Information Technology for Development 9 (2000) 77–90
IOS Press

Software engineering services for export and small developing economies

Han Reichgelt
Department of Mathematics and Computer Science, University of the West Indies, Mona, Kingston 7, Jamaica
Tel./Fax: +1 876 977 4470; E-mail: han@uwimona.edu.jm

Abstract. A number of authors and multi-national organizations have suggested that providing information services, and in particular software engineering and programming services, for export afford an important economic opportunity for poor countries. Throughout the world, developing countries have acted on this advice. This paper will argue that the opportunities for software engineering services in particular are limited, at least for *small* developing economies. The main argument is that software engineering and programming are labor-intensive activities and that small developing countries simply do not have the required resources to acquire or train a sufficient number of software engineers and programmers. Any development policy that blindly follows the tenet that small developing countries can improve their economic position through the provision of information services for export is therefore bound to fail. Hence, more sophisticated policies are called for. This paper will also examine a number of such policy options, including an innovative human resource development policy being developed in Jamaica.

Keywords: Information services for export, economic development policy, small developing countries, Jamaica

1. Introduction

Various authors have argued that the information revolution provides an important opportunity for developing countries. The argument has two sides. First, information technology can be used to address the various problems that developing countries face. For example, Talero and Gaudette [18] state that information technology can be used to

- Educate people and support lifelong learning;
- Make governments more efficient, accountable, and transparent;
- Increase effectiveness of economic reforms;
- Monitor and protect the environment;
- Reduce information and income inequalities;
- Overcome natural disadvantages;
- Promote small and medium enterprises.

The second part of the argument that the information revolution provides an important opportunity for developing countries is that it allows poor countries to diversify into information-intensive industries (e.g., [15,18]). Proponents of this proposition point to the fact that, with the exception of hardware manufacture, the establishment of an information services industry requires only modest amounts of start-up capital. Also, salaries in developing countries are generally lower than salaries in developed countries and with salaries being the main cost in software development, developing countries have an important advantage. Moreover, the demand for information services is growing rapidly. For example, the world software market was estimated to be around $225 billion in 1993 [14] and to be growing at between 15 and 18% annually [3]. Since entry into this market is cheap and developing countries have the advantage of lower salaries, poor countries should be able to take advantage of this opportunity, or so the argument goes.

Further support for this proposition comes from countries, such as Ireland and Israel, which now earn substantial amounts from information services for export. For example, according to Bord Trachtalia, Ireland's trade agency, Ireland exported software worth 419 million Irish Pounds (c. $260 million) in 1996 and is expected to increase this to 1 billion Irish Pounds (c. $625 million) by 2001. Dedrick, Kraemer and Goodman [5] discuss a number of other small countries that make substantial amounts of money out of the export of information services and computer hardware.

However, the jewel in the crown of the argument that poor countries can substantially increase their GDP by providing information services for export, is India. India is an extremely poor developing country that is estimated to earn around $1 billion from packaged software in 1996 (*The Economist*, 1996 05 25). It has been estimated that India now earns 10% of its GDP from information services for export, and that this percentage is growing.

These observations have persuaded a number of governments of developing countries to make the provision of information services for export a corner stone of their economic policies. Examples in the Caribbean are Barbados and Jamaica. Other examples are Bangladesh, Mauritius, Mexico, the Philippines and Sri Lanka.

There are of course many different types of information services. Included are data entry services, data conversion services, back office operations, telemarketing, the maintenance of so-called legacy systems, old computer programs that are still in use in many organizations, and the provision of software engineering and programming services. Correa [4], Heeks [8] and Vonortas and Safioles [20] discuss strategies that developing countries may adopt to establish a presence in the international software market.

While some governments have explicitly targeted the lower value added end (e.g., data entry and data conversion), most do not restrict themselves in this way. For example, the Government of Jamaica, in its recently published National Industrial Policy, explicitly states that the country should aim to provide software engineering and IT consulting services for export. In this paper, I will concentrate on the higher value added end of the IT services spectrum and in particular software engineering and programming services.

This paper will not take issue with the first part of the argument that the information revolution provides important opportunities for developing countries. Indeed, the increasing globalization of the world economy and the growing role of information and telecommunications technologies in all spheres of human activity, suggests that developing countries have no choice but to embrace these new technologies. However, the paper will take issue with the second part of the argument. In particular, the paper will argue that the opportunities for export in the higher-value added end of the IT services market, and especially programming and software engineering services, are rather limited, at least for small developing countries. Although the paper will often use Jamaica as an example, it is important to note that in most cases, Jamaica will merely be used as an example and that the arguments are applicable to all small developing countries.

