

## Software Product Development in India: Lessons from Six Cases\*

**Rishiksha T. Krishnan**

**Ganesh N. Prabhu**

*Indian Institute of Management Bangalore, Bannerghatta Road, Bangalore 560076 India  
rishi@iimb.ernet.in; gprabhu@iimb.ernet.in*

*We draw lessons for policy and practice from six in-depth case studies of software product development projects in India. The six projects were chosen to provide a diversity of industrial contexts, types of firms, types of products, types of markets, and levels of customisation. We provide the salient features of each project and a cross-case analysis of the major issues, strategies adopted and trade-offs made in them. We draw implications for policy makers and managers of software product development projects in India and other developing countries.*

Rishiksha T. Krishnan is Associate Professor of Corporate Strategy and Policy at the Indian Institute of Management at Bangalore. He obtained his doctoral qualification from the Indian Institute of Management at Ahmedabad where his thesis proposal won the IFCI award. His research interests are in strategic management, innovation and new product development. He has been involved with the founding of two start-up firms, besides consulting for small and large companies and the Indian government. His recent publications have appeared in *International Journal of Technology Management and Science Technology & Society*.

Ganesh N. Prabhu is Associate Professor of Corporate Strategy and Policy at the Indian Institute of Management at Bangalore. He obtained his doctoral qualification from the Indian Institute of Management at Ahmedabad. His research interests are in new product development and technology management. His industry experience includes managing a turnaround and consulting for a variety of firms. He is a recipient of the Best Young Faculty Award from the Association of Indian Management Schools. His recent publications are in *Research Policy* and *Academy of Management Review*.

---

\* We thank the editors E. Sridharan and A. D'Costa for useful comments on earlier drafts of this paper. We thank the six companies studied for providing us access and their product development team members for spending considerable time to patiently articulate their organisational and partnership experiences. However all conclusions reached and inferences drawn are those of the authors and do not necessarily represent the views of the companies studied.

## 1. Introduction

While the software industry is considered a highly successful economic growth engine in India (NASSCOM 2002), its rapid growth in recent years has been achieved by firms primarily providing manpower-intensive customised software development and maintenance services to foreign clients (Arora et.al. 2001). While this business model has enabled Indian software companies to transit smoothly from software "body-shopping" services to off-shore software development in India, it has also made them vulnerable to the business cycles in client countries. The software services model is manpower intensive and growth is achieved only by proportionate increase in number of software engineers employed. This results in large organisations that become increasingly difficult to co-ordinate and control. The competitive advantages of this business model are narrow (Arora, Gambardella and Torrisi 2001) and may be eroded over time by lower cost countries like China.

One alternative to the software services model is a business model centred on software products. While the distance from India of the largest market for software products (the United States) makes a continuous assessment of user needs in that market difficult, there are other options such as addressing the needs of the Indian market or creating "core products" (Prahalad and Hamel 1990) that could form the basis for software solutions in Indian and foreign markets. The availability of products suited to Indian user needs might induce Indian manufacturing companies to overcome their tendency to use "custom development for standard functionality"<sup>1</sup> (Accenture 2002: 31) and thereby enhance their ability to leverage information technology. From the perspective of national competitiveness, it could be beneficial to have more diversified software firms whose clients use "packaged solutions" that incorporate best practices for standardised functions at lower costs, while utilising customised software services primarily for unique applications (Accenture 2002: 54).

Indian software companies may be able to create software solutions for clients more easily and quickly if they build these solutions around products they have developed. Packaged products could also enable firms to benefit from the multiplier effect - by selling multiple copies of the same software at low marginal costs. In spite of these potential benefits, few Indian firms have made a serious attempt to create a portfolio of software products, nor has there been a growth of independent software product companies. The Indian software industry association NASSCOM has listed "low presence in global packaged software market" and "lack of original technology development orientation" as the two major weakness of their industry (NASSCOM 2002: 32).

This study attempts to understand the issues that impact the creation and management of software products in the developing country context of India. We draw lessons for policy and practice, from the collective experience of software product development in six diverse software product development projects in India. We draw implications from the study for software firms in general, the software industry, policy makers in

India and, by extension, to other developing countries.

## **2. Research on Product Development**

Before exploring some of the specific issues regarding software products and the influence of the geographical context, it is useful to summarise the findings of the extant generic research on the management of product development:

The key objectives in product development are to minimise time-to-market, and to maximise the fit between customer requirements and product characteristics (Schilling and Hill 1998). Achieving these objectives helps a company recoup its development costs rapidly and make reasonable economic returns in highly uncertain environments. Companies seek to get to market early, develop products with external and internal 'integrity' (Clark and Fujimoto 1990), make optimum use of development and other resources, and develop not one, but a stream of new products over time.

Companies are advised to pay adequate attention to the 'fuzzy' front end<sup>2</sup> of the product development process (Khurana and Rosenthal 1997), listen to the voice of the customer (Griffin and Hauser 1993), improve intra-organisational communication, develop a common language to communicate and share product development ideas, and to deploy cross-functional teams in a structured product development process (Ulrich and Eppinger 1995).

A clearly articulated strategic intent (Hamel and Prahalad 1989), a development strategy that chooses the right set of projects and helps integrate strategic planning with R&D strategy (Wheelwright and Clark 1992), strong top management involvement at early stages of the project, empowerment of project leaders as 'heavyweight project managers' (Clark and Fujimoto 1990), transferring and leveraging skills and competencies across the company (Prahalad and Hamel 1990), and setting targets for revenues from new products are considered best practices. Top management is urged to create an organisational climate in which honest failures are tolerated, creativity is rewarded and inter-functional and inter-divisional barriers are lowered (Kanter, Kao and Wiersema 1997).

Some of the attributes of software products and the context in which product development is undertaken, influence the complexity and challenges involved in software product development in India. These issues are discussed in the following sections.

### **2.1 Software Products and Software Product Development**

Computer software is the stored, machine-readable code that instructs a computer to carry out specific tasks. The three types of software are: (a) *tools* used to generate applications to retrieve, organise, manage, and manipulate data, (b) *applications* designed to solve specific problems inherent across all industries or in a particular industry or business function and (c) *system-level software* that control the internal

operations of a computer (Mowery 1999). All three types of software can be provided in either 'standard' or 'customised' form. The term software product generally refers to a traded, standard software program though some of the more complex forms of software products involve some degree of customisation as well. In this research we focus exclusively on application software projects.

