

# Software R&D Strategies of Developing Countries

1

Shirley Tessler and Avron Barr Stanford Computer Industry Project

An adequate supply of software talent is the key enabler for developing a sound technology base in software R&D.<sup>2</sup> Other factors, however, such as a nation's physical and financial infrastructures, its piracy rates and government policies, can change the timeframe markedly for building that software R&D capability. We discuss the factors that led to the success of Bangalore in India, and then examine strategies in other countries for emulating India's accomplishment. We look in detail at Russia and Malaysia: two countries with significantly different market sizes and technology development strategies. We conclude that it will be many years before another developing country achieves the kind of success in software that India has.

Software is a growing industry, and one which requires relatively low capital investment. It is also an industry that a number of nations have identified as essential to their successful participation in the future global economy. Our goal in this paper is to explore what country characteristics and strategies enable the development of a software R&D capability. We will review India's success in growing a notable software services export industry, and then will examine one large country, Russia, and one small country, Malaysia, both of which aspire to be future software powerhouses. They are taking very different approaches to building their competence in software R&D.

Software R&D

Three points are worth repeating from our December working paper. First, all software is created by programmers,<sup>3</sup> but it is used in a wide variety of ways. Some

<sup>&</sup>lt;sup>1</sup> A position paper for the January 9, 1997 meeting of the Council on Foreign Relations' Study Group on the Globalization of Industrial R&D. The authors are independent software industry analysts working with Prof. William F. Miller, Graduate School of Business, Stanford University, on a Sloan Foundation grant to study the global software industry. http://www-scip.stanford.edu/scip/

<sup>&</sup>lt;sup>2</sup> See our position paper for the December 12, 1996 meeting of this Study Group for a complete discussion of the software labor situation and its impact on globalization of R&D.

<sup>&</sup>lt;sup>3</sup> Throughout this paper we will use the term "programmer" in its most inclusive sense, to mean any software professional such as programmer, project manager, systems architect or testing engineer.

software appears as published software products, most of which is sold to large corporations. These organizations build much more software themselves, including the "information systems," which have become the central nervous systems of most business enterprises and government agencies. Although most of this custom-built software is written in-house, there is a growing software services industry that also provides these organizations with software. It is here, in the software services segment, that the Indian software industry participates, for example. Finally, there is more and more software being embedded in devices, instruments and equipment from cars to cellular phones. All of this software is built in pretty much the same way by professionals with nearly identical skill requirements. The demand for software and its central importance to many industries continues to increase.

The second point we made is that the demand for software continues to increase at an increasing rate, and a global shortage of talented software people is developing. (There is a big difference between the best software developers and average software developers.) We believe that the search for talent, not wage differentials or access to local markets, is driving the globalization of software R&D.

The final point to reiterate is that software R&D is fundamentally different from R&D in other industries: "the software R&D process differs from other technology R&D in that there is no tooling or manufacturing phase of product development; rather, when R&D is finished, the program is ready to copy, ship and use." Thus when we talk about a country building a software R&D capability, we are actually discussing its capacity to produce software products and/or services, i.e., to be a player in the industry.

# Enabling factors

To understand the factors that enable the development of a software R&D capability, we look at the example of Bangalore in India. It is often used as a development model in other parts of the world. Bangalore is a major contributor to India's \$750 million software services export industry. It also plays a role as the lead R&D area in which its business approaches and experiments are watched closely, and then the successful experiments are emulated by the other Indian technology centers.

Although Bangalore has its own unique advantages, we note some characteristics that it has in common with software R&D centers in developed nations:

1. A sufficient supply of software talent (or technical professionals who could be trained as programmers). Having a good supply of software professionals requires in turn an extensive educational system with an adequate amount of college-level training on modern computing equipment and systems, as well as the availability of more extensive, post-graduate training in high-demand skills and state-of-the-art computing systems.

In countries such as India, having a large enough supply of software talent available to create a meaningful software R&D *export* capability also implies a

relatively low rate of economic development generally. First, if there are many alternate career choices for talented young people who could become good programmers, a smaller percentage would choose to enter the software profession. Second, a more vigorous emerging economy would consume all of its available software talent domestically in the building of the banking, telecommunications, manufacturing, transportation and distribution systems.

- 2. Widespread English-language competence, which is useful in keeping up-to-date with software technology, but is essential in working with multinational development teams.
- 3. Adequate capital, an entrepreneurial culture and effective business environment to find markets for software and to encourage investment in software R&D centers.
- 4. An attractive domestic market for software, or the potential for one, which encourages foreign investment in R&D and motivates local entrepreneurs. Software piracy is a major factor diminishing the effective size of many countries' domestic markets.
- 5. A reliable, state-of-the-art telecommunications infrastructure to allow programmers to communicate and collaborate with foreign R&D partners, keep up-to-date with technology, and increasingly, to distribute locally developed software products to a global market.
- 6. Government policies that encourage investment in infrastructure and training, promote intellectual property protection, and facilitate the flow of new technology into and within the country. We will now discuss the key role of government in more detail.

