OCCASION

This publication has been made available to the public on the occasion of the 50th anniversary of the United Nations Industrial Development Organisation.

DISCLAIMER

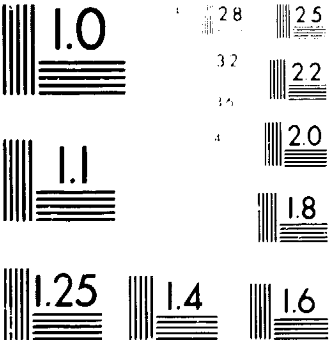
FAIR USE POLICY

CONTACT

Please contact publications@unido.org for further information concerning UNIDO publications.

For more information about UNIDO, please visit us at www.unido.org

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS

STANDARD REFERENCE MATERIAL 1010a

(ANSI and ISO TEST CHART No. 2)

Distr.
LIMITED

UNIDO/IS.439
10 February 1984

ENGLISH

UNITED NATIONS
INDUSTRIAL DEVELOPMENT ORGANIZATION

13385

GUIDELINES FOR SOFTWARE DEVELOPMENT IN DEVELOPING COUNTRIES*

by

R. Narasimhan**
UNIDO Consultant

* The views expressed in this paper are those of the author and do not necessarily reflect the views of the secretariat of UNIDO.
Mention of firm names and commercial products does not imply the endorsement of UNIDO.
This document has been reproduced without formal editing.

** Director, National Centre for Software Development and Computing Techniques, Tata Institute of Fundamental Research, Homi Bhabha Road, Bombay-400 005, India

V.84-81527

CONTENTS

1. INTRODUCTION

1.1 Software as Technology

In order to work out guidelines for software production in developing countries for achieving self-reliance, it is important to have, to begin with, a clear idea of the technology of software production. The software production process -- i.e., the process of producing software products -- is intrinsically delimited by the nature of software technology. When we talk about developing countries achieving self-reliance in software production, ultimately what is intended is that these countries should become self-reliant in the deployment of software technology. That is, they should become fully conversant with the production processes determined by this technology so that specific software products to meet identified end-uses (according to the needs and priorities of each country) could be produced by these countries and deployed for use appropriately. At the basic level we are thus discussing acquisition of know-how relating to a technology so that a country could exercise autonomous control over its deployment and usage to suit its assessed needs.

The problems involved in accomplishing this are neither simple, nor straightforward, even in the case of established, well-understood hardware technologies. The problems become very much more complex and difficult when we consider advanced technologies like biotechnology, microelectronics technology, nuclear technology, space technology, and so on. In acquiring meaningful know-how in these technologies, the technological preparedness of a country becomes a determining factor. Developing countries span a very wide spectrum of technological preparedness and competence. Clearly, not all advanced technologies can be fully autonomously controlled by all developing countries for their use. Nevertheless, it is being increasingly realized by the international community that, even if fully autonomous control over the production and use of advanced technologies cannot be exercised by a developing country, it would stand to lose much by not preparing itself to come to grips with such technologies.

An International Forum on Technological Advances and Development organized by UNIDO at Tbilisi, recommended that developing countries, individually and collectively, should examine their current states of technological capabilities and take steps to reorient their institutions and structures as necessary and appropriate to respond to technological changes in accordance with their own objectives and conditions. In addressing the question of differences in levels of competence, the Forum identified three possible entry levels to begin the mastery of a new technology as follows:

> <u>Minimum level</u>: awareness, continuous monitoring, critical and relevant technological intelligence; identification of relevance, ability to assess, select, negotiate and utilize technology; autonomous decision-making.

Medium level: the above and in addition ability to adapt or generate technology.

High level: the above as well as capacity for commercialization, design, manufacture of equipment, and participation in competitive international markets.

These recommendations have been followed up in a subsequent UNIDO-organized Workshop held at Dubrovnik. A conceptual schematization of the technology absorption process in terms of actors, activities, and their interconnections was articulated at this Workshop. And on this basis institutional and structural responses essential for technology absorption were identified in great detail [UNIDO 1983a].