Software products differ from physical products in a few respects. While physical products need to have production issues explicitly addressed during the product development process, and late changes in product designs can have cost and time implications, software products do not have such constraints. Instead, other issues like compatibility with hardware platforms and operating systems - both of which change rapidly - become critical. Software products are easier to replicate, either through piracy or by copying key features and providing adequate protection of intellectual property rights is difficult. Network economies play an important role in software products - having a community of users and an installed base of systems software is often essential, as it facilitates interaction and file exchange among users. While architecture and compatibility issues are important, specification of features can often be delayed to near launch date to incorporate the latest needs of the target customers.

Generic packaged software applications often need large volumes to break even. Software duplication costs are low compared to that of development, maintenance and marketing. There are high costs associated with maintaining and servicing a product over its lifetime. Users expect frequent upgrades and demand the option to shift to an upgrade at a small incremental price. Upgrades have to be compatible with earlier versions and need to retain features that users have become accustomed to. Once the user gets used to a particular software package, there may be a 'lock-in' effect, as switching to another software requires learning. Launch and promotion costs are often high as companies stage public relations events, advertise value propositions, and enforce product cannibalisation to market newer versions. Marketing software products in international markets may need localisation of the product itself as well as the marketing mix.

The US market is the largest market for packaged software, accounting for nearly 50% of the \$200 billion global market for packaged software (OECD 2002:320). Since 'user-friendliness' and adaptation to local operating conditions are critical in the case of application software, proximity to users is essential in developing appropriate products and being effective in the product development process (Mowery 1999: 163). Therefore US based software firms that develop application software products hold an advantage over other firms that have no presence in the US (Mowery 1999: 160). The offshore structure of software production which is well suited to customised software development, is not suited to the short development cycles, rapid prototyping, and high responsiveness to user needs, that are essential for packaged software development.

## **2.2 Product Development in the Indian Software Industry**

Indian software companies have largely stayed away from product development. Even industry majors that set ambitious targets for product revenues have given up their targets over time. For example, the major firm Infosys Technologies, had about 4% revenues from software products in 2001-02 (SEC 2002), though at one time they hoped to achieve a target of 40% revenues from products by 2000 (Prahalad 1996)<sup>3</sup>.

There are some reasons for Indian software firms to stay away from developing products. India is well established as a source of software services and there are large firms that are successful role models for new firms and smaller firms to emulate. This makes services the dominant logic among Indian firms. Software services enable Indian companies to be highly profitable with relatively low risks and ensure regular immediate cash flows, in contrast to product development that involves large initial investments and future, uncertain cash flows. The major software services firms have high valuations and generate high expectations from investors and analysts who expect these firms to "de-risk" their ventures. The software industry is characterised by low physical capital intensity and high human capital intensity (Mowery 1999: 156). The steady supply of qualified software professionals who can generate revenue immediately through services, have probably added to the inertia of success among software service firms as their business model is not seriously threatened.

The software services business model is based on limited client-specific relationship marketing to generate business, and efficient project management and quality management to execute it. Indian software firms have been content to leave the conceptualisation and design of a software solution to a client-appointed information technology consultant and undertake only the coding and implementation of the software. Such software projects are usually well defined and have clear deliverables. In contrast, software product development is viable only if the firm can internally conceptualise unique applications and capture value largely through the marketing part of the value chain. The limited demand potential of the Indian domestic market and the large geographical distance from the biggest market for packaged software, the United States, makes the entry and marketing of software products a difficult proposition. Software product firms need to have appropriate reference clients to convince other clients to buy expensive software products and Indian client installations are often not considered appropriate reference installations. Further, maintenance over the life cycle of a product in foreign markets is expensive. So is providing support in multiple locations worldwide, though limited support can be provided over the internet. There are problems with software piracy and the lack of adequate venture capital that make software product development in India difficult.

Product development can enable software firms to be at the leading edge of technology by exploiting synergies between their service and product businesses. But there are evidently difficulties in managing both types of projects. Nambisan (2001) found in a survey of 137 firms (with a predominantly Indian sample) that 87% of product ventures initiated by software service firms were unsuccessful, primarily due to inappropriate transfer of organisational practices and development culture. Some Indian software firms may also face an additional barrier due to their being locked-in

at a lower level of innovation, due to their prolonged involvement with low-end service projects (D'Costa 2002).

However, software product development has the potential to give high returns to Indian software companies (Krishnan and Prabhu 1999). Among software professionals, there is often a higher level of prestige associated with working for software product firms compared to software service providers. Software product development projects could therefore potentially attract high quality Indian software development talent that currently migrates to US software product firms. While the lack of proximity to the large US market may be a hindrance, many Indian firms have a strong US presence for providing software services that can potentially be extended for developing software products. There are product application areas in developing countries that are not served by products from developed countries. There is potential for Indian firms to develop niche products at low costs specifically for the Indian market and then extend these products to other developing country markets to generate volumes. For example Soft Systems, India has developed a low-cost specialised enterprise resource planning software for managing plantations in India, and now has 560 installations across 12 countries (Jishnu 2002).

### **2.3 Product Development and Innovation in India**

It is intriguing that there have not been more visible efforts at software product development, particularly when the software industry is not affected by the factors that hold other Indian industries. There is a pattern of limited product innovation that characterises the Indian industry across all sectors of the economy. Much of this is due to the legacy of a protected economy where innovation was unnecessary and often thwarted by government policy (Forbes 1999, Bowonder and Richardson 2000). The scarcity of investment capital made most entrepreneurs risk-averse. Many engineering products manufactured in India were governed by technology licensing agreements that placed restrictions on their slightest physical modification or even improvement. Import restrictions made it difficult for companies to source particular components or skills and capabilities they lacked. Scarcity of design skills and experience, lack of appropriate engineering resources, lack of a strong market orientation, and centralised control by business family heads have been some of the other constraints to product development in other sectors in India (Krishnan and Prabhu 1999).