# Government policy

Some government action is critically important to the emergence of a software R&D capability, while some typical government actions have had a dramatically inhibitory impact on the industry in many countries. Government helps the software industry most when it confines its contribution to promoting higher education, funding a diverse, researcher-motivated range of basic research activities, and providing a regulatory environment conducive to corporate investment. Although there are exceptions, major government programs to pursue a specified research direction in the hopes of becoming a leader in some aspect of the technology have often harmed the very industry they intended to nurture.

The problem with government programs in information technology is that their organization and execution preclude the industry from keeping up with the rapid rate of change in hardware and software technologies, as well as with market demands. Breathtaking change has been the hallmark of the computing industry for the past 15 years. The only way to succeed in this kind of environment, where technologies and markets can't be predicted 18 months in advance, is to follow diverse strategies

simultaneously. For example, the US government has long promoted extensive diversity and innovation through its fortuitous lack of a "grand plan" for funding software R&D in universities and national research centers. In more recent years, venture capitalists in Silicon Valley (and elsewhere in the US) have encouraged innovation in a similar fashion since, as a group, they invest in multiple approaches to solving the same technology problems. They each know that not all of their technology choices and market predictions will pan out, but that they will reap adequate return on the few companies whose strategic choices do lead to success.<sup>4</sup>

Detailed government programs for software R&D also inevitably end up with specific government decrees on standards or directions in hardware platforms, programming languages, or other aspects of computing technology. The more consensus there is on which direction to bet on, the more risk is being taken. Japan, for example, developed a detailed IT industrial policy in the 80's, called the Fifth Generation Project. Among the "big bets" mandated by the policy was the use of the Prolog programming language and the development of hardware platforms optimized for use with Prolog. By the time it was obvious that Prolog had failed to become a major software technology, Japan had fallen significantly behind in some key areas of computer science research, like object-oriented programming. It had funded its targeted programs lavishly while foreclosing parallel efforts in other computing technologies. In contrast, failed efforts during the same time period in the US to commercialize the LISP technology barely made a ripple in the ongoing evolution of the domestic software industry because so many alternatives had been under development simultaneously.<sup>5</sup>

Other counter-productive government policies to nurture high-tech R&D include protective tariffs and import restrictions. India, for example, attempted to encourage a domestic computer hardware industry through both high tariffs and strict import quotas as far back as the 1960's. These actions, coupled with the 1971 designation of ECIL as the single authorized computer manufacturer in India, left the country years behind technologically, and still unable to develop a viable hardware capability. Meanwhile, Indian programmers had no access to modern systems for training and development. India's hardware and software industries only began to come to life when these restrictions were gradually reversed in the mid- to late-80's.<sup>6</sup> Similarly, Brazil used a "market reserve" strategy to severely limit both hardware and software imports until 1990, with the same adverse results as India. Since the easing of many of these restrictions, Brazil has experienced a boom in both industries.

Censorship, especially restrictions on Internet access, is shaping up as the key counterproductive government policy of the 1990's. China, Singapore, and Malaysia are among the countries expressing concern about the societal implications of free access

<sup>&</sup>lt;sup>4</sup> See Barr & Tessler, An Overview of the Software Industry Study

<sup>&</sup>lt;sup>5</sup> See Tessler, Barr and Feigenbaum, US Software Products Industry: Success Factors in an Era of Rapid Change

<sup>&</sup>lt;sup>6</sup> See Hanna.

to information on the Internet. Unfortunately, governments that restrict access for censorship purposes may cause their citizens a far greater problem by unnecessarily inhibiting their ability to compete in the high-tech global marketplace. Software entrepreneurs in countries with small domestic markets might be especially harmed, since the Internet already provides the opportunity to develop and distribute software for a worldwide market and is itself the new "platform" sparking more creativity and invention in software than anything that's come along since the PC.

## Bangalore

Bangalore has become the symbol for India's successful development of a software technology base. Software services exports to the US, Japan and Europe in 1996 have grown to about \$750M, of which Bangalore contributed one-third of the total. In 1988, India's revenues from software services were less than \$50M. Basically, this software services activity makes India something of the equivalent of an Andersen Consulting or EDS in the software industry, offering software development services for hire abroad. India still offers almost no software products for export, and has only a small (but growing) domestic market for software. Still, its software services export industry is a major contributor to India's economy and a phenomenon unmatched by any other developing country.