The organization of the paper is as follows. In Section 2, I discuss the software engineering process. The main point will be that software engineering is an extremely labor intensive process. It follows that a necessary, but by no means sufficient, condition for the establishment of a vibrant software engineering sector is the availability of a large pool of highly trained individuals. In Section 3, I then use this to argue that the opportunities for entry into the programming and software engineering services market are limited for small developing countries with populations less than 10 to 15 million. In this section, I will also analyze the factors that led a number of small countries, most notably Ireland, Israel and Singapore, to become important players in the international market for IT services. In Section 4, I discuss four possibilities to overcome the limitations identified in Section 3. One of the options is the introduction of innovative human resource development policies such as that being adopted in Jamaica. Section 5 concludes.

2. Software engineering

In order to determine whether small developing countries are likely to be successful in their overall goal of increasing their GDP by making the provision of software engineering and programming services central in their economic development plans, it is necessary that we briefly examine the software engineering process.

There are different models that one can use to manage the software engineering process. Each model divides the software engineering process into a number of distinct phases. The most widely used model for managing the software engineering process is the so-called waterfall model [17] and we will restrict the discussion to this model. There are different versions of the waterfall model, but all versions roughly agree that the software engineering process goes through at least five phases. The first is the problem definition phase. The aim of this phase is to formulate as precisely as possible and in user terms the problem that an organization faces and that the software is expected to solve. The second phase is requirements analysis and results in a document describing the requirements that the software is to meet. The third phase is the specification phase in which one specifies what the software will look like and how it will meet its requirements. The fourth phase is the programming phase in which one turns the specification into program code. The final phase is maintenance in which one fixes bugs, errors that have been discovered in the software, and possibly increases its functionality.

Although there is a popular perception of a software engineer or a programmer as a loner, reality is different. Almost all software is developed by teams, ranging in size from 2 to several hundreds [17], and choosing the right team structure is one of the most important determinants of the success of a software development project. Moreover, in a by now somewhat dated survey of programmers working at IBM, McCue [12] found that they spent about 50% of their time interacting with other team members, and merely 30% working alone.

Moreover, software engineering in general and programming in particular is a time consuming process. Programmer productivity is known to be extremely hard to quantify [17]. In any estimation, a large number of factors need to be taken into account, such as aptitude of the individual programmer, complexity of the user interface, team experience and the use of modern design methodologies and programming languages. As a result, programmer productivity may be as low as 30 lines per programmer month, or as high as 600 lines per month [17]. Since programs of more than 100,000 lines are not uncommon, it will be clear that software development is an extremely time-consuming process. Assuming an average programmer productivity of about 300 lines per month, the development of a program of 100,000 lines will take roughly 28 programmer years. Clearly, the only way to speed up the delivery of the software is by having large teams of programmers working together on the same project, even if this is likely to decrease the individual programmer's productivity because of the need to spend time interacting with other team members.

It is also important to re-iterate that software engineering and programming do not merely involve writing new software. It also involves maintenance, fixing bugs and increasing the functionality of the software. However, like writing code, maintenance is also a labor-intensive process. For example, Lientz and Swanson [11] discovered that large organizations devoted about 50% of programmer time to maintenance. Boehm includes a function for estimating the cost of software maintenance in the well-known COCOMO model for estimating the cost of developing software [2]. In this model, the annual maintenance effort is a direct function of, *inter alia*, the time it took to develop the software in the first place, the expected number of changes, the availability of experienced support staff and the use of good programming practices for system development. However, assuming that one needs to change (or at least inspect) about 10% of the code in any one year (probably a conservative estimate), Boehm's model would predict that one requires between 1 and 2 programmer months for maintenance per programmer year used in the original development. Thus, a program that took about 10 programmer years to develop will require one full time programmer for every year that the company wishes to maintain it. While modern design and programming techniques, in particular object-oriented techniques, have made software maintenance more straightforward, perhaps decreasing maintenance costs by 50% or more, there is no denying that software maintenance remains a relatively time and labor intensive activity. It follows that any company that wishes to provide software engineering services must not only employ staff to develop software; it must also employ staff to maintain the software, thus significantly increasing the numbers of staff required.

The final point to make about software engineering is that most programmers and software engineers employed today are highly trained individuals. As a cursory glance at job advertisements in the information technology sector shows, almost all software firms fill their entry-level positions with people who have a University degree, or at least a significant post-secondary training. This is confirmed by the US Department of Commerce [19, Appendix to Chapter IV] which states that computer programmers generally require a Bachelor's degree, i.e., at least 4, but not more than 5 years of post-secondary training. Thus, a successful software house does not merely require a large number of staff; it requires a large number of highly trained staff.

Further, indirect, evidence for the importance of sheer numbers of staff for software engineering can be gathered from looking at the size of successful software engineering company. Thus, ORACLE, producers of probably the most successful database management system, employs more than 25,000 software engineers (source: *www.oracle.com/corporate/annual_report/html/ao.html*). Microsoft, producers of Windows and a host of application software packages, employs more than 20,000 technical staff (source: *www.microsoft.com/jobs/guide/emstats.html*). Thus, successful software companies employ large numbers of technical staff. Even organizations that are not directly involved in providing information services often employ large numbers of IT staff. Thus, in 1996, the Department of Commerce in the United States of America had approximately 2,600 IT employees, the US Navy about 13,000 and the Environmental Protection Agency roughly 1,300.

