecomer software industry, which attempts to develop own software products and services (like Brazil, for example), and to investigate whether different patterns of accumulation of capabilities emerge. Another direction of research is to compare successful and less successful software companies in order to outline the capabilities that appear critical and those that appear difficult to develop in the process of capabilities accumulation. Another important direction of research has to be included in the framework and it concerns the learning activities in the companies. It is important for studies to scrutinise the learning activities contributing to accumulation of different technological capabilities in the companies.

A last point related to this framework of technological capabilities in latecomer software companies concerns its variation and deviation from a framework for analysing innovation capabilities in advanced software companies. The proposed framework has been developed to reflect the specifics in development of latecomer software industries and thus it incorporates and emphasises features that are specific to the latecomer software industries. It disentangles the capabilities by deliberately reaching a high level of disaggregation in order to reveal a wide array of constituencies involved. It also explicitly focuses on basic capabilities, as the latecomers may face difficulties to develop even those.

The approach scrutinises the technical capabilities and classifies them in three major groups starting from basic technical capabilities to reveal the sequential nature and the existence of thresholds in capability development. It is important to underline that the latecomer software companies may face difficulties in building even basic technical capabilities. The capability development task becomes even more daunting with respect to the advanced technical capabilities. The proposed approach has explored also organisational capabilities, as these are responsible for introducing the dynamics of technological learning and are thus critical for development of technological capability. In analysing the organisational capabilities the study has reached a high level of disaggregation and has focussed even

on general organisational capabilities alongside critical organisational capabilities underlying technological capability building, as all these may be difficult for latecomer companies to develop given their limited business experience.

Closely related to this is another important point concerning the relation of this framework to the Capability Maturity Model Integration (CMMI), which is the dominant and universal model for assessing capabilities for software development in companies. The CMMI serves as a template for assessing the level of capability accumulation in all software companies. CMMI lists the whole range of activities associated with software development and portrays the different levels of technological sophistication they involve. Thus, it provides a fruitful base for a detailed examination of the range of software development activities, which a software company undertakes and their technological sophistication. These arrays of activities are related to a particular capability for software development, be it technical or organisational. In this sense, the CMMI is a powerful tool for assessing systematically the array of activities underlying software development, the respective technological sophistication associated with the variety of activities, and the underlying technical and organisational capabilities.

CMMI depicts only the capabilities that are immediately responsible for software development. However, developing mastery over the capabilities for software production and their effective deployment is directly shaped by some other capabilities that establish and shape the dynamics of execution of the capabilities for software production and therefore are important to consider in analysing latecomer software development activities.

Two major types of capabilities fall in this latter category. The first are technical capabilities that are not directly involved in software production but rather underpin them and reflect some business aspects, like capabilities to develop specialised expertise in a particular domain, capability to diversify products and services offered, capabilities to operate in various environment, capabilities for network applications, capabilities to monitor technological development and identify potential niches, etc.

The second type of capabilities that fall in this category are organisational capabilities. The organisational capabilities can exercise a significant impact on execution of capabilities for software development in the company and hence it is important to take them into account. There are different types of organisational capabilities that can exercise an impact on successful development of software activities in the latecomers. One group consists of organisational capabilities that directly underpin the specific business activity in the firm in this case the software production and these are for example capabilities for project management. A second group includes more general organisational capabilities like capabilities for effective management (e.g. communication, coordination, control, etc.). Another group includes 'higher' level organisational capabilities like vision and leadership, strategic thinking, marketing, etc. In the case of latecomers it is also particularly important to consider one additional group that includes capabilities that reflect the specifics of technological learning and technological development in the latecomers, like capabilities to establish organisational culture facilitating learning, capabilities for expeditious learning, capabilities for integrative learning, linkage capabilities, and also vision and leadership, strategic thinking, etc.

Further, if the analyses are to investigate the complexity in development of software development activities in the latecomers in particular, they not only need to register the level of capability accumulation but also need to scrutinise the complexity in technological learning and development of technological capability. The latter requires further analysis to outline the specifics in analysing technological capabilities in latecomer software companies, the interfaces between the accumulation of organisational and technological capabilities, different paths pursued by the latecomer software companies in developing software activities, emerging patterns of capability accumulation, and the implications for capability accumulation and technological capability building. All these issues are taken into account in this research and in this sense it can be seen as complementary to the CMMI in assessing the capability accumulation in the latecomer software companies.