Education and the labor pool. Bangalore's physical infrastructure was built up by the British as an armaments center because of its centralized location in the interior. Defense considerations also stimulated India's investment in engineering education after three wars with neighbors in the 60s. Many of these new technical universities were established in the Bangalore area. The result, a generation later, was a sizable engineering population that could not be utilized effectively by India's economy. Even after the "brain drain" of top talent to the US and Europe, there were many more talented engineers in India, especially in Bangalore, than there were projects to employ them. This labor surplus was the seed of the Indian software industry.<sup>7</sup>

Entrepreneurs find a market. Capitalizing on wage differentials, Indian entrepreneurs entered the outsourcing market of the late 1980s with a service product targeted for IS departments looking for ways to reduce costs. US, European and Japanese MNC's made significant investments in communications infrastructure, in "mirror" systems in India which matched the clients own computing environment, and in training in the latest software skills.

Another form of investment stemmed from the realization that software development work required very close communications between client and programming team. The best English-speaking engineers were sent to the US or to Europe to work on the client's site, becoming familiar with the systems, the operations, the people, and the newest technology. Many then returned to India to be managers of software teams in Bangalore or to start their own software businesses.

<sup>&</sup>lt;sup>7</sup> See Raghavan.

Government investment. India had relatively little telecommunications infrastructure (even in Bangalore) but was willing to finance some development and allow foreign companies to do the rest. In addition it established a number of Software Technology Parks (STP) and Export Zones where foreign investment was facilitated. The resulting investments in software development facilities in Bangalore by multinational companies significantly contributed to the city's becoming an internationally known center of excellence.

Quality and timeliness. After a few years of being the low-priced vendor of software services, the Indian entrepreneurs and their corporate IS clients realized that quality software delivered on time had become a rare and critically important business service. The Indian software firms no longer had to offer huge discounts. Profits went up and continued investment attracted more and more engineers, and even a few doctors and lawyers, into the ranks of India's now relatively well-paid software developers. Wages more than doubled over the past 5 years, but still remain very low compared to Western pay rates.

*Piracy.* India is a high-piracy country. Piracy shrinks the size of its domestic market, but more importantly, raising concerns for firms who don't want their software ideas spread around. The issue has been diffused, again by clever entrepreneurial action, with special intellectual property agreements signed by individual Indian software companies and endorsed by former clients in respected Western companies.

The future. India's software services industry continues to grow rapidly, in spite of power brownouts, water shortages and lack of an airport in Bangalore, and uneven telecommunications capabilities throughout the country. Overall, the physical infrastructure within the STP's is improving every year, but fiber optic networks and satellite links require great diligence to maintain in the Third World. The biggest challenge to the future of this industry, however, comes from its greatest strength, the labor force.

The business that the Indian software industry has built is a services business — developing custom software systems. Services is a business that grows proportionately with the number of programmers, as opposed to a software publishing business, for example, that can grow much faster. Despite some interesting stories of reverse immigration, India still has a brain drain of top software talent, and she cannot produce an infinite supply of talented software developers. Talent will ultimately be the limiting factor for the Indian software services industry.

Continued investment in training, quality assurance and software development methodology can extend the industry's growth by maximizing the size and skills of the labor pool and each programmer's productivity. Developing more software products would leverage its labor pool considerably more. India is only just now at the point of being able to capitalize on its software expertise to enter the software publishing business for domestic consumption of business software such as relational database management systems and accounting packages. A recent Indian government decision to computerize it operations will guarantee strong future growth in that business. Relatively little packaged software will be developed for export, as India continues to have an underdeveloped presence in the global software products market. Nevertheless, India's successful and rapidly growing software services export industry is a model for other developing economies.

## Russia

On the surface, Russia's situation today appears similar to that of India's in the 1980's. It has a surplus of programmers, great interest from foreign investors, and a government in the process of dismantling its most onerous barriers to high-tech development. Through interviews with a number of analysts and business people involved in building Russia's commercial software R&D capability, we have come to believe, in fact, that Russia's development path is likely to be very different.

Labor pool. The reason that Russia and other former Soviet states, like the Ukraine, are often cited as future software powerhouses is the large number of very bright "programmers" now available for commercial opportunities after the collapse of the Soviet defense/aerospace program. Analysts estimate that the former Soviet States together possess two million programmers, including 300,000 to 400,000 in the Ukraine and most of the rest in Russia.<sup>8</sup> These programmers were trained, through the high-quality Soviet educational system, in a variety of disciplines, most notably, physics and mathematics. Relatively few, however, were formally trained specifically in computer science or software engineering.

Much has been made of the high skill level of Russian programmers.<sup>9</sup> They are often portrayed as having superior abilities to create innovative solutions to programming problems because they were forced to work around the limitations of having inadequate computers or no computers at all. This characterization certainly stokes Western interest in Russian programmers, but it is an overstated and romanticized notion.