There is a final observation that indirectly points to the importance of large numbers of software engineers. Those countries that earn most from software employ large numbers of programmers. Thus, according to *The Economist* of 1996 05 25, the United States controls about 75% of the market for packaged software, estimated to be worth $100 billion in 1995. However, according to figures produced by the Bureau of Labor, the US employs around 2 million software engineers and programmers nation-wide. Germany and the United Kingdom, which, between them, control about 10% of the packaged software market, employ around 500,000 programmers each. India, with an income from packaged software of around $1 billion, employs around 100,000 programmers. Finally, Israel, which in 1993 was estimated to have earned $175 million in software exports, employs around 5,500 technical staff devoted exclusively to software engineering out of a total population of 40,000 IT professionals [1]. Thus, countries that earn substantially from software also employ a large number of software engineers.

The latter argument might seem circular. After all, the argument merely seems to state that in order to make large amounts of money from software services, a country must provide a lot of software engineering services and hence employ many software engineers. However, the point of the argument is slightly subtler. There seems to be a belief that successful software engineering companies can be small in terms of the number of employees. If this were indeed the case, then one would expect to be able to find countries that get a substantial amount of their income from software services but without employing large numbers of software engineers. However, such countries do not seem to exist, thus lending indirect support to the proposition that for a country to become successful at providing software services for the international market, it must employ a large number of programmers.

3. The establishment of a software engineering industry in small countries

As argued in the previous section, software engineering is a labor-intensive process, and one that requires highly trained people to boot. This entails that a country that wishes to make the provision of software engineering and programming services a corner stone of its economic development policy, must be able to either train a large number of software engineers itself, or must be able to attract programmers from abroad. This factor seems to escape many authors who argue for the proposition that poor countries can improve their economies through the provision of information services for export. In any discussion of obstacles that such a country has to overcome, authors concentrate on such issues as the need to have a modern telecommunications infrastructure, or to have investment incentives, rather than the crucial human resource component. I do not wish to underestimate the importance of these other factors. However, a large enough group pool of programmers is a necessary condition for a country to meet if it wishes to establish itself in the international market as a provider of software engineering and programming services. It is my contention that, currently, small developing countries do not satisfy this condition and therefore cannot take full advantage of the opportunities provided by the heavy global demand for software and programming services.

The main argument for this position is based on the observation that developing countries tend to have a much lower percentage of the eligible population enrolled in tertiary education than developed countries. Thus, according to figures of the OECD, as quoted in *The Economist* of 1997 10 04, in 1994 the USA has a graduation rate of about 43%, where graduation rate is defined as the ratio of University graduates to population at theoretical age of graduation. Graduation rates of some other developed countries are 35% in Canada and Britain, 30% in New Zealand and Norway, about 25% in Denmark, Ireland and Japan, and between 10 and 15% in Germany, Italy, Greece, Switzerland and Austria. According to the same figures, Turkey has a graduation rate of just about 8%. In Jamaica, the graduation rate is under 5%, and in this respect Jamaica is not untypical of small developing countries.

It is also interesting to look at the number of IT graduates in various countries. A World Bank report (1993) provides information about the number of University graduates with a degree in Computer Science, or a Computer Science related subject such as Computer Engineering or Electronics, for a number of countries. For example, according to this report, South Korea produces 1,100 such graduates per million per annum, the United States 1,000 and Japan 830. India is reported to produce 140 graduates of this type per million in the population per annum. According to figures provided by the two universities in Jamaica (the University of Technology, Jamaica, and the University of the West Indies, Mona, Jamaica), the number of University graduates in Computer Science or a Computer Science related subject per million in the population in Jamaica is approximately 60 to 80. However, both universities expect a doubling of the number of graduates in the early 21st century, hence bringing the number of graduates per million in the population per annum to the level obtained in India.

However, such relative figures hide the full story. For example, a relative figure of 140 IT graduates per million in the population per annum for India means an annual production of roughly 135,000 new IT graduates. For Jamaica, the annual output of new IT graduates would merely be 350.

In Section 2, we observed that large numbers of IT professionals seem to be essential to the establishment of a successful software engineering industry. It is therefore unlikely that a figure of 350 new graduates per annum is sufficient. Given the importance of large numbers of IT professionals and the seeming inability of small developing countries to train such professionals in large numbers, one is forced to conclude that it is unlikely that small developing countries can establish themselves as major players in the international software and programming services market.