None of the above factors are germane to the Indian software industry. Further, in recent years, after the liberalisation of the Indian economy was initiated in 1991, there has been increased interest in product development in Indian companies in diverse industries (Forbes 1999, Krishnan and Prabhu 1999). In an increasingly competitive and crowded marketplace, the ability to develop and launch new products has gained in importance, both to create product differentiation as well as increase primary demand. As the software industry requires low physical capital and high human capital, Indian software firms could be expected to be at the forefront of product innovation. However, this is not evident. India has an abundance of programming and software development skills, yet creates few packaged software products. Software

industry insiders attribute this lack of a package orientation to factors in the larger innovation system outside the software companies. Desai (1998) believes that 'innovation can flourish only in an ecosystem that has the elements of market, money, university, cluster of companies, attitudes, culture and the appropriate regulatory and legal environment' and while Mehta (1998) emphasises that software product development requires 'an R&D culture, market intelligence, skills to develop user-friendly software and documentation, availability of funds and special marketing skills'

### **3. Understanding Recent Product Development in India**

The Indian software industry provides a rich research context to understand the factors and processes that influence software product development in a developing country context. Given the late evolution of product thinking in Indian industry, and the dominant service orientation of the software industry, it is likely that the major challenge in software product development in India is developing a product orientation. This consists of being able to conceptualise a product in terms of what need it meets, what is the target market, and being able to come up with a distinctive product that meets user needs adequately and yet differentiates itself from competitive offerings in ways that customers value. This task is complicated as research suggests that service and product businesses differ in terms of project management, culture and organisational practices (Sawyer 2000, Nambisan 2001).

It is also likely that firms that focus on niche markets in India and then extend the product to other developing country markets over time are more likely to succeed. Also Indian firms that build products around customised software that they have earlier developed may be more likely to succeed. In India, product development speed and efficiency considerations may be secondary to considerations of fit with customer requirements. In the light of these initial propositions, we study a diverse set of software product development efforts of Indian companies to understand the factors that affect the initiation and implementation of software product development projects. We also attempt to understand the motivations and commitment of pure product companies in India that operate in a context where service companies are dominant and successful. We seek to understand whether the developing country context, and the sequence of entry from services to products, creates unique opportunities and difficulties for software companies that undertake software product development projects in India.

Given the relative lack of literature on the variables and factors affecting software product development in India, we chose to use a qualitative exploratory research method, the case method of research, that is considered appropriate for such research contexts (Yin 1994). We conducted in-depth case studies of six software product development projects developed by six software firms in India. We chose one software product in each firm and traced the process of initiation and implementation of the software product development effort within the firm. Through the study of company records, in-depth interviews, secondary material and industry/press reports

we constructed detailed case studies of the product development process for each of these projects.

Our approach has been a multiple case study approach (Yin, 1994). Potential cases were identified from the newspaper and magazines. The projects studied were only of applications software products, and were chosen to provide a variety of industrial contexts, types of firms, types of markets, and levels of customisation. We covered projects from both "pure-product" companies and "service-product" hybrid companies. We covered both commercial successes and failures, products selling in large volumes, niche products, and products that emerged from customised software development projects. By a process of analysis within cases and synthesis across cases, we have drawn key learnings regarding software product development in the Indian context.

We analysed the cases across nine dimensions from four important themes that reflect the issues emphasised in previous sections. First, the product-market choice issues were looked at under the dimensions of product definition and positioning, company size and choice of product, and whether the product choice emerged from technology-push or market-pull. Second, the product design and user interface issues were explored through the interplay of technology choice and product architecture, product architecture and customisation. Third, customisation and version management as well as marketing and product launch issues were investigated as one dimension. Finally, the influence of internal organizational factors such as reward and incentive systems and quality systems on the product development process was examined across cases.

### 3.1 Six Companies, Six Products

Summaries of the products studied and the companies in which they were developed are given in Table 1. We also provide a short description of each firm and its product.

**Table 1: Companies and Products Studied**

<b>The Company</b>	<b>The Product</b>
<b>1. Infosys Technologies Ltd.</b> – India’s most well-known large software firm.	<b>BANCS 2000</b> – a banking product for the domestic and international market.
<b>2. Ramco Systems</b> – One of India’s larger product-focused software firms.	<b>Marshal</b> (later re-named e-applications) – enterprise resource planning product.
<b>3. Tata Consultancy Services</b> – India’s largest software company. Part of the Tata conglomerate and privately held.	<b>EX</b> – financial accounting product positioned as the world's friendliest accounting software.
<b>4. Concept Software</b> – a small software company focussing on multi-language desk top publishing products.	<b>InPage</b> – an Arabic script based multi-language desktop publishing software product.
<b>5. Eastern Software Systems</b> – a small software company focusing on enterprise resource planning products.	<b>MakESS</b> - enterprise resource planning product targeted at small companies.
<b>6. RiteChoice Technologies</b> – a small	<b>Spectrum</b> – a product to manage back -

software company focussing on the stockbroking domain.	office and middle office of a stockbroker.
--	--

The case studies are based on data collected upto June 2000. Subsequent events to the extent known to us have had no impact to our general findings and implications.

#### Infosys Technologies and BANCS 2000:

Infosys Technologies is one of the largest software firms in India and the first Indian software firm to be listed on a US stock exchange - NASDAQ. Its major business is in software services with its revenues from software products being about 2% of their total revenues. It is known for excellence in management and corporate governance.

The development of the BANCS 2000 product owes its inception to a pilot project for bank computerisation initiated by the Indian Banks Association in 1989. Infosys was one of eight firms chosen for participation in this project. Infosys decided to create a 'foundation framework' and an integrated approach to this product, even though this delayed the first deliveries, in the hope of being able to create a more robust and easily expandable and scalable solution for marketing to other banks. The major challenge faced by Infosys has been to meet customer expectations for customisation and for an offering that was differentiated from that delivered to other customers, while at the same time maintaining a single code base (believed by Infosys to be an important aspect of product development). The formal inauguration of the product took place in 1994 though some bank branches were already working with this product earlier.

Despite its small size relative to the service business at Infosys, the banking project had, since its inception, commanded a high level of top management interest as it is seen as the basis for creating a larger product business in future. A major revision of the product was undertaken in 1997 to enable the product to meet the requirements of foreign banks such as multi-currency operations and centralised banking operations. Subsequently, the product has been internet enabled through a new utility called BankAway. The product has been adopted by some Indian private sector banks and has found markets in Southeast Asia and the Middle East. It faces tough competition from at least ten major competing products worldwide. As the product requires relatively high level of customisation, a key factor in influencing prospective sales is the demonstration of installations with high profile customers in foreign countries. Banking products remain the only significant product offering of Infosys. Other products developed by Infosys for retailing and warehousing were spun-off into Yantra Corporation in the US, in which Infosys now has a minority stake.

Subsequent to our study, Infosys completely re-engineered the product and named it Finnacle. Finnacle is now reputed to be among the world's top three banking products.

#### Ramco Systems and Marshal:

Ramco Systems is part of a large South Indian industrial group with major interests in

cement and building materials. The vision of creating a world-class software product company revolving around enterprise applications goes back to the late 1980s when the Vice Chairman and founder of Ramco Systems was a graduate student in the US.