There are great programmers all over the world, but not great software industries. It is true that Russian software wizards built a major defense/aerospace computer infrastructure in isolation from the West, rarely benefiting from state-of-the-art equipment or tools. This success was a monumental intellectual achievement. Unfortunately, decades of government mismanagement pushed programmers towards poor software development practices and instilled a bureaucratic mindset that stifled innovation and severely limited the robustness and maintainability of most Russian software. (For example, Russian programmers had serious disincentives to building software systems that could be used reliably without their continued presence.)

<sup>&</sup>lt;sup>8</sup> See Rupert

<sup>&</sup>lt;sup>9</sup> We will address the situation in Russia, since we are most familiar with it and because it still possesses the majority of the Soviet programmers.

English competence and business experience. Western companies who have explored the use of Russian software labor invariably acknowledge the strong educational background, but say that it's not enough. Russian programmers are several years behind in key software technologies. They have few learning resources, such as publications, trade associations or continuing education programs, to bring themselves up-to-date on those skills. Much of their skills lies in developing and maintaining big custom defense systems, or in reverse engineering and adapting Western packaged software. Only a handful of systems architects, programmers, and project managers, have experience in independently developing commercial software or in dealing with software development in a competitive, time-sensitive environment.<sup>10</sup> Furthermore, as a group, Russian programmers are not as proficient in English as their software development competitors elsewhere in the world. The language barrier is often given as the source of problems in existing R&D alliances, or as one of the reasons that Russia was not considered for an R&D project at all. Finally, and most importantly, their civil service employment background give them no experience in how to deal in a commercial environment with customers, budgets, and deadlines. As one interviewee put it, Russians were "not trained to deliver." This lack of a customer-oriented spirit is the single most often cited drawback of Russian software labor.<sup>11</sup>

*Government.* While problems with programming skills can be ameliorated with training, problems with government are still often seen as insurmountable. Both corporate executives and industry analysts cite government disarray and corruption, especially in customs and taxation, as serious impediments to establishing an R&D presence in Russia. Moreover, Russia's famous bureaucracy makes it unclear to outsiders which ministry should be approached to find the right people, or how to negotiate an alliance with a high-tech cooperative, government institute, or institute "spin-off." Software entrepreneurs and investors live in a fast-paced world and cannot spend the time required to work through these intricacies — other countries are cutting the red tape in order to attract high-tech investment.

Domestically generated R&D is as much in jeopardy as Western sourced R&D. The post-communist government continues to fund university and institute-based basic research, but has almost abandoned applied R&D under the theory that it will be supported by industry. Unfortunately, industry does not have the resources presently to fund this type of activity.<sup>12</sup> Thus, the majority of funds for applied R&D activities come from foreign corporations and from foreign governments, especially from "privatized" US grant programs aimed at converting the Soviet military capability to a commercial orientation. The effect on an industry of having the majority of its R&D efforts dictated through the funding choices of foreign entities is not clear.

<sup>&</sup>lt;sup>10</sup> Tetris is one of the few examples of a Russian software products company (games) that has had international commercial success.

<sup>&</sup>lt;sup>11</sup> See Tessler and Barr in Software Projects in Russia: A Workshop Report

<sup>&</sup>lt;sup>12</sup> See Balzer

Government funding of a wider array of civilian projects in the future would certainly help ensure that more internal needs are met.

The government provides another type of impediment to the development of a modern software technology capability in its continued domination and mismanagement of the telecommunications infrastructure. At the beginning of the decade, the USSR had 15 telephones for every 100 Soviets, compared to a Western average of 50 phones per 100 people. Today, in Russia that number is essentially unchanged.<sup>13</sup> Adequate telecommunications is essential to developing effective links to R&D collaborators abroad. While Russia continues to struggle to upgrade its 1930's-era infrastructure, the less well-developed Asian countries are able to leapfrog from almost no telecommunications to the most modern, high-bandwidth fiber optic and satellite networks.

*Piracy.* In early 1996, The Business Software Alliance declared that Russia was still a "one-copy country," meaning that piracy was still rampant, in spite of several years of US government and software trade association negotiation efforts to curtail intellectual property abuses. In 1993 Bill Gates went to Russia to recruit Microsoft pirates to join the company as way to reduce copyright infringements on its products. He received little response since many of the counterfeiting efforts were financed and run by the government itself. Microsoft and others continue to work for better intellectual property protection in Russia, because of its huge future potential as both a buyer of commercial products and a supplier of software talent. The difficulties in securing acceptable licensing agreements and adequate enforcement of IP regulations, however, has clearly inhibited the entrance of many foreign software firms. Moreover, it has put a serious damper on domestic R&D efforts for packaged software, especially for PC platforms.