The obvious way to counter the above argument is to point to small countries which have been successful in establishing themselves as major players in the international software market. Examples are Ireland, Israel and Singapore. However, in all three cases, there were exceptional circumstances.

If we start with Ireland, then the first thing to note is that Ireland's initial entrance into the Information Services sector was through the provision of back office services, primarily for US based companies. The main factor in Ireland's success was a cheap and literate work force. Once Ireland had established itself in this section, expansion into other Information Services was certainly helped by Ireland's membership of the European Union. Thus, although small itself, with a population of about 3.6 million, Ireland provided access to a market many times that size and larger than the North American market. Moreover, Ireland received significant cash inflows from the European Union. According to figures from the European Commission and the Centre for European Reform, quoted in *The Economist* of 1999 03 20, Ireland received 2.8 Billion ECU (US$ 3 billion) in inflows from the European Union in 1997 equivalent to ECU 800 or US$ 880 per person. While I do not wish to argue that such inflows are necessary or sufficient to set up a sophisticated educational system, it is clear that they clearly are helpful in creating such a system.

Interestingly, Ireland recently identified a shortage of skilled IT professionals as a major factor limiting its further growth in this area. Currently, Ireland is producing approximately 800 Computer Science graduates per year from its Universities. It is planning to increase this to around 2000 by the year 2001, excluding the large number of graduates who major from diploma programs in Information Technology offered by a variety of other training and educational institutions. Moreover, the Irish Government is currently pursuing policies aimed at attracting trained professionals both from the Irish community overseas and from other European countries.

Israel's situation is also somewhat unusual. First, it is important to realize that Israel is primarily known for communications software. Moreover, the vast majority of the software engineers working in Israel have spent a significant period in the Israeli army and it will be clear that secure communications are essential for a modern army. Thus, it seems likely that one of the reasons that Israel is a world leader in the area of communications software is that many of its software engineers have had the opportunities to hone their skills in this area in the army.

A second reason why Israel has been able to become an important player in the software market is that, as in the case of Ireland, Israel has an excellent tertiary educational system and produces between 500 and 700 computer science graduates from its various universities annually [22]. Moreover, following the fall of the Iron Curtain in 1989, Israel has been able to absorb some 400,000 immigrants from the former Soviet Union, about 25% of whom held degrees in Engineering. As a result, Israel currently has 135 engineers and technicians per 10,000 population, compared to, for example, 18 in the United States (*The Economist*, 1999 02 20). Finally, Israel, like Ireland, is a substantial beneficiary of net official aid. According to the OECD, Israel received roughly US$ 1.2 billion in net official aid in 1997, primarily from the United States, corresponding to roughly 1.2% of its GDP (figures quoted in *The Economist*, 1999 03 13). Not many other small countries, including many which are significantly poorer than Israel, have the same advantages as Israel has in this respect.

Finally, Singapore's planned entrance into the information services market has been well documented (e.g., [6,9,10,16,21]). Information technology played an important role in Singapore's transition from a relatively poor developing country to a newly industrialized country. The role of Singapore's government through the National Computer Board (NCB) in establishing the country as an important provider of information technology services to the world has also been extensively discussed. Moreover, Singapore earns significant amount of money from the IT industry. For example, in 1994, total IT sales to end users from IT vendors were estimated to be worth US$ 3 billion, of which roughly half came from export sales (figures from Singapore's National Computer Board, quoted in [21]). However, it is important to realize that 72% of these sales were derived from hardware and a mere 12% from software and 16% from IT services. If we make the assumption that 12% of the export earnings came from software as well, then this means that in 1994 Singapore earned around US$ 170 million from software export, or less than two percent of its GDP. This figure is consistent with figures from the Singapore Trade Development Board, quoted in [4], which put the sales of software for export at $138.1 million in 1993. In other words, while there is no doubt that Singapore is a significant player in the international IT industry its earnings from the export of software are relatively low.

It is also interesting to note that one of the obstacles to Singapore being able to implement its admittedly ambitious plans for the further development of its IT sector has been identified as a lack of trained human resources. For example, Gurnaxani et al. [6] quote Government sources as estimating that Singapore is likely to experience a shortfall of more than 1,000 IT professionals in 1990 and expecting this number to increase in the future. The Singaporean government is trying to solve this problem by attempting to attract IT professionals from other Asian countries and by encouraging the use of telecommuting. For example, 25% of the 19,000 students at Singapore's Nanyang Technological University, primarily an engineering university, are from abroad and all are encouraged to remain in Singapore after graduation (Prof. T. Chan, personal communication, 1999). In a similar vein, 40% of the 2,500 students in Computer Science at the National University of Singapore are not from Singapore (Prof. I. Png, personal communication, 1999). Despite all this, the supply of IT professionals is still expected to fall far short of the demand.