This research emphasises the critical importance of technological capabilities for shaping the development of latecomer software industries. However, it is also important to recognise the impact of other factors, like entry barriers for example, which affect directly the possibilities of latecomers to develop indigenous software industries. Alongside development of technological capabilities, some context-specific factors might exercise an impact in development of latecomer software industries and must be taken into account. For example, studies have recognised that some situational factors, like proficiency in English language and also the time zone have contributed significantly to the successful development of the India software industry. The role of situational factors should be definitely recognised in the analyses. This research explores the specifics and complexity of technological development in latecomer context and acknowledges the impact of the context-specific factors. Nevertheless, this study underlines the critical importance of the capability base in the country and the company-level efforts to upgrade. To illustrate this point one can compare the development of the software industry in Pakistan, a country which has similar advantages to India in terms of time zone and language proficiency but has reaped far moderate success in development of a software industry. The Indian success has been largely attributed to the availability of skilful human capital base, as all of the studies have outlined (Arora et al. 2001; Athreye 2005; Desai 2005; Tschang 2001) and the development of technical and organisational capabilities over time, as Athreye (2005) has revealed. The question, which remains open for studies to explore, is whether and to what extent the Indian companies will be able to make a shift to expand and deepen their capabilities in order to be able to develop own products and services and successfully launch them in the international markets on their own.

Appendix

Table. 1. Classification of software development activities according to their technological sophistication

Technological sophistication of software development activities
Lower technological sophistication
coding (writing new code based on existing designs or algorithms, modifying existing code, etc.)
testing
redesign (redesigning an existing software program)
Higher technological sophistication
initial software design (analysis of customer needs, requirements analysis, architecture, system analysis)
market and feasibility analysis (including analysis of competitor capabilities)
minor software process change (alterations in the work flow or division of labour involved in creating software)

Table 2. Capabilities underlying development of technological capability in latecomer software companies

Technical capabilities
Basic
Capabilities for software programming
Capabilities for testing and high quality assurance
Intermediate
Capabilities for project management
Capabilities for software process management
Capabilities in various operating environments (optional)
Capabilities for network applications (optional)
Advanced
Capabilities for software design
Capabilities to develop specialised expertise in a particular domain
Capabilities to diversify the products and services offered (optional)
Capabilities to monitor technological development and identify niches for potential developments
Organisational capabilities underpinning the accumulation of technological capability
Establishment of organisational culture facilitating learning
Capabilities for effective management (e.g. communication, coordination, control, project management, prompt delivery, change management, human resource development)
Capabilities for expeditious and integrative learning
Linkage capabilities
Marketing capabilities
Capabilities for strategic thinking