The domestic market for software in Russia remains disproportionately low for an economy of its size and level of development. Piracy is the main cause. According to IDC, only half a billion dollars of software was sold in Russia in 1995. At \$613 million last year, even the India domestic software market was bigger than Russia's. Much of the software purchased in Russia is for bigger systems that are harder to pirate. Easily-pirated PC software continues to have a tough time gaining a market foothold in the former Soviet Union.

While piracy has long been an accepted course of action in Russia, (partly because Western products were otherwise unavailable), in recent years a few vocal Russian software entrepreneurs have begun to lobby the government for IP enforcement, as they have finally recognized that the development of a viable domestic PC software market depends on it. We cannot even begin to speculate when piracy will be brought under control in Russia. In the interim, however, a huge demand is clearly looming for the larger, less-pirated systems necessary for a modern business and government information infrastructure, towards which end both Russian and (Russian-based)

<sup>&</sup>lt;sup>13</sup> See Communism, Computing, and the Case of Esther Dyson in The Economist.

foreign software firms could fruitfully direct the majority of their R&D efforts for a number of years to come.

*Capital.* Apart from foreign grant and loan programs, and foreign firms attempting to start joint ventures, little financing exists for software R&D efforts. There is no domestic venture capital financing, no support from the troubled banking system, and little capital available from the fledgling equity market. As we mentioned previously, the only capital available from the government is directed towards long-term basic research, and that funding is also not up to the level of spending of the developed nations.

Two factors mitigate the funding problems: first, relatively little capital is required for software R&D compared to other high technology endeavors. The more entrepreneurial can still bootstrap their efforts until a program is developed enough to generate outside interest and financial support. Second, the vast potential of the Russian market is sufficient to force continued corporate investment, in spite of failures and difficulties, in order to ensure a solid market presence when the anticipated market boom materializes.

*Timeframe.* While no master plan has been articulated by the government, Russian software professionals believe that a suitable strategy for the development of the Russian software industry would be to start with an Indian-style software services outsourcing business, then move on to creating Russian versions of Western software, and finally, applying the expertise so acquired to developing a domestic and export business of both software products and high-end custom services. The timeframe for such a scenario, however, is likely to be far longer than these industry participants anticipate.

Russian programmers, most of whom were, after all, trained as physicists and mathematicians, have bought into their own reputation as programming elite: they are scornful of the "good-enough" quality of much of Western technology, and furthermore, they believe that the routine software development tasks that are the bread and butter of the Indian services firms, for example, are beneath their abilities. Russian programmers could jump-start their entire export industry on just Year 2000 conversions, for example, but have made no noticeable progress in obtaining that type of business. India and Eastern Europe firms will make a fortune in this area in the next few years.

In addition to these attitude problems about what would be suitable programming work, additional roadblocks will slow the development of a software services export industry: the language barrier, the lack of entrepreneurial spirit and know-how, inadequate infrastructure development, and substantial government regulatory impediments. The most likely scenario for the development of the Russian software industry still lies with the investment of the high-tech MNC's in alliances with Russian software groups to perform software R&D for domestic commercial use. Companies like Oracle, for example, have managed to establish working relationships with government officials and Russian distributors through significant

### Software R&D Strategies of Developing Countries

and sustained investments. Their efforts are directed mainly towards training software programmers to use their products, and training distributors' sales personnel on how to support their customers. At this point, these investments are not producing significant financial returns, but they are establishing market share for the company, and at the same time, furthering the long-term development of the Russian software technology base.

In the next decade, the programmers and sales people whom companies such as Oracle, IBM, Hewlett-Packard, SAP, and others, are training in Russia will become the source of new domestic ventures to offer services and develop value-added solutions for the same customers, based on the offerings of their Western former employers. Software R&D aimed at new products for domestic business and consumer markets will lag considerably (and may well be oriented towards niche markets) because of slow growth in the installed base, continued piracy, and the dominance of Western software firms such as Microsoft in the larger software segments. Substantial success in exporting packaged software, or significant software services exports, is very unlikely for even more than a decade.

#### Malaysia

Malaysia provides an interesting contrast to Russia. It has a small domestic market potential, a tiny software labor pool, and a government intent on planning and controlling every aspect of the country's proposed ascent to a Asian software power by the year 2020. It also has one of the fastest growing emerging markets in the region, fueled in large part by foreign investment in semiconductor and consumer electronics manufacturing and export.