4. Possible solutions

It will be clear from Sections 2 and 3, that a country can only hope to earn significant amounts from the export of software engineering and programming services if it can produce or attract a large number of software engineers and programmers. If a country does not have the required human resources, it has four, not necessarily exclusive, options open to it. First, it can, of course, abandon a development policy based on the assumption that it can obtain a significant increase in its income through the provision of software engineering and programming services. Second, it can try to import IT skills. Third, it can set itself more moderate goals and try to focus on niche markets. Fourth, it can pursue policies that would enable it to significantly increase the number of locally trained IT staff itself. We will discuss these various options in detail.

4.1. Option 1: Abandon a development policy based on providing software engineering and programming services for export

The first option open to small developing countries is clearly to give up on any policies aimed at increasing the country's income from the provision of software engineering and programming services for the international market. One can argue that the reasons put forward for adopting such policies in the first place were extremely weak. The argument seems to involve the following steps. First, there is a large and growing international market for such services. Second, entry into the market is easy, or at least cheap. Third, developing countries can provide such services cheaper than other countries as their salaries, the main cost in providing such services, are considerably lower. However, the analysis put forward in this paper clearly indicates that there is an important additional condition that tends to be overlooked, namely the availability of a sufficient number of software engineers and programmers. If a country has such resources available, then entry into the market may be cheap for individual firms. However, there is an important hidden cost, namely the cost of training. Setting up an education system capable of producing top-class software engineers and programmers requires significant resources, typically borne by national governments. Entry costs for an economy as a whole may therefore be much higher than traditionally stated and may actually be beyond the reach of developing countries, the resources of whose national governments are very limited.

This is not to argue that developing countries should simply abandon information technology. The argument put forward is that small developing countries are unlikely to be able to significantly increase their national incomes from the provision of software engineering and programming services to the international market. There is no doubt that, no matter what economic activities such countries choose to focus on, information technology has an important role to play in making these activities more efficient or in marketing the goods or services provided through these activities. It might therefore be argued that small developing countries should use whatever human resources they can secure in the general area of IT on those areas in which they have a competitive advantage. One could argue that Singapore did exactly this when it expanded considerable efforts in establishing TradeNet, software to make its transhipment operations more competitive, even though it already was a significant player in the international transhipment market [10].

However, given the enormous size of the market for software engineering and programming services, small developing countries are unlikely to simply want to give up the policy of increasing their income through the provision of software engineering and programming services to the international market. I will therefore investigate the three other options.

4.2. Option 2: Import software engineering skills

This option is unlikely to be successful in small developing countries. First, there is a severe shortage of IT skills worldwide. In Section 3, we pointed out that both Ireland and Singapore have explicitly identified a shortage of IT skills as one of the major obstacles for them to realize their plans for the IT sector. The United States already has a severe shortage of IT professionals and the United States Congress has increased the number of visas for high-skilled IT professionals from 65,000 to 115,000 for the fiscal years 1999 and 2000. The cap for 1999 was reached in June [19]. It follows that a developing country that wishes to attract trained software engineers and programmers from elsewhere will have to compete for scarce skills with developed countries.

While salaries are not the only element in a compensation package for software engineers, they clearly are an important aspect. Thus, if a country wishes to compete for software engineers, then it will have to offer salaries that are attractive by international standards. However, one of the reasons put forward for believing that developing countries can increase their income by providing software engineering services for export are their lower salaries. We therefore see a vicious circle emerging. If a developing country is to attract software engineers, it will have to offer competitive salaries, and hence erode one of the competitive advantages that it might have had in the international software market, namely that of lower salaries.

It is interesting to observe that a number of IT organizations in Jamaica have indeed recruited programmers from abroad, notably from India. This initiative has met with mixed success. While certain firms have been able to attract excellent staff from abroad, others have struggled. Moreover, even those firms who have been able to attract good staff have had difficulties retaining them. Such staff have often left Jamaica after a short while for financially greener pastures elsewhere, in particular the United States. Judging from the Jamaican experience, the policy of recruiting IT skills from elsewhere in pursuit of the goal of establishing oneself as a major player in the international software market has met with very limited success. It is likely that the factors that made this policy problematic in Jamaica will hold in other developing nations as well.

Incidentally, the lure of higher salaries elsewhere poses yet another problem for developing countries, namely the migration of locally trained skilled professionals. A conservative estimate is that at least 10% of the graduates majoring in Computer Science from the Jamaican campus of the University of the West Indies migrate, typically to the Canada, the United Kingdom, or the United States. Heeks [7] reports that India loses about 15% of its software workers annually. As the US Department of Commerce puts it:

> Information Technology itself should be an important development tool allowing LDCs (less developed countries, HR) to move forward faster. However, the demand of industrially advanced countries makes it difficult for much of the developing world to keep their limited pool of IT workers at home. ([19, p. 44].)