All these analyses & recommendations, as we shall see in this report, apply mutatis mutandis to software technology and its absorption and use also. Software technology, however, differs from hardware-related technologies in some essential aspects. It is important to be aware of these differences in order to be able to appreciate the issues that need to be faced and solved in acquiring mastery over software technology.

1.2 Software as Engineering

A typical engineering industry is concerned with production in large quantities of a specific item. This is true whether the concerned end-item is a discrete hardware (automobile, computer), or a bulk material (chemicals). Typically, a factory delimits a production environment within which this quantity production takes place. To realize copies of the identified end-product (or bulk quantities of it, in the case of chemicals, etc.), various input materials are taken through a complex network of physical processes. Specific classes of tools (machine tools, chemical processing equipment, and/or people with hand-tools) are deployed appropriately to carry out the processing at the various stages.

The technology transfer in such a situation can be summed up as consisting of transfer of know-how and information concerning the following facets:

1. Factory planning
2. Setting up production facilities, providing tools, plant machinery and equipment
3. Technical documentation
4. Manufacturing processes
5. Quality assurance & inspection methods
6. Manpower training.

At a later stage one may want to consider the transfer of know-how to create new engineered products of the same type or related types (a new automobile, new computer designs, new chemicals, etc.), and establish the appropriate variant of the original production process so that quantity production of this new item can be realized. The transferred know-how could, in the best of cases, then, be expected to go from

Local Manufacture ⟶ Local Engineering
⟶ Local Development
or, in other words,
Manufacturing know-how ⟶ Engineering know-how
⟶ Design know-how.

In the case of a software product, in contrast to the conventional engineering products as discussed above, once a software package has been designed, coded, tested, and found to work, making copies of it is a trivial process, if the copy has to run on an identical hardware. A non-trivial conversion process has to be gone through if the same software routine has to be made to run on a different hardware. Since hardware keeps changing all the time, and customers like to opt for improved hardware as these become available (through technological innovation), already running software routines need to be converted to run on new hardware. The conversion market is, thus, an ever-present and conceivably lucrative market for a software production agency.

But the legitimate objective of software production cannot be either copying or converting, but creating ab initio the first version of a software package. Since a software package is produced only once, a production facility set up for software production has more the character of a design centre than a factory. This design-intensive character of software technology makes a software production facility resemble more closely a microelectronics fabrication facility rather than traditional engineering factories. This is because, in both these cases, a production facility, once established, is much less tied to the production of a particular engineered item than is the case with conventional engineering factories. Hence, to be able to exploit the full potentials of a production facility in these two cases, design know-how to design new products is an essential requirement.

The early efforts at software generation were really small-scale R&D efforts. Software packages were created, using tricks, clever strategems, and ad hoc techniques, by clever individuals or small teams of highly knowledgeable persons. This methodology was acceptable so long as the software created was used by the designers themselves or by an equally well-informed in-group. But as the complexity of the software products increased, larger teams of persons had to be involved in creating them. Also these products had to be increasingly created for wide distribution for use by a variety of persons with little or no knowledge of the design process involved. In these circumstances, the earlier informal, ad hoc, production techniques could no longer work. These had to be replaced by more formal and systematized design, coding, and testing processes. Precise and detailed documentation assumed great importance both at the design and coding stage and, more importantly, for the continued maintenance of the packages. Also, when a production team was concerned with designing and producing a wide variety of software products, the problem of improving the productivity of the personnel began to assume increasing importance. The incorporation of production tools (their design and their appropriate deployment) became a necessity [Kernighan & Plauger 1976, Reifer & Trattner 1977]. The software production process thus increasingly began to assume the character of an organized engineering industry. The software industry as it currently flourishes in the advanced countries, thus, has many of the ramifications of more conventional, established industries. Nevertheless its basic character remains quite distinct from that of a factory-based manufacturing activity. This is essentially due to the intrinsic design-level orientation of the activity.

Since software technology and engineering are still going through rapid developments even in the advanced countries, it should be useful to acquire some appreciation of the historical trends in this development. This would enable one to understand the kinds of pressures that have given shape to the software activities as we find them in the developed countries to-day. The products that are being created and used in these countries, and the structures that have evolved to create these products, may not all be immediately relevant to developing countries. We must have a clear perception of the essentials and the accidentals. The future trends in the developed countries are of some consequence to developing countries in the context of export and trade. Hence, it is of importance to understand how things came to be the way they are in the software sector in developed countries and where things are headed in the immediate and foreseeable future.