Malaysia's exceptionally bold plan calls for the development of a futuristic capital city and technology center called Putrajaya. It will encompass the "Multimedia Super Corridor (MSC)," a special high-tech R&D zone between Kuala Lumpur and Putrajaya. This massive top-down government effort, called Vision 2020, appears to be aiming for a cross between Silicon Valley and Brasilia. Malaysia anticipates a \$11.5 billion investment in Putrajaya, which promises "world-class" facilities. Over 150,000 people are expected to work there, and 100,000 more will reside in the city. Building of the airport and office towers is already nearing completion, with the rest of the city and infrastructure still in a very early development stage.<sup>14</sup>

The MSC is designed to be the center of R&D activity for the country's informationbased industries. The plan is to focus on multimedia education, research and cuttingedge applications in a number of key sectors such as healthcare, government and manufacturing. To attract both foreign and domestic R&D projects, the corridor will boast state-of-the-art telecommunications links, and an appropriately liberalized tax and ownership environment. The Malaysian government also promises to put in place regulations that will allow importation of foreign skilled labor for 10 years,

<sup>&</sup>lt;sup>14</sup> See speech transcript by Prime Minister Mahathir

guarantee economic incentives for 5 to 10 years, and provide strong intellectual property protections.

Labor. This nation of 19 million people has a labor force of approximately 8 million, including one million mostly unskilled foreign workers. The country also hosts up to another one million illegal workers not included in the official headcount of 8 million.<sup>15</sup> The labor shortage in Malaysia is acute: the demands of local and foreign industry has caused a severe shortage and significant upward pressure on wages. The shortfall has been growing for the past decade as foreign firms have located manufacturing facilities, such as chip fabrication plants, in Malaysia to capitalize on the low-cost labor. Building of new high tech manufacturing and assembly facilities has already slowed down, moving to now lower-cost environments such as Indonesia, Thailand and Vietnam.

While no accurate figures are available on the number of software people in Malaysia, we estimate that population to be about 50,000. Moreover, their educational background is not strong. The Malaysia Education Ministry recognizes that the educational system needs to be completely revamped, and has started to make sweeping changes in primary, secondary and college-level programs.<sup>16</sup> University training for computer science and software engineering is also inadequate. Only now, in connection with Vision 2020, will private universities be allowed, and graduate studies in software-related subjects be available and supported.

The country also has a severe "brain drain" problem, especially in information technologies, as the more talented look for a better technical education and more challenging work abroad. The government has pledged to network all of its schools, and upgrade the curriculum, and reverse the flow of software talent leaving the country, but it is very difficult to achieve results in the short-term. Continued investment in educational reform will have a high payoff, however, in the 10 to 20 year time horizon.

The success of the Vision 2020 plan is expected based on the success of earlier efforts to turn the economy from a commodities producer to a manufacturing exporter in just a decade. However, the labor force that was qualified to staff the assembly plants is not the highly-skilled systems and software professionals that will be required in abundance in the new economic plan.

The current software labor force does not have the training and up-to-date skills required for Vision 2020. Moreover, Malaysia does not have enough programmers, either to develop the basic information infrastructure that the country currently lacks, or to engage in "world-class" software R&D within the MSC. Malaysia's strategy to increase the number of qualified programmers in the short-term is to import foreign "knowledge workers" by enticing high-tech companies with the state-of-the-art

<sup>&</sup>lt;sup>15</sup> See Clifford

<sup>&</sup>lt;sup>16</sup> See Shukor Rahman

telecommunications and information infrastructure in Putrajaya, and the potential of especially high financial returns from joining the MSC within the next year. This strategy may be quite risky, as we argue below.

First, we believe that the main driver for firms in the developed nations to source their R&D offshore is the global search for software talent. Malaysia cannot offer sufficient software talent. Instead, MNC's would need to bring software people to do the research as well as to train and manage the local workforce. Second, foreign companies invest heavily in training and infrastructure in countries such as Russia and China in the hopes of capturing a share of a potentially vast domestic market. Malaysia cannot offer a sizable market for Western software products and services. Finally, Malaysia, like many other small nations, is concerned about the extremely high proportion of foreigners in its current workforce. One of the secondary goals of Vision 2020 is to reduce its long-term reliance upon foreign workers. Foreign skilled professionals might well be resented for earning the riches that enticed them there, while their contributions to the country's economy may not be as obvious. It is unclear how welcome they will be when the current guarantee on their presence expires.

*Government.* In Malaysia the government is the key enabling factor in building the nation's technology base. It will finance and direct the development of the telecommunications and information infrastructure, lead efforts to attract foreign investment, offer regulatory relief and tax incentives, guarantee intellectual property protection, and undertake a comprehensive long-term program of educational reform. It will no doubt also be the primary source of any *inhibiting* factors which might arise in the course of building Malaysia's technology base, through censorship or other legal restrictions, ethnic or religious bias, or counter-productive technology policies.

Malaysia's leader, Prime Minister Dr. Mahathir Mohamed is the driving force behind the country's move to high-tech. He is the visionary who conceived of Putrajaya, and he is responsible for the political will that has seen the project through to this point. He is also simultaneously committed to maintaining a strict Islamic social order within the country. His government imposes a number of regulations, such as restrictions on Internet access, that we speculate will be almost certain to clash with the liberal policies he intends to promulgate to attract foreign R&D investment to the Multimedia Supercorridor.