4.3. Option 3: Identification of niche markets

If small developing countries cannot either attract or produce the personnel to establish themselves as important players in the international software market, then another option might be to consider other policy options and to set themselves more moderate goals. An obvious more moderate goal would be to identify niche markets and concentrate their efforts on establishing themselves in these niche markets. Some developing countries have been able to establish themselves as important providers of certain goods or services, and they might be well advised to use their limited human resources in the area of IT to provide software for the areas in which they have a position of strength. Again taking Jamaica as our example, one area in which Jamaica has an internationally competitive position is all-inclusive tourism and it might therefore be possible to provide software for this particular market.

However, focussing on a particular area of application is not the only possible implementation of the strategy of trying to establish oneself in a niche market. In Section 2, we briefly described the waterfall model of software development. The waterfall model, like any other life cycle model for software development, divides the software development process into different phases. There is some disagreement among software engineers as to how to exactly schedule these phases. For example, some argue that the earlier phases have to be totally completed before a software development project can move into its next phase. Others argue that it should be possible to return to an earlier phase even after the project has moved into a later phase. Yet others argue that phases can temporally overlap. However, there is a general agreement that it is in principle possible for the different phases to be performed by different personnel and even at different places and time. Indeed, many software developers in the United States and Europe outsource their actual programming, i.e., the translation of the specifications developed earlier in the project into program code, to companies in India, while they perform all the other phases in their software development projects themselves in the US or Europe. Thus, an obvious alternative implementation of the strategy of establishing oneself in a niche market is for a country to focus on particular phases in software development, and India can indeed be analyzed as having done so. Although India is expanding its services into the earlier phases of the software development process, initially it explicitly restricted itself to programming services.

What makes this latter strategy particularly attractive for smaller countries is that not all phases in the software development process are equally labor intensive. In Section 2, we argued that software development is a labor intensive activity and we used this observation in Section 3 to argue that it is unrealistic for small developing countries to believe that they can establish themselves as major providers of software for the export market. However, a closer look at the argument will reveal that its main focus is on programming. Indeed, the earlier stages of software development are often more difficult than the later stages and certainly require more highly qualified people, who understand not only the technology but also the business that the information system is being developed for. Moreover, these earlier stages are also more lucrative. Finally, the earlier stages require fewer people. Indeed, many consulting firms concentrate their efforts on the earlier stages. Here then lies an opportunity for small developing countries. If they can establish themselves in a position of strength in providing consulting services of the type, then they may be able to boost their GDPs substantially from the IT services, despite their shortage of human resources. Obviously, the problem with this strategy is how to establish oneself in the international market as a provider of consulting services, especially in the light of Heeks' observation that many companies in developed countries still have fear, uncertainty and doubts about contractors in developing countries and their business environment.

4.4. Option 4: Strengthening tertiary education

Clearly, a final policy option for a small developing country wishing to establish itself as a provider of programming services is to promulgate policies to try to produce such skills itself through various educational institutions and programs. As stated in Section 2, software engineering and programming are activities that require highly trained individuals. Entry-level positions in the IT industry are generally filled by recent university graduates, or personnel with at least significant post-secondary training. Developing countries could therefore try to overcome the shortage of skilled personnel through the establishment of good training institutions.

It is clearly not beyond the capabilities of developing countries to establish good post-secondary training programs or universities. For example, one can point to a number of highly regarded universities in developing countries. Moreover, India was able to establish a position of strength for itself in the programming market through the establishment of a wide range of good training institutions.

However, merely establishing good post-secondary training institutions or universities is insufficient; one also needs to ensure that a sufficient number of students attend these institutions. The argument that small developing countries lack the required human resources to establish themselves as major players in the international software market is a quantitative one, not a qualitative one. The argument is not that such countries are unable to produce good software engineers; the argument is that their small populations make it unlikely that they are able to produce such people in sufficient quantities. As can be deduced from the arguments put forward in Section 3, ensuring that a small developing country produces a sufficient number of software engineers is highly problematic. Many developing countries, whether small or large, have a low proportion of the eligible age group enroll in their post-secondary and tertiary training institutions. I also argued that such relative figures are even more worrisome for countries with small populations. A low enrollment rate in a country with a large population, such as India, still results in a large number of trained graduates in absolute numbers. The arithmetic clearly works against small countries.

Moreover, educational policies aimed at ensuring that a larger number of a country's population receives significant post-secondary or tertiary education are likely to be very expensive. Such policies should ensure that a larger number of students stay within full-time education, including secondary education, at an age where economic conditions often force them to take on full-time work. Moreover, if one accepts the assessment by the US Bureau of Labor that computer programmers require at least a Bachelor's degree, then prospective programmers require at least four years of full-time post-secondary education. So, students should stay in full-time education for a relatively long period of time and hence forego income for themselves and their families.