Product development on the Marshal product started in 1989 and the first release was in 1993, based on a robust technical architecture that had been conceived in 1992. A number of releases followed in the Indian market till 1995. A major effort to upgrade the product to international standards followed after 1995 with about 400 software professionals involved, leading to the release of Version 3.0 in 1997. Version 3.0 combined good functionality with strong product architecture and was launched in multiple international markets.

The product has been relatively successful in certain markets and geographies – e.g. human resources applications in British legacy markets (Malaysia, Singapore, India, UK); enterprise asset management in capital-intensive industries such as power, gas, aluminium where maintenance is critical; in process industries like food, chemicals, textiles, as the structure of the software was well-suited to these businesses. The installed base as of May 2000 was 156 customers in 650 locations and 11,000 users in 11 countries. The largest implementation was in the Swatch group (17 locations). The product was internet enabled and re-launched as "e-applications" in 1999.

#### Tata Consultancy Services and EX:

Tata Consultancy Services (TCS) is the largest software firm in India and part of the Tata business conglomerate. Though a majority of its revenue is from software services, TCS has pioneered or developed several software products in India in the areas of banking, accounting, insurance and healthcare. They have developed several software development tools such as MasterCraft, Adex and Assent that offer them an advantage in software product development.

EX is TCS's major mass marketed packaged software. Positioned as the world's friendliest accounting software, the software is equally amenable for single person, small business and large-firm accounting. It was the first mass-marketed Indian software product. It was launched in 1991, with an innovative stage play that featured well-known Indian stage actors, full-page newspaper advertisements, launch events in major cities and information brochures being distributed widely.

The present product E.X. Next Generation 1.5 is a comprehensive business accounting software with document designer and extensive reporting capabilities for medium sized businesses. For larger corporations with higher volume requirements, EX Next Generation 1.5 Multi-User is positioned as a robust, reliable and extendable accounting solution. The product has developed over several versions – both in feature upgradation from small single user to multi-user versions, and in increasing capacity for international versions. TCS runs its own accounts on EX. The product was distributed through 'Keydealers' and direct 'Keymen' commission agents and backed by online user support. TCS collected user feedback through a registration process

that was linked to upgrades. Attractive package design, multiple computer media options and bundled offers supported sales. A 'transparent' hardware lock (a hardware lock that did not block a port) protected the software from piracy. TCS considers this product conceptually strong, user-friendly, feature rich and affordably priced. EX offers off-the-shelf installation and training. Over the years, the product has been backed by a consistent promotion focusing on user friendliness. The product is estimated to have eighty thousand users. It has won the Best Software Product Award from the Computer Society of India as well as the Most Popular Software Award, the Best Packaging Award and the Best Communication Award of the Society.

Since its launch, upgrades of the software for new features were essentially through software layering and add-on coding over the basic software. In 2000, a new and independent team initiated a major restructuring of the EX package. The restructuring was aimed at enhancing quality, features and performance by using an entirely new software code built using internally developed software tools while retaining a major part of the graphic user interface that was familiar to EX's current users.

#### Concept Software and In-Page:

Concept Software is a small software firm based in Delhi promoted by two engineering graduate classmates from the Indian Institute of Technology at Kanpur. The partnership firm started with several small product development projects that had mixed success. The last of these software development projects was to develop Russian language word processing software. While that project was not a commercial success, the partners gained valuable knowledge in the design and development of generic word and character processing software.

The partners realised that the best application area for their knowledge of character processing was in the development of Arabic script based word processing software. Unlike English that requires only a simple placement of letters next to each other, the Arabic scripts use compound lettering that require complex changes in the shapes of basic letters as extensions to a character are added. This means that, while the number of characters is limited (unlike the large number of characters in the Chinese script), a simple set of characters and extensions are insufficient for Arabic scripts. Words need to be built by developing complex shapes within a single complex character as multiple characters are added on a keyboard. Arabic script software that was available before Concept Software's InPage made compromises in the authentic shapes of the script to suit software limitations, that were aesthetically unacceptable in communities that view the Arabic script as an aesthetic form. These software packages also made the script difficult to read. Publishers of the best newspapers and books in Arabic scripts did not use such compromised software and relied on hand-written calligraphy.

Concept Software was the first and only firm to build an Arabic script software - InPage - that used a compounded lettering scripting technology developed internally for a limited set of Arabic based languages. An employee trained in classical Arabic scripts and trained by the partners in computer programming helped the firm develop

a very large library of compounded Arabic letters that appeared and changed on screen as individual alphabets were typed using a conventional keyboard. They also developed a set of readymade and commonly used Arabic mastheads as graphic files within the package. These were popular Arabic verses and quotations that were scripted into very complex aesthetic shapes that could not be created through the keyboard. Both the compounded lettering and readymade graphic files soon made InPage the undisputed best Arabic script software in the world.

Concept Software launched the first version of InPage with a software lock at a fairly high price. However the software lock was soon 'broken' and the package was pirated widely at low prices in several countries. Concept Software subsequently built an improved version and marketed it at a far lower price than its first version, with a hardware lock that has not been broken so far. The product is currently a monopoly and no product of comparative quality and features exists. Almost all Arabic newspapers and publishers in the world use either of the two versions of InPage. Most of the high end users have purchased authentic copies of the new version of InPage. However the pirated low priced copies of the earlier version of InPage represent the major competition for the new version – especially among individual buyers. The market for this product is largely in Arabic countries and marketing is primarily through the firm's agents based in UK, as the company does not have the resources to sell directly in multiple Arabic nations.

#### Eastern Software Systems and MakESS:

Eastern Software Systems (ESS) is a small software firm based in Delhi that has built an enterprise resource planning (ERP) software package, MakESS, targeted at small firms. Competing ERP products from international industry majors such as SAP and BAAN are priced at over US\$50,000 in India, far too expensive for small Indian firms who cannot afford such high investments and do not see commensurate benefits that justify such a price. On the other hand, low-featured, customised ERP products, that are developed by small firms for US\$ 2000 or less in India are quite unreliable.

Seeing the gap develop in the small firm market for ERP software, ESS developed MakESS in three interlinked versions. The entry-level package (human resource management only) retailed for about US\$ 2000 (useful for almost any product or service organisation), a second full version suitable for non-manufacturing firms at US\$10,000 (this included the human resource package) and a third full version for manufacturing firms at US\$16,000. With over a hundred installations in India, some in fairly large multi-location firms, MakESS has occupied a niche as a relatively low cost, value-for-money, ERP package in India.