Malaysia is composed roughly of 35% Chinese, 10% Indians, and 50% native Malay (and 5% other).<sup>17</sup> The government points to this multicultural population as one of the strengths of the country which will attract foreign investment. It hopes that high-tech company's wanting to do business with China, Indonesia or India, for example, will locate their base operations in Malaysia, where the multi-ethnic Malaysians can act as intermediaries. The same multiculturalism is not necessarily a strength in other contexts, however. The Malays control the government, and Chinese and Indian

<sup>&</sup>lt;sup>17</sup> Source: Department Of Statistics, Malaysia.

software businessmen that we have interviewed complain that they will be largely left out of the coming multimedia boom unless they have strong Malay contacts within the government.<sup>18</sup> As the high-tech labor shortage grows even worse, perhaps these social problems will be put aside in favor of utilizing the available local labor pool most efficiently. The government, however, does not officially acknowledge the existence of this problem.

As discussed earlier in this paper, government-created technology policies are extremely difficult to keep up-to-date because of the severe mismatch between the rapid rate of technology change and the more deliberate pace of government bureaucracy. Malaysia, however, has considered that mismatch in its Vision 2020 plan, and has made provisions to review the key components of its plan on a yearly basis. With the software world now talking in "Web-years" (equal to three calendar months), this top-down government effort may still not be nimble enough, but it is an admirable attempt which analysts will continue to watch closely.

Domestic market. Malaysia's domestic software sales are small but growing in double digits. There is especially robust growth in the sales of the large, enterprise systems which are less vulnerable to piracy. A number of foreign software firms are in fierce competition with each other to gain market share in Malaysia. They undertake the training of both their own employees and their customers' programmers, and their overall investment level is sufficient to contribute noticeably to the country's development of a domestic software technology base. Of the significant R&D alliances that are undertaken, the majority are for software developed for domestic use rather than for export. The focus of resources in the MSC, and the government's plan to build a paperless administrative capital at Putrajaya, will fuel a domestic software boom in Kuala Lumpur, and will most likely have beneficial spill-over effects to the other high tech areas of the country, especially Penang.

*Timeframe.* Putrajaya, with its Multimedia Supercorridor, is an ambitious plan that relies heavily on enticing foreign investment and *foreign labor* to develop Malaysia's software R&D capabilities into an Asian Silicon Valley. It will take many years to train enough people internally, and it is unclear how much foreign labor it will realistically be able to attract. Moreover, cultural factors that hamper progress in other areas will not necessarily disappear in Putrajaya. Thus, we expect only modest growth in a domestic software R&D capability in the next decade, with few breakthrough technologies coming out of the MSC, some products and services exports to Asia, and almost none to the West. Malaysia will continue to rely on US, Japanese and other MNC's to play a dominant role in its domestic software R&D, well beyond the expected 10-year ramp-up period towards self-sufficiency.

*Malaysia versus Singapore.* In 1986 Malaysia's next door neighbor and perennial rival, Singapore, also instituted a grand plan for its IT industry, called IT2000.<sup>19</sup> This

<sup>&</sup>lt;sup>18</sup> See Barr and Tessler Malaysia Interview Notes

<sup>&</sup>lt;sup>19</sup> See A Vision of an Intelligent Island: The IT2000 Report

ambitious, government-directed and -funded effort included networking all 3.5 million of its citizens, substantially increasing the supply and quality of IT professionals through educational reform, investing in more R&D, and creating a business climate which encouraged foreign IT investment. Many of these steps were similar to what Malaysia now plans. The two nations are similar in other ways too, in that both have essentially single-party governments, restricted Internet access, a growing labor shortage, and not quite as much acceptance to high-technology in many sectors of the economy as their governments assert.

Singapore's IT plan appears to be making good progress. Perhaps its longer experience as a regional financial and industrial center will give its IT labor force a valuable edge. Perhaps networking the entire country rather than concentrating resources in a few particular areas, as in India and Malaysia, will prove more effective in developing a domestic market or luring foreign MNC's. In any case, Singapore proves that direct government action is not necessarily fatal to developing a solid software technology base, and that Malaysia could reach its goal to be an information-based economy by the year 2020. However, the same plan that proves a boon today could easily become an inhibitor tomorrow as technology continues to evolve and the "installed base" becomes a constraint on change.

## Conclusion

While each country's situation is different, developing nations will not have an easy time growing their software R&D capabilities. Several enabling factors must be present. Great strength in one area alone, like Russia's software talent pool, China's huge potential market, or Malaysia's massive investment, will not be sufficient to guarantee success.