The policies aimed at keeping students in full-time education are therefore only likely to be successful if they somehow can make it attractive for them to do so from a financial perspective as well. One way of achieving this might be the establishment of tax rebates for parents who keep their children in full-time education. However, it is not clear to what extent it would achieve its objective. It seems reasonable to expect that the main beneficiaries of such tax breaks are parents who would have kept their children in full-time education anyway, rather than encouraging parents who normally would have taken their children out of full-time education. A more direct policy, such as the provision of grants and scholarships, would seem more likely to yield the desired results.

Unfortunately, it is not clear whether poor countries can indeed afford such policies. One again seems to be caught in a vicious circle. In order to improve the economic situation of the country through the provision of software services for export, poor countries would seem to have to pursue educational policies which, given the states of their economies, they are unlikely to be able to afford.

In this context, it is also interesting to again briefly look at the Singaporean situation. Singapore's success in the international IT industry was partly possible through the establishment of some excellent educational institutions and policies. Mingat [13] compares Singapore's educational policies with those of other high performing Asian countries. However, in Section 3, we also saw that there is concern in Singapore about a lack of IT skills. Moreover, K.C. Lee, a high-ranking official at Singapore's National Computer Board, as quoted in Gurnaxani et al., expressed the fear that any increase in the number of students in IT related training programs may come at the cost of other professional skills that Singapore needs as well. One can infer that Singapore's small population may indeed be one of the factors that may prevent it, or at least present it with great difficulties in, achieving the goals it has set itself.

4.5. A Jamaican option: Short-term training programs

Fortunately, there may be yet another option worthy of consideration. As stated, there is a general consensus that programmers require a Bachelor's degree. However, it is not clear what the basis of this consensus is. It is not too hard to justify the position that early phases in the software development process (e.g., problem definition and requirements specification) indeed require training up to that level and probably beyond. After all, in order to produce a good problem definition document, a consultant requires an in-depth understanding of the business that the information system is to be developed for and of the current computer and communications technology. Moreover, he or she must possess excellent analytical and inter-personal skills. Without it, he or she would be unable to fully grasp the problem facing the users requiring the new system. It seems obvious that such knowledge and skills can only be acquired after a significant amount of post-secondary training.

However, an analysis of the programming process makes it more difficult to see why a programmer needs a Bachelor's degree. After all, a programmer merely has to translate the detailed specification and design that result from earlier phases in the software development process into program code. Clearly, this requires a certain amount of intelligence and additional post-secondary training. It is, however, unclear whether this training should be to a Bachelor's level. The skills required for programming seem similar to the skills required to solve so-called word problems in mathematics, where one is given a description of a problem in a natural language, asked to translate it into a set of equations and to solve the resulting equations. Children as young as 10 years old are taught such skills and one certainly expect a child with a good secondary education to have sufficient language and numeracy skills to be able to perform such tasks. Two factors make programming probably more difficult than solving word problems. The first is the complexity of the language the problem is to be translated into. The language of mathematics is very simple and easy to learn; programming languages are large and complex. The second is the size of the problem to be translated into the formal language. Word problems are typically 2 to 3 sentences long; specifications for computer programs certainly are much longer than that. However, it seems unlikely that these complicating factors are sufficient to justify an additional four years of training. It might therefore be possible to design special training programs of a much shorter duration to train entry level programmers.

The Caribbean Institute of Technology (CIT) in Jamaica is based on this assumption. CIT was set up as a joint venture between three universities (one from the West Indies, one from the United Kingdom and one from the United States), a software house from the United States and the Government of Jamaica's training agency. It offers a nine-month full-time training program for candidates with at least four years of secondary education and good achievements in English and Mathematics, and aims to produce entry level programmers. Currently, CIT only aims to produce C++ programmers but there are plans to broaden the course offerings so that it can also start producing programmers for other programming languages. CIT started operations early in 1999 and it is at this stage premature to say how successful its programs will be. The first batch of 45 programmers graduated at the end of 1999. They have all been employed, and initial indications are that they fully meet the expectation of their employer. However, since they have been employed by the software house involved in the establishment of CIT, one would clearly need additional evidence to determine the success of CIT's training programs.

However, assuming that CIT does indeed meet its objective to produce good entry level programmers, then it may provide a model that other small developing countries might wish to consider. Nine-month training programs are clearly considerably cheaper to fund, either by governments or by students enrolled in them, than four-year programs. It might therefore be possible to use this avenue to create the human resource base which is so critical for the establishment of successful software engineering and programming industry.