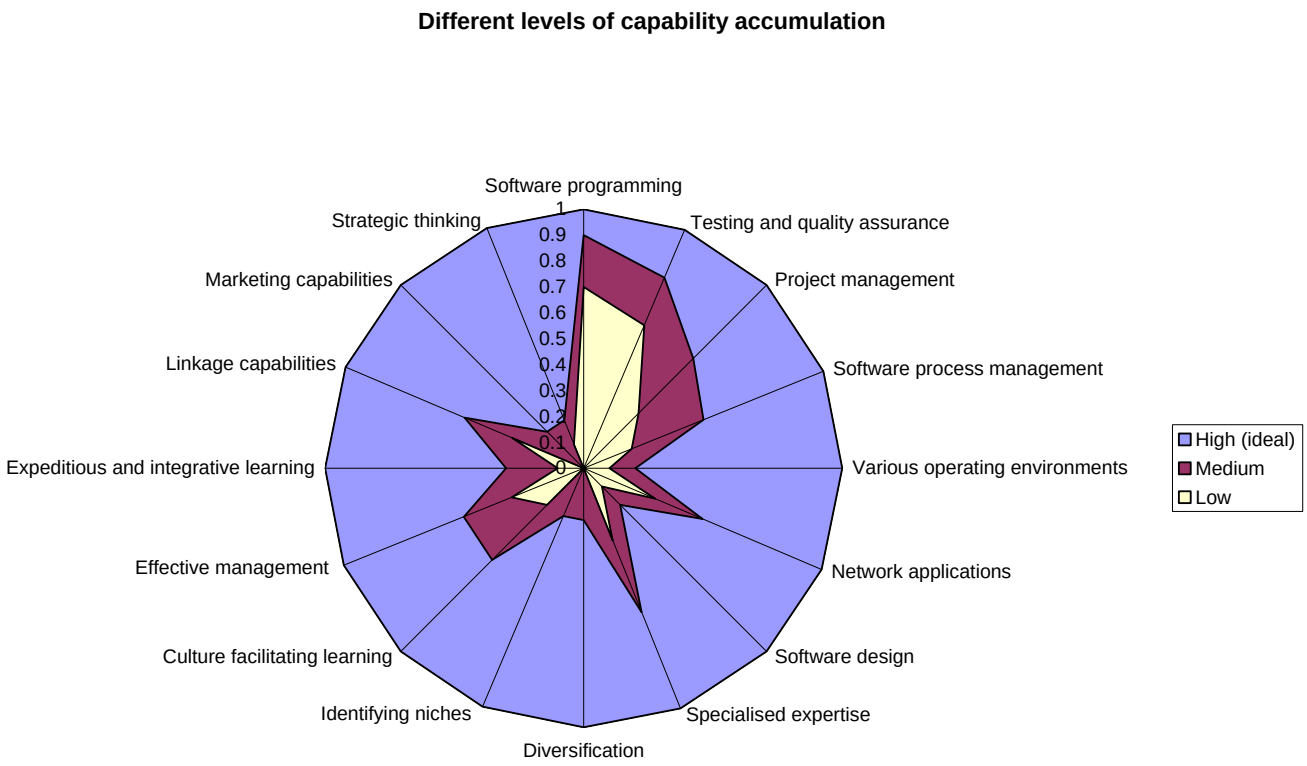

Fig. 1 Different levels of capability accumulation in latecomer companies

Acknowledgements:
This research comes as a result of a research project 'TechCapaBuild' supported by Marie Curie Individual Fellowship within the 6th Framework Programme of the EU. I am grateful to Edward Steinmueller for his valuable comments and substantial help with this research. I am also grateful to Kaushalesh Lal for his comments on an earlier version of this paper. Earlier drafts of this paper have been presented at 4th GLOBELICS Conference, 4-8 October, 2006, Kerala, India and 5th International EMAEE Conference, 14-17 May, 2007, Manchester Business School, UK. I am solely responsible for omissions and the usual disclaimer applies.

Reference:

Arora, A., et al. (2001). The Indian software services industry, *Research Policy* **30**(8): 1267-1287.

Arora, A. and A. e. Gambardella (2005). *From Underdogs to Tigers. The Rise and Growth of the Software Industry in Brazil, China, India, Ireland and Israel*, Oxford, Oxford University Press.

Arora, A. a. G., A. (2005). Bridging the Gap: Conclusions. *From Underdogs to Tigers. The Rise and Growth of the Software Industry in Brazil, China, India, Ireland and Israel*. A. a. G. Arora, A. (eds.). Oxford, Oxford University Press**:** 275-302.

Athreye, S. (2005). The Indian software industry. *From Underdogs to Tigers. The Rise and Growth of the Software Industry in Brazil, China, India, Ireland and Israel*. A. Arora and A. e. Gambardella. Oxford, Oxford University Press**:** 7-40.

Behrens, A. (2005). Brazil. *The Software Industry in Emerging Markets*. S. e. Commander. Cheltenham, Edward Elgar**:** 189-220.

Bell, M. and K. Pavitt (1993). Technological accumulation and industrial growth: contrasts between developed and developing countries, *Industrial and Corporate Change* **vol 2**(No. 2): 157-210.

Botelho, A. J., et al. (2005). The Brazilian software industry. *From Underdogs to Tigers. The Rise and Growth of the Software Industry in Brazil, China, India, Ireland and Israel*. A. Arora and A. e. Gambardella. Oxford, Oxford University Press**:** 99-130.

Carmel, E. (2003). Taxonomy of new software exporting nations, *Electronic Journal on Information Systems in Developing Countries* **13**(2): 1-6.

Commander, S. e. (2005). *The Software Industry in Emerging Markets*, Cheltenham, Edward Elgar.

Contractor, F. a. K., S. (2004). The role of export-driven entrepreneurship in economic development: A comparison of software exports from India, China and Taiwan, *Technological Change and Forecasting* **71**(8): 799-822.

Correa, C. (1996). Strategies for software exports from developing countries, *World Development* **24**(1): 171-182.

Desai, A. (2005). India. *The Software Industry in Emerging Markets*. S. e. Commander. Cheltenham, Edward Elgar**:** 32-72.

Dutrenit, G. (2000). *Learning and Knowledge Management in the Firm. From Knowledge Accumulation to Strategic Capabilities*, Cheltenham, Edward Elgar.

Ernst, D., et al. (1998). *Technological Capabilities and Export Success in Asia*, London, Routledge.

Ernst, D. and L. Kim (2002). Global production networks, knowldge diffusion and local capability formation, *Research Policy* **31**(8-9): 1417-1429.

Figueiredo, P. (2001). *Technological Learning and Competitive Performance*, Cheltenham, Edward Elgar.