The firm promotes modification of client business practices to best practices that are incorporated in the software, but unlike larger ERP firms, is relatively more willing to customise the package for specific requirements by charging the client additionally on a person-month basis. By making the package highly user friendly and targeting firms in specific industries that can inherently adapt easily to it, the firm saves on

installation, customisation programming and support costs. For example, pursuing this marketing strategy, the firm has built an innovative add-on special module for milk procurement activities in milk co-operatives (a specialised application not available in other ERP software) and is targeting hundreds of milk co-operative plants throughout India through the National Dairy Development Board.

The firm recently released its internet based ERP version on a 'no initial fee - low monthly charge' basis. The firm sees internet versions as an extension of the market to smaller firms rather than a shift in the core market of larger firms that are more likely to opt for intranet based versions. Major software firms worldwide have reposed faith in ESS as they hold minority equity stakes in the firm.

RiteChoice and Spectrum:

The Spectrum product is the outcome of a decade long effort to provide a product that can manage all the essential operations of a stockbroker. Started as a partnership in 1988-89, its first product was launched in 1989 and reflected 'stockbroking as taught by brokers'. A LAN-based, scaled-up version was launched in 1991. RiteChoice was started in 1992. A new prototype concept was subsequently created in 1995 and upgraded and ported to Oracle as Spectrum 2000E.

The Spectrum product has client-server architecture, and can handle back-office (clearing, settlement, brokerage) and middle office (risk management) functions for multiple exchanges, multiple locations and multiple instruments. It is designed for large brokerages with multiple branches and many sub-brokers. A major feature of product development at RiteChoice has been the intense academic study of the broking process to gain domain knowledge before initiating product development. A concept paper incorporating state-of-the-art best practices in the area formed the basic user knowledge used for product design and coding. Experts vetted the proposed product design before actual product engineering took place. Their knowledge-intensive approach has been acknowledged by the minority investment of IT major Intel in this company.

### **3.2 Explaining Product Development in the Six Companies**

In this section, we summarize and discuss our analysis across the six products and companies that we studied. The analysis is across nine dimensions covering four themes - product-market choice, product design-user interface issues, marketing and product launch, and interface with organizational systems.

#### Product Definition and Positioning

An important aspect of software product development is the clear definition of the target market, the need being met and the distinctiveness or competitive edge of the product being offered (most visible in MakESS, Spectrum, In-page, and later versions of Marshal). Clarity on these issues enables clear definition of the product itself -

which is essential for development of a product that has integrity, and that can be developed in a reasonable time frame without 'creeping functionality'. While it may not be possible to do all this the first time around, especially in a complex product, a successful product developer learns to address these questions rapidly and adequately (Marshal to a large extent, and to a smaller extent BANCS 2000).

Early product definition is more likely to take place rapidly in a 'product company' (Marshal) than a 'services company' (BANCS 2000) that also makes products, as in the latter there is a conflicting culture of a propensity to make incremental changes to meet emerging customer requirements. A product company is more likely to put a stronger hold on functionality for each version.

Developing products for highly regulated industries in emerging markets necessitates the ability to keep track of new notifications and other developments and to continually update and upgrade the product. RiteChoice has found that new transaction types and new instruments allowed by the Securities and Exchange Board of India over time have necessitated frequent changes in the product. On the research side, they have to be continually ready for introduction of these new features at short notice.

#### Company Size and Choice of Product

For small companies, niche products are an effective way of entering the product market (In-page in Arabic publishing and Spectrum in stockbroking). Typically, niche products that address specific, usually narrower or lower order needs of a small group of users, are less complex. They are more customised and can be developed in relatively shorter time frames and typically do not need huge development resources. The absence of a strong network of independent software dealers however makes it difficult for these niche products to reach the relevant and possibly distributed market, as it is prohibitively expensive to set up a separate marketing network for a single product or a small group of products. This problem exists in international markets as well, though the challenge in that case would be to set up a partner network that would enable greater market access and penetration (Concept Software has a sales agent in UK for world sales).

Software product development requires major investments in development time and money. Larger firms can hire a large number of developers and invest for long periods in developing new products that are either supported by revenues from customised software development (BANCS 2000) or implicitly by other businesses of the business group (Marshal). Smaller firms can also invest time and resources to product development projects if supported by larger firms (ESS has venture capital investments from large software firms). Smaller software firms with little external support often have to depend on client firms that are willing to allow the software firm to develop and experiment with new products as long as they get a reasonably customised product for themselves.

The strongest, inimitable, knowledge-based competencies seem to be in the small companies – in the case of In-page, character and image manipulation; in the case of Spectrum, knowledge of stockbroking and related issues. The presence of a strong core group facilitates this process. In the case of In-page software development, the two partners have been working together on several related ventures to build up their knowledge base. They have also internally trained a person with classical Arabic knowledge from a traditional Arabic school, to work on software design – a unique combination that is possibly an inimitable asset for the firm. In the case of Spectrum, about one-third of the total strength of about sixty people have been working on the product for at least five years. The founder, Mr. Rangarajan, has been focusing on the product and the stockbroking sector for over a decade. In comparison, the large companies seem to have complementary competencies in tool development, and building powerful architectures for their products.

### Technology Push or Market Pull?

Software product development can happen by either technology-push or by market-pull. Indian firms seem to have been driven more by the former than by the latter. BANCS 2000 started with a specific requirement of a bank (external market-pull) but later became technology-driven (client-server); In-page was competence driven - the firm wanted to know how to leverage the competence they had already developed in image and character manipulation. Marshal was started with a project to develop an enterprise solution for a particular industry group. The firm also had a desire to become a product player and extended the product to other industries till it became more technology-driven in its conceptualisation of the product and its versions.

Software development depends on developing an iterative focus on technology-push and market-pull, never allowing one or the other to become dominant. Competitive advantages can emerge from either of the two forces as they can lead to unique product features that are valued by customer. Products can become more powerful by identifying and focusing on latent needs. For example, RiteChoice found that brokers do not have adequate information and control over how much they are funding their sub-brokers. Setting limits for each sub-broker by itself is not enough, as there would be an opportunity loss associated with one or more sub-brokers not utilising their allotment fully. They therefore integrated a dynamic mode of risk management of sub-broker accounts into their product. This contrasts with Marshal, which was more dependent on information from consultants and others about customers' needs. In both cases, the product companies were able to identify and develop unique features for their products.