5. Conclusion

A number of small developing countries have recently followed the advice from a host of international agencies and consultants and made the provision of information services to the international market an important element of their economic policies. This paper argues that, when it comes to providing software engineering and programming services for export, such policies require further refinements. The argument is based on the observation that software engineering and programming are activities which require larger numbers of highly trained human resources than such countries are likely to be able to secure at the moment. Clearly, if the analysis put forward in this paper is correct then the governments of small developing countries would be well advised to reconsider economic development policies which rely heavily on the provision of IT services for export. The paper offered a number of alternatives, such as a clear identification of niche markets or the establishment of additional training programs. However, a policy which does not address the critical human resource factor and the formidable obstacles that result from the lack of sufficient number of trained software engineers and programmers is unlikely to be successful. The Caribbean Institute of Technology (CIT), set up in Jamaica, may provide one such policy option. CIT provides short-term training programs aimed at preparing school leavers for entry-level programmers. Institutions like CIT are clearly considerably cheaper to set up than the tertiary level educational institutions that are traditionally used to produce entry-level programmers, and may provide small developing countries with the resources to produce the numbers of programmers needed for success in the export market for software engineering and programming services.

Acknowledgements

While the overall responsibility for the views expressed lies exclusively with the author, a number of people have made valuable contributions to the ideas presented here. They include Professor Norman Girvan and Professor Gordon Shirley of the University of the West Indies, as well as my colleagues and students in the Department of Mathematics and Computer Science, notably Cedric Harold and Lisa Hanna. I am also grateful to Dr David Algeo of Trinity College, Dublin, for some of the data on Ireland.

References

[1] G. Ariav and S. Goodman, Israel: Of sword and software plowshares, *Communications of the ACM* **37**(6) (1994), 17–21.
[2] B. Boehm, *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
[3] C. Correa, Strategies for both software development and export in Latin American countries, Invited talk at EMPRETEC's Technical Workshop on Expert Development of Computer Software and Services, Montevideo, Uruguay, 1993.
[4] C. Correa, Strategies for software exports from developing countries, *World Development* **24** (1996), 171–182.
[5] J. Dedrick, S. Goodman and K. Kraemer, Little engines that could: Computing in small energetic countries, *Communications of the ACM* **38**(5) (1995), 21–26.
[6] V. Gurnaxani, K. Kraemer, J. King, S. Jarman, J. Dedrick, K. Raman and C. Yap, Government as the driving force toward the information society: National computer policy in Singapore, *The Information Sciences* **7** (1990), 155–185.
[7] R. Heeks, *India's Software Industry*, Sage Publications, New Delhi, 1996.
[8] R. Heeks, Software strategies in developing countries, *Communications of the ACM* **42**(6) (1999), 15–20.
[9] W.S. Hon, Exploiting Information Technology, *World Development* **20** (1992), 1817–1828.
[10] J. King and B. Konsynski, Singapore TradeNet: A tale of one city (Harvard Business School case) 1990, pp. 191–199.
[11] B. Lientz and E. Swanson, *Software Maintenance Management*, Addison-Wesley, Reading, MA, 1980.
[12] G. McCue, IBM's Santa Teresa laboratory – architectural design for program development, *IBM Systems Journal* **17** (1978), 4–25.

[13] A. Mingat, The strategy used by high-performing Asian economies in education: Some lessons for developing countries, *World Development* **26** (1998), 695–715.
[14] R. Schware, Software industry entry strategies for developing countries: A "walking on two legs" proposition, *World Development* **20** (1992), 143–164.
[15] R. Schware and S. Hume, *Prospects for Information Service Exports from the English-Speaking Caribbean*, The World Bank, 1996.
[16] N. Siong and C. Soh, IT2000: Planning for a new national infrastructure, in: *Exploiting Information Technology for Business Competitiveness: Cases and Insights from Singapore-based Organizations*, N. Siong, ed., Addison-Wesley, Reading, MA, 1996.
[17] I. Summerville, *Software Engineering*, 3rd edn, Addison-Wesley, Wokingham, England, 1989.
[18] E. Talero and Ph. Gaudette, *Harnessing Information for Development: A proposal for a World Bank Group Strategy*, The World Bank, 1996.
[19] US Department of Commerce, *The Emerging Digital Economy II*, 1999.
[20] N. Vonortas and S. Safioleas, Strategic alliances in information technology and developing country firms: Recent evidence, *World Development* **25** (1997), 657–680.
[21] P.-K. Wong, Implementing the NII vision: Singapore's experience and future challenges, *Information Infrastructure and Policy* **5** (1996), 96–117.
[22] D.M. Zimmerman, *Information Technology in Israel*, http://gurukul.ucc.american.edu/mogit/dz9586a/humapage.html (1995).

About the author

Han Reichgelt received his first degrees in Philosophy and Psychology from the University of Nijmegen in the Netherlands and his PhD in Cognitive Science from the University of Edinburgh in Scotland. After brief stints as a research fellow in Artificial Intelligence in Edinburgh and lecturer in Cognitive Science at the University of Nottingham, England, he took up the chair in Computer Science at the Mona campus of the University of the West Indies in 1991. His research interests include Artificial Intelligence and Expert Systems, Data Mining and Information Systems Policy, both at the firm and the national level. He is married and has two children.