Gee, S. and W.-j. Kuo (1998). Export Success and technological capability. Textiles and electronics in Taiwan Province of China. *Technological Capabilities and Export Success in Asia*. D. Ernst, Ganiatsos, T. and Mytelka, L. London, Routledge**:** 46-86.

Grundey, M. and R. Heeks (1998). Romania's hardware and software industry: building IT policy and capabilities in a transition economy. *Development Informatics Working Paper Series*. University of Manchester.

Heeks, R. and B. Nicholson (2002). Software export success factors strategies in developing and transition economies. *Development Informatics Working Paper Series*. Manchester.

Hobday, M. (1995). *Innovation in East Asia: The Challenge to Japan*, Cheltenham, Edward Elgar.

Hobday, M. (2000). *East versus Southeast Innovation Systems: Comparing OEM- and TNC-led Growth in Electronics*, Cambridge, Cambridge University Press.

Kim, L. (1997). *Imitation to Innovation: The Dynamics of Korea's Technological Learning*, Boston, Harvard Business School Press.

Kim, L. and R. e. Nelson (2000). *Technology, Learning and Innovation. Experiences of Newly Industrialized Economies*, Cambridge, Cambridge University Press.

Kuznets, S. (1957). *Modern Economic Growth: Rate, Structure and Spread*, New Haven, Yale University Press.

Lall, S. (1987). *Learning to Industrialize. The Acquisition of Technological Capability by India*, London, Macmillan Press.

Leonard, D. (1995). *Wellsprings of Knowledge. Building and Sustaining The Sources of Innovation.*, Boston, Ma., Harvard Business School Press.

Marcelle, G. (2004). *Technological Learning. A Strategic Imperative for Firms in the Developing World*, Cheltenham, Edward Elgar.

Mytelka, L. and D. Ernst (1998). Catching up, keeping up and getting ahead. The Korean model under pressure. *Technological Capabilities and Export Success in Asia*. D. Ernst, Ganiatsos, T. and Mytelka, L. London, Routledge**:** 325-336.

OECD (2004). *Information Technology Outlook*, Paris, OECD.

Pavitt, K. (1984). Patterns of technical change - evidence, theory and policy implications, *Research Policy* **13**(6): 343-366.

Rousseva, R. (2008a). Identifying Technological Capabilities with Different Degree of Coherence: the challenge to achieve high technological sophistication in latecomer software companies (based on the Bulgarian case) *Technological Forecasting and Social Change, forthcoming*

Rousseva, R. (2008b). Revisiting the framework for learning and development of technological capability. *UNU MERIT Working Paper, fothcoming*.

Saxenian, A. and X. Quan (2005). China. *The Software Industry in Emerging Markets*. S. e. Commander. Cheltenham, Edward Elgar**:** 73-132.

Schware, R. (1989). The world software industry and software engineering. opportunities and constraints for newly industrialized economies. *World Bank Technical Paper*.

Schware, R. (1992). Software industry entry strategies for developing countries: a walking on two legs proposition, *World Development* **20**(2): 143-156.

Soete, L. (1985). International diffusion of technology, industrial development and technological leapfrogging, *World Development* **13**(3): 409-422.

Steinmueller, E. (2001). ICTs and the possibilities for leapfrogging by developing countries, *International Labour Review* **140**(2): 193-210.

Steinmueller, E. (2004). The European software sectoral system of innovation. *Sectoral Systems of Innovation: Concepts, Issues and Analyses of Six Major Sectors in Europe*. F. e. Malerba. Cambridge, Cambridge University Press**:** 193-242.

Torrisi, S. (1998). *Industrial Organisation and Innovation. An International Study of the Software Industry*, Cheltenham, Edward Elgar.

Tschang, T. (2001). The basic characteristics of skills and organisational capabilities in the Indian software industry. *ADB Institute Working Paper Series*.

Tschang, T. (2003). China's software industry and its implications for India. *OECD Working Paper Series*.

Tschang, T. and L. Xue (2005). The Chinese software industry. *From Underdogs to Tigers. The Rise and Growth of the Software Industry in Brazil, China, India, Ireland and Israel*. A. Arora and A. e. Gambardella. Oxford, Oxford University Press**:** 131-170.

UNIDO (1988). The Software Industry: Developing Countries and the World Market. Vienna.

von Hippel, E. (1988). *The Sources of Innovation*, Oxford, Oxford University Press.