### Choices of Technology and Product Architecture

Technology and architecture choices are important as they have a continuing impact on the evolution of the product. Choice of a robust technology or framework facilitates upgrades and addition of features while at the same time ensuring compatibility with other hardware and software products. These technological choices

are particularly important for complex and high-value products that are likely to be in use by a customer over an extended cycle of up to ten years. (Marshall, BANCS 2000).

While technology is an area of strength for Indian software product developers, there is also a tendency to try and solve problems with a technology solution rather than a marketing solution. For example, if the range of customer requirements are not clearly identified and prioritised in advance, developers may consider all requirements as equally important and build a complex product that addresses all possible needs, instead of building a simpler product that meets the prioritised needs of the target set of customers more effectively. This makes the engineering activity more complex, delays development, and may divert attention from real market problems that have to be solved through hard marketing choice decisions (BANCS 2000). On the other hand firms sometimes try to avoid adding features that were initially not considered important enough to be addressed, but are in increasing demand from the market, as they involve major changes in the basic software architecture and therefore extensive rewriting of code (BANCS 2000, EX). Once the market demand is persistent, the need is met initially by overlaying the basic software with an add-on layer that addresses the need – a compromise solution that increases processing time but reduces feature development time. Eventually such features that emerge as essential over time are built into the main product only when the product is entirely re-engineered (EX) by a fresh software development team that has to often ignore all the software code written earlier as it is too complex to reuse.

### Product Architecture and Customisation

Customisation of software products is relatively easier if the software is architected to be modular and layered. Modular and layered architectures however, compromise on processing speed and other output performance parameters. Therefore customised software developed for single clients are usually built with more integral architecture to provide superior performance. However, once successful at a single client stage, if such software is expanded into a product with the same integral architecture, it poses severe problems for both new feature building and customisation. Software layering over the basic integral architecture becomes the preferred route in such cases thus decreasing performance but allowing customised software to be 'productised'. However with each layering the product becomes more and more difficult to modify thus leading to a situation where eventually a major reengineering becomes essential.

An important decision in the product design and architecture is which variables to hard-code and which ones to parameterise (BANCS 2000). Hard-coding allows for faster software development and higher processing speed and performance. However, it compromises on flexibility for both the customer and the developer. From the customer's point of view, hard-coding makes related policy changes in the application that require changes in the software product settings difficult to implement without reworking the code. A pre-parameterised code allows the customer to make such changes internally and rapidly. However, it also requires the customer to maintain security features so that unauthorised persons do not make parameter changes. More

importantly, it also requires the customer to think carefully into the future and estimate the range of parameterisation that may be required. Some firms that are also involved in customised software development found that their customers were often not willing to do such advanced thinking. They preferred hard-coding to meet present software performance requirements rather than parameterisation that involved higher development time and had uncertain utility in the future. However, from the software product developer's point of view, greater parameterisation is essential to develop a generic product, so that product features can be enhanced as customer requirements and expectations increase. To do so, the developer needs domain expertise and need to forecast possible changes that are likely to occur in that domain. In the relative absence of this understanding, the developers have to depend on user need statements that may be inadequate and compromise on future flexibility through lack of parameterisation. This creates difficulties in the conversion of initially customised software into potentially packaged software products (BANCS 2000).

### Customisation and Version Management

One of the major trade-offs in software product development is between adding new and customer demanded features as and when they are discovered, either incrementally in the existing product, thus creating a range of minor version releases; or retaining each version for longer periods while customising those features for those who demand them, thus creating a range of customised version releases. Firms do not seem to have arrived at an effective way of making this trade-off.

In cases where some of the customisation is requested by the client firm in order to trade off against making changes in the existing work flow, the software firms have sought to persuade the client into making workflow changes on the grounds that the software incorporates 'best practice' in workflow design (Marshal, BANCS 2000). However, in cases where the client firm does not agree to making workflow changes, the smaller firms are more likely to agree on greater customisation (MakESS) while the larger ones are likely to either opt out of such projects or not to market their products to firms that are likely to make such demands (Marshal). Smaller firms can enjoy an edge over larger firms by agreeing to higher levels of mutually justifiable customisation, compared to larger firms that seemingly cannot manage the complexities of customisation given their larger client base and their larger development and maintenance groups.

The degree of customisation that the software allows its users is a key factor for its acceptance. In a high value, complex product, it may be advisable to allow a higher degree of customisation as this could provide a competitive edge (BANCS 2000). Marshal seems to have used this effectively by charging well for customisation. MakESS seems to allow a greater degree of customisation to take place if demanded and relies on the higher user friendliness of the software to reduce the incidence of customisation as it takes away their limited number of good software engineers from development. However version control is a major concern from the software maintenance point of view.

## Marketing and Product Launch

In practically all the cases, there seemed to be an under-utilisation of partnerships with other firms, particularly for marketing. Forging such partnerships would overcome the problem of setting up expensive marketing channels all over the world for what is essentially a limited product pipeline. While large companies like Ramco and Infosys could conceivably hope to set up an international sales network in the medium term, for small companies like Concept Software, partnerships are perhaps the only way to reach the market effectively, though new channels like sales over the internet is an alternative.

Timing of product release is important in external markets. By the time Ramco launched its version 3.0 in international markets, the market for ERP software was beginning to mature. Their late entry was partly compensated by their simpler design and reduced cost of implementation but other problems like lack of endorsement from the 'Big Six' consulting firms, lack of an implementation methodology and inflexibility in hardware and software platform choice could not be comprehensively overcome (Sadagopan 2000).<sup>4</sup> These lead to complex and slower implementation processes at client sites.

## Reward and Incentive Systems

Top management support is important but not sufficient to ensure the growth of a product culture within a services company. Given the longer gestation period for products and the greater uncertainty involved, reward and incentive systems have to be made more attractive to get the best developers involved in the product development activity (BANCS 2000). On the other hand developers are interested in acquiring new skills and are therefore willing to work at comparatively low salaries as long as they are in projects that use the latest software tools and technologies (EX) that have broader applications and can help them secure higher paying positions in foreign countries. Firms may encourage high turnover of software professionals at junior levels to bring in new software professionals who are younger and probably more hardworking. Projects that offer limited scope for lateral or upward (or outward) movement, due to their narrow application scope or relatively dated nature of tools and technologies used, have difficulty in attracting developers. This is especially true when the mainstream customised development activity of the firm offers developers better and more varied application scope and range of software tools and technologies (BANCS 2000).