It is not at all likely that significant new R&D centers like Bangalore will emerge in the next decade. The reasons, again, vary from country to country, but important missing pieces are:

- Software talent pool large enough to support an international technology center as well as service the demands of a growing domestic economy
- Adequate capital to invest in infrastructure and education
- Big enough potential domestic market to attract corporate investment
- Government policies and programs that promote diversity and innovation

In addition, if a software export industry is to evolve, both widespread Englishlanguage competence and entrepreneurial know-how are also key.

Most developing nations recognize how critical information technology is to their long-term viability. Software is both an essential element of their own infrastructure, and a driver of modern industrial competitiveness. Thus, they have no choice but to develop a domestic software R&D capability. Yet most will remain highly dependent on imported software technology. Only a handful of countries will be able to become

#### Software R&D Strategies of Developing Countries

a major software R&D center. Many developing nations point to Bangalore as a model for their own efforts. Bangalore demonstrates clearly that a competent software capability can be developed in a reasonable period of time, but its success has yet to be duplicated.

Bibliography

**Balzer, Harley**. Dismantling Russia's Technotopia: Six Ministries in Search of an Industrial Policy. **In Sedaitis, Judith, ed**. Commercializing High Technology: East and West, Selected Conference Papers. **Stanford University, Center for International Security and Arms Control. January 1996**.

Barr, Avron and Tessler, Shirley. An Overview of the Software Industry Study. Stanford Computer Industry Project. Research Brief, June 1995.

Barr, Avron and Tessler, Shirley. Malaysia Trip Report and Interview Notes, unpublished. Stanford Computer Industry Project. November 1995.

Barr, Avron and Tessler, Shirley. Singapore Trip Report and Interview Notes, unpublished. Stanford Computer Industry Project. October 1994.

**Barr**, **Avron & Tessler**, **Shirley**. *The Globalization of Software R&D: The Search for Talent*. **Position Paper for Council for Foreign Relations Study Group**. **December 1996**.

Cao, Jingjun. Offshore Programming: Development and Impact. Stanford Computer Industry Project. Research Brief, June 1994.

Communism, Computing, and the Case of Esther Dyson. The Economist. January 13, 1996.

Clifford, Mark. Can Malaysia Take That Next Big Step? Business Week. February 26, 1996.

Dave, Rishi. Patterns of Success in the Indian Software Industry. Stanford Computer Industry Project. Senior Thesis, May 1996.

Dave, Rishi. Bangalore Trip Report, unpublished. Stanford Computer Industry Project. July 1995.

Dave, Rishi. Talk slides, Analysis of the Success of the Indian Software Industry. Stanford Computer Industry Project. Fall Seminar Series. November 1995.

Dedrick, J.L., Goodman, S.E., and Kraemer, K.L. Little Engines That Could: Computing in Small Energetic Countries. Communications of the ACM. May 1995.

Hanna, Nagy. Exploiting Information Technology for Development: A Case Study of India. World Bank Discussion Papers. 1994.

Liu, Junhe and Chen, Zhiqun. Survey of the Chinese Computer Market. Stanford Computer Industry Project. Research Brief, December 1994.

Mahathir Mohamed, Prime Minister Datuk Seri Dr. A Global Facilitator of the Information Age. Speech at Opening of the Multimedia Asia Conference. August 1996.

Tessler and Barr

McHenry, William K. *The Emerging Soviet Software Industry*. Georgetown University, School of Business Administration. January 1990.

Miller, William. *The Art of Rapid Change.* Stanford Computer Industry Project. Research Brief, March 1995.

National Computer Board, Singapore. A Vision of an Intelligent Island: The IT2000 Report. March 1992.

**Raghavan, Harsha**. Is This the Next Valley? An Analysis of the Software Industry in Bangalore. **Stanford Computer Industry Project. Research Brief, May 1996**.

**Rupert, James.** Ukraine's Silicon Valley Seeks Role in Designing World's Software. **The Washington Post. July 5, 1995.** 

Shukor Rahman, A. Education system to make use of IT. New Straits Times. September 23, 1996.

**Tessler**, Shirley and Barr, Avron. Contributions to Bernstein, David. Software Projects in Russia: A Workshop Report. Stanford University Center for International Security and Arms Control. June 1996.

**Tessler**, **Shirley**. *Some Thoughts on Regulating the NII*. **In the Proceedings of Infotech** Malaysia '95. Kuala Lumpur, Malaysia. November 1995.

Tessler, Shirley and Barr, Avron. Russian Software Industry Interview Notes, unpublished. Stanford Computer Industry Project. 1996.

**Tessler, Shirley, Barr, Avron and Feigenbaum, Edward.** *The US Software Products Industry: Success Factors in an Era of Rapid Change.* **Stanford Computer Industry Project Research Brief. August 1994.**