The UNU-MERIT WORKING Paper Series

2007-01 *Developing science, technology and innovation indicators: what we can learn from the past* by Christopher Freeman & Luc Soete

2007-02 *The impact of innovation activities on productivity and firm growth: evidence from Brazil* by Micheline Goedhuys

2007-03 *Estimations of US debt dynamics: Growth cum debt and the savings glut in Kouri's model* by Thomas Ziesemer

2007-04 *States and Firms on the Periphery: The Challenges of a Globalizing World* by Gabriel R.G. Benito & Rajneesh Narula

2007-05 *How Do Consumers Make Choices? A Summary of Evidence from Marketing and Psychology* by Zakaria Babutsidze

2007-06 *Inter-firm Technology Transfer: Partnership-embedded Licensing or Standard Licensing Agreements?* by John Hagedoorn, Stefanie Lorenz-Orlean & Hans Kranenburg

2007-07 *The Behavior of the Maximum Likelihood Estimator of Dynamic Panel Data Sample Selection Models* by Wladimir Raymond, Pierre Mohnen, Franz Palm & Sybrand Schim van der Loeff

2007-08 *Location and R&D alliances in the European ICT industry* by Rajneesh Narula & Grazia D. Santangelo

2007-09 *How do social capital and government support affect innovation and growth? Evidence from the EU regional support programmes* by Semih Akcomak & Bas ter Weel

2007-10 *The Micro-Dynamics of Catch Up in Indonesian Paper Manufacturing: An International Comparison of Plant-Level Performance* by Michiel van Dijk & Adam Szirmai

2007-11 *Financial Constraint and R&D Investment: Evidence from CIS* by Amaresh K Tiwari, Pierre Mohnen, Franz C. Palm & Sybrand Schim van der Loeff

2007-12 *The Spatial Hierarchy of Technological Change and Economic Development in Europe* by Bart Verspagen

2007-13 *The origins and implications of using innovation systems perspectives in the design and implementation of agricultural research projects: Some personal observations* by Andy Hall

2007-14 *Technology supply chain or innovation capacity?: Contrasting experiences of promoting mall scale irrigation technology in South Asia* by Andy Hall, Norman Clark and Guru Naik

2007-15 *Are firms that received R&D subsidies more innovative?* by Charles Bérubé & Pierre Mohnen

2007-16 *Foreign direct investment and firm level productivity. A panel data analysis* by Geoffrey Gachino

2007-17 *Technological spillovers from multinational presence towards a conceptual framework* by Geoffrey Gachino

2007-18 *Technological capability building through networking strategies within high tech industries* by Wim Vanhaverbeke, Bonnie Beerkens and Geert Duysters

2007-19 *External technology sourcing: the effect of uncertainty on governance mode choice* by Vareska van de Vrande, Wim Vanhaverbeke & Geert Duysters

2007-20 *Exploration and Exploitation in Technology-based Alliance Networks* by Wim Vanhaverbeke, Victor Gilsing, Bonnie Beerkens, Geert Duysters

2007-21 *ICT Externalities: Evidence from cross country data* by Huub Meijers

2007-22 *Knowledge Flows, Patent Citations and the Impact of Science on Technology* by Önder Nomaler & Bart Verspagen

2007-23 *R&D offshoring and technology learning in emerging economies: Firm-level evidence from the ICT industry* by Zhe Qu, Can Huang, Mingqian Zhang & Yanyun Zhao

2007-24 *The Environmental Porter Hypothesis: Theory, Evidence and a Model of Timing of Adoption* by Ben Kriechel & Thomas Ziesemer

2007-25 *Measuring the Effectiveness of R&D tax credits in the Netherlands* by Boris Lokshin & Pierre Mohnen

2007-26 *The productivity effects of internal and external R&D: Evidence from adynamic panel data model* by Boris Lokshin, René Belderbos & Martin Carree

2007-27 *National System of Innovations and the role of demand. A cross country comparison* by M. Abraham Garcia-Torres

2007-28 *The Global Challenges of the Knowledge Economy: China and the EU* by Can Huang and Luc Soete

2007-29 *Redefining Foreign Direct Investment Policy: A Two Dimensional Framework* by Sergey Filippov & Ionara Costa

2007-30 *Redefine the Nexus between Foreign Direct Investment, Industrial and Innovation Policies* by Ionara Costa & Sergey Filippov

2007-31 *Innovation and Competitive Capacity in Bangladesh's Pharmaceutical Sector* by Padmashree Gehl Sampath

2007-32 *R&D collaboration networks in the European Framework Programmes: Data processing, network construction and selected results* by Thomas Roediger-Schluga & Michael J. Barber

2007-33 *Determinants of alliance portfolio complexity and its effect on innovative performance of companies* by Geert Duysters and Boris Lokshin

2007-34 *Strategic Partnering with Chinese companies: Hidden motives and treasures* by Geert Duysters, Tina Saebi & Dong Qinqin

2007-35 *Approach for analysing capabilities in latecomer software companies* by Rossitza Rousseva