## Quality Systems

An essential aspect of developing world class products is the establishment of effective quality systems within the product development project and across the organisation. The quality systems and productivity measures used for services can be inappropriate for products. Large firms like TCS and Infosys have developed

elaborate quality systems for their customised software development projects. They have also sought to extend these systems to software product development. However this is difficult in most cases (Nambisan, 2001). Initial (first version) software development often takes a relatively adhoc development route compared to service projects, due to the high levels of uncertainty regarding potential customers and their requirements and the large range of software development paths to choose from. Often the firm's standard quality systems come in at the point when a first workable version is available, and the systems are then quickly adapted to meet the need of controlling quality of commercial versions and releases. At that stage the quality systems start resembling the firm's systems for customised software development. On the other hand, pure product companies like Ramco and ESS have an internally developed quality system in place from the initial stages of product development and have improved these systems with their product development experience.

#### **4. Moving Ahead with Product Development**

##### **4.1 Coping with Challenges**

All the companies that we have studied till June 2000 have persevered with their products with reasonable success (though profitability information is not available). A number of other Indian software companies that started on the product route (e.g. Mastek and Wipro Systems) have since then changed track and become predominantly software service companies. Software services companies such as Infosys that at one time set ambitious targets for revenues from software products have changed their mind. Some smaller companies have persisted with niche products with mixed results. Our study confirms that successful product development is not easy. Should Indian software companies fight against these odds and commit resources to product development?

Given the physical distance of India from the world's largest market for software products, the United States, it is unlikely that Indian companies can be successful at the development of products with very short life cycles and low prices. There does however seem to be an opportunity for Indian companies to use their high quality software developers to develop products of intermediate to high complexity with longer life cycles. Smaller companies could focus on products of intermediate complexity targeted at niche markets. The emergence of the internet and electronic commerce as a sales and delivery channel opens a new route to reach out to customers outside India. Product functionality can also be made available to customers all over the world through Application Service Providers (ASP). The ASP route opens up new possibilities for alliances and partnerships.

Indian companies can also use their software technology skills to drive technology-based products such as tools and specialised utilities. With the increasing penetration of computers in the Indian market both among businesses and homes, there will also be greater opportunities for products targeted at the Indian market.

But the strongest argument in favour of developing products is the gradual erosion of the cost advantage of Indian software companies primarily due to increase in employee costs. If it were not for the tremendous demand-supply gap in the global software industry, perhaps Indian software companies would have been facing a different situation than they are today. New entrants into the software industry like companies from China are reported to be making rapid advances in developing their skills and may have cost advantages over Indian companies. Product development is one way for Indian software companies to strengthen their competitive position in the long run.

Indian software firms may eventually have to move towards becoming "solution companies" rather than service or product companies. Solution companies build a portfolio of product components and combine them in unique ways to provide complete software solutions for their client's specific requirements. This requires a strong service orientation in a product oriented company and an ability to conceptualise the entire solution space in the software firm's chosen market and develop a stream of product components to address those requirements. Solution companies can potentially exploit synergies between service and product projects and Indian firms that are currently active in both areas have an opportunity to transform themselves to solution companies that are both effective and efficient.

What Indian companies and the government can do to further the development of a product orientation is discussed in the following sections.

#### **4.2 Company Initiatives**

Companies involved in software product development need to have a subtle understanding of the issues related to customisation and version management. Customisation is not necessarily contradictory to a product approach; very successful products like SAP R/3 require a high degree of customisation. What is more important though, is understanding which features need to be in the core product, what parameters need to be user definable, and how to manage customised add-ons and modifications through the version upgrade path. Managing these trade-offs again requires close inter-functional working between the design, engineering, marketing and support groups. These capabilities will improve as companies go through repeated product iterations. To ensure that this learning takes place, companies need to put in place comprehensive review mechanisms.

Companies need to be careful in converting customised solutions developed for particular users into products. While the learning gained from the customised solution will have value in developing a related product, the reuse of customised software for developing more generic products is likely to result in limited acceptance by a wider group of customers, as customer specific compromises emerge as inappropriate for a larger customer group.

There is a need for better version management and strategic product planning, which

are currently under-developed areas in most companies. This needs persons who have an overall understanding of the structure and technology of the product, an in-depth understanding of the market and its trends and an ability to forecast customer needs and tap on latent needs. These persons can plan versions that are technically designed to allow for faster development and release, yet are robust to allow customer or developer based modifications, and flexible enough to allow enhanced versions to be developed without complete re-engineering. This indeed is a tall order for one person but firms that have the benefit of one or more of such people have a definite edge in software product development over the long run.

Product development requires discipline – in analysis, decision-making, and implementation. It involves intuitive understanding of markets, users and their needs combined with creative problem-solving, elegant design and robust architecture. It needs to be led by developers and engineers who are sensitive to the market, and marketers who are appreciative of engineering constraints. More effective cross-functional working and integration across these groups and with customers would facilitate better and early product definition.

Companies interested in developing the product business need to create a broader product pipeline to help justify the investments in creating a product-marketing infrastructure. Products with medium to high level of complexity and with well defined and easy-to-address markets are good candidates for development by Indian companies. Companies with non-competing products could look at forming a marketing consortium to lower international marketing and support costs.

Smaller companies seem to be good at building inimitable competencies and appropriate domain expertise in niche areas that are too narrow to be of interest to large firms. Large companies seem to be good at developing appropriate tools and building new (software) technological frameworks that are too large in scope for small firms to build internally and exploit effectively. Both small and large companies need to find ways of working together, to use these complementary and mutually reinforcing advantages. Large companies interested in the product business may find it advantageous to identify as acquisition targets the small companies that have strong competencies and domain expertise in areas that are emerging as important.

Service companies interested in developing the product business need to create separate incentive and reward systems to help motivate employees working on product development. In addition to ensuring that the product development takes place on the latest platforms and with contemporary technology, given the longer time horizon of product development, product developers need to be compensated with stock options on terms different from those used for people in the service business. This would ensure that the immediate revenue based incentive structures in the services businesses do not pull away promising product developers into services.

To make the product development activity more market-oriented, companies may like to hire more management graduates in the product development business as well as

encourage senior developers who take product leadership roles to undergo some formal management education.

### **4.3 Government Policy**

There are both direct and indirect ways in which the government of India can support the development of a product orientation in Indian software companies.

In the early years of the software industry, the government provided export-marketing support through a finance scheme operated by the Export Import Bank of India. Other development financial institutions have also provided various forms of support for the Indian information technology industry (George and Prabhu 2003). Given the difficulties in creating an infrastructure for product marketing in foreign markets, the government may like to formulate appropriate schemes for providing one-time support to Indian companies for creating their marketing infrastructure in foreign countries.

Steps to deepen the internal market for software products would provide impetus to Indian software companies for developing more products. Result oriented incentives for potential client organisations to adopt information technology effectively can spur software development. One of the key measures in this direction is the prevention of software piracy and the protection of intellectual property. The government should continue and intensify policies to check software piracy and identify new ways of protecting intellectual property in software. As a large customer, the government norms for purchase of software for internal use should protect intellectual property and select best of class software available anywhere in the world rather than favour indigenously developed software, as this will force software vendors to develop world class products.

It is evident from our study that while Indian software product companies ideally need both a high degree of domain expertise and technological competence, the latter is more common than the former. The relatively low sophistication of most Indian user industries makes genuine domain expertise hard to find. Government policies that continue to emphasise that Indian industry needs to be globally competitive, and policies that open the Indian market to foreign competition, are likely to increase the sophistication of domestic industry. These can indirectly help software firms acquire the essential domain expertise to be internationally competitive in software products.

### **References**

Accenture (2002) "Making Indian Manufacturing Companies Globally Competitive". New Delhi & Mumbai: Accenture.  
[http://www.accenture.com/xd/xd.asp?it=enweb&xd=locations\india\india\\_ideas\\_3.xml](http://www.accenture.com/xd/xd.asp?it=enweb&xd=locations\india\india_ideas_3.xml)

Arora, A., V.S. Arunachalam, J. Asundi and R. Fernandes (2001), The Indian Software Services Industry, *Research Policy*, 30, pp. 1267-1287.

- Arora A., A. Gambardella and S. Torrissi (2001) *In the Footsteps of Silicon Valley? Indian and Irish Software in the International Division of Labour*, SIEPR Discussion Paper No. 00-41, June, Stanford CA: Stanford Institute for Economic Policy Research.
- Bowonder B and P. K. Richardson (2000) "Liberalisation and the Growth of Business Lead R&D: The Case of India" *R&D Management*, 30 (4), pp. 279-288.
- Clark, K. and T. Fujimoto (1990), 'The Power of Product Integrity', *Harvard Business Review*, 68 (6), pp. 107-118.
- D'Costa, A. (2002) "Export Growth and Path Dependence: The Locking in of Innovations in the Software Industry" *Science Technology and Society*, 7(1), pp.51-89.
- Desai, A. (1998), *Presentation at the Workshop on the Context of Innovation in India: The Case of the Information Technology Industry*. New Delhi on 24 July 1998.
- Forbes, N (1999) "Technology and the Indian Industry: What is Liberalisation Changing?" *Technovation*, 19, pp.403-412.
- George, G. and G. N. Prabhu (2003) 'Developmental Financial Institutions as Technology Policy Instruments: Implications for Innovation and Entrepreneurship in Emerging Economies', *Research Policy*, 32 (1), pp. 89-108.
- Griffin, A. and J.R. Hauser (1993), 'The Voice of the Customer', *Marketing Science*, 12 (1), pp.1-27.
- Hamel, G. and C.K. Prahalad (1989), 'Strategic Intent', *Harvard Business Review*, 67(3), pp. 63-76.
- Jishnu L. (2002) 'The Greening of Achamma' *Business World* 2 September, pp.50-52.
- Kanter, R.M., M.J. Kao and F. Wiersema (1997), *Innovation: Breakthrough Thinking at 3M, Du Pont, GE, Pfizer and Rubbermaid*. New York: Harper Business.
- Khurana, A. and S.R. Rosenthal (1997), 'Integrating the Fuzzy End of New Product Development', *Sloan Management Review*, 38 (2), pp. 103-120.
- Krishnan, R.T. and G.N. Prabhu (1999), 'Creating Successful New Products: Challenges for Indian Industry', *Economic & Political Weekly*, 34 (31), July 31-August 6, pp. M114-M120.
- Mehta, D (1998), 'A Time for Consolidation', *Information Systems Computer World*, 3 (17), pp. 44-44.
- Mowery David C. (1999) 'The Computer Software Industry', in D.C. Mowery and R.R. Nelson (eds.) *Sources of Industrial Leadership: Studies of Seven Industries*

Cambridge: Cambridge University Press, pp. 133-168.

Nambisan, S. (2001) "Why Service Businesses are Not Product Businesses?" *Sloan Management Review*, Summer, pp.72-80.

NASSCOM (2002), *Software Industry in India: A Strategic Review*. National Association of Software and Service Companies: New Delhi: NASSCOM.

NASSCOM (2002), *National Association of Software and Service Companies*, New Delhi: NASSCOM, Website: <http://www.nasscom.org>

Prahalad, C.K. and G. Hamel (1990), 'The Core Competence of the Corporation', *Harvard Business Review*, 68(3), pp. 79-91.

Prahalad, D.N. (1996), *Interview*, <http://web3.asia1.com.sg/timesnet/data/ab/docs/ab1008.html>.

Sadagopan S. (2000), *Website of Prof. S. Sadagopan*, <http://www.iiitb.ac.in/ss>.

Sawyer S. (2000) "Packaged Software: Implications of the differences of custom approaches to software development" *European Journal of Information Systems*, 9, pp.47-58.

Schilling, M.A. and C.W.L. Hill (1998), 'Managing the New Product Development Process: Strategic Imperatives', *Academy of Management Executive*, 12 (3), pp. 67-81.

SEC (2002), Infosys Technologies Form 20-F for year ending March 31, 2002, <http://www.sec.gov/Archives/edgar/data/1067491/000089161802002223/0000891618-02-002223.txt>

Ulrich, K.T. and S.D. Eppinger (1995), *Product Design and Development*, New York: McGraw Hill.

Wheelwright, S.C. and K. Clark (1992), 'Creating Project Plans to Focus Product Development', *Harvard Business Review*, 70 (2), pp. 70-82.

Yin, R. K. (1994) *Case Study Research: Design and Methods*, Beverly Hills, California: Sage.

---

<sup>1</sup> This consists of getting a program written for a standard business function such as accounting or inventory management rather than using a readily available software product.

<sup>2</sup> The early stages of the product development process involve ideation, understanding user needs, forecasting demand and usage trends, and coping with technological uncertainties. These stages are sometimes referred to as the fuzzy front end of the product development process in view of the uncertainties involved. Companies take steps to assess and manage risks as the product development process proceeds, and uncertainties also reduce as the launch date comes close, thereby reducing the fuzziness involved.

<sup>3</sup> D.N. Prahalad was then Senior Vice President at Infosys - Interviewed in June 1996.

---

<sup>4</sup>S. Sadagopan is an expert on enterprise resource planning (ERP) systems.