1.3 The Needs of Developing Countries

As we discussed earlier, the central issue is one of determining how best the developing countries can master the production know-how in software technology to be able to deploy it to meet their real needs. Our analysis must, thus, start with ascertaining what their real needs are. Broadly the needs could be grouped into four categories: (1) applications that relate to socio-economic programmes addressing the basic needs (primary health; literacy; rural employment; housing, transport, water, etc.;) and the agricultural sector; (2) applications in the government sector; (3) applications that relate to increasing productivity in the industrial sector (goods as well as services); and (4) applications relating to export and the international market in the software area. In each of these categories we can then determine the software products and services that are of central importance. Keeping in mind the picture in the developed countries, we can then analyse what are appropriate structures and methodologies for realizing these products and services for each of the four categories of applications.

We shall see that these four categories of applications require software support of two different kinds: (1) software products (packages); (2) software-supported systems. In the first case, if the needed products are available, one could think in terms of obtaining them for immediate use -- through purchase or lease. Modifications would still be needed to existing products before they become practically usable [Datamation 1980]. But, in general, this is the simpler of the two cases to cope with. Software-supported systems, on the other hand, almost always have to be tailor-made to suit the specific end-use environments. Even if similar systems are in use elsewhere, transporting them, modifying them, and fitting them to match local needs, may not be easy. In fact, in many cases, creating systems <u>ab initio</u> taking into account, in the design stage, local specificities may be a preferred solution. The structures and expertise needed to create such systems may have to be dealt with on a case by case basis. Also transfer of know-how in these system design areas is likely to be less straightforward. But precisely these application areas are the ones of great immediate importance to developing countries. It is in meeting the needs in these application areas that available software production models in the developed countries are likely to be of very little relevance: (see Sec. 3.2.1 for illustrative examples).

Existing models in the developed countries are likely to be more valuable and applicable to application categories (3) and (4) listed earlier. But, even in these cases, the local conditions and contexts may require the creation and use of new structures. Industrial production establishments in the developing countries -- even in the more advanced ones -- seldom have the level of informatics-awareness that is usually taken for granted in such establishments in the developed countries. Information processing practices, and software packages created to implement such practices in the developed countries cannot, therefore, be transported to developing countries and made to function effectively in a straightforward way. Software practices, such as distributed-processing and word-processing, assume new dimensions in countries where the telecommunication infrastructure is undeveloped, or where the local script and mode of writing differ radically from those in European countries.

Creating structures and transferring methodologies for the software production process are predicated on the availability of manpower with the requisite backgrounds in software engineering, knowledge of application areas, expertise in consultancy and, not the least, marketing skills. The methodologies for manpower creation, again, cannot be directly transferred from the developed to the developing countries. Development in developed countries has been the result of a historical, evolutionary process. This applies to skill and knowledge development also. The developing countries are trying to achieve development on a broad front simultaneously. The needed manpower also has to be developed at many levels simultaneously. A variety of approaches would have to be resorted to at the same time. It is, therefore, important to determine what are the varieties of manpower development mechanisms that need to be created to achieve this skill and knowledge development at many levels simultaneously and over a short period. It should then be possible to identify the kinds of inputs from developed to developing countries that are likely to have maximum practical value in the manpower development area.

1.4 The Structure of this Report

To recapitulate the main trends of our arguments so far, we started out by clarifying that technological self-reliance should not be confused with technological self-sufficiency. Few developing countries could hope to become totally self-sufficient in producing all the varieties of software packages and systems they need for their development. But depending on the level of preparedness of a country -- we identified three levels: minimum, medium, and high -- the software technology absorption and acquisition process could start at an appropriate stage. Software technology differs in essential respects from the traditional engineering technologies. Although software industry is rapidly acquiring the character of an engineering industry, it has a more design-intensive base. It is important to become aware of the status and trends of this industry in the developed countries to-day. Developing countries must begin with an analysis of their software needs before viable production structures could be set up to meet these needs. Manpower development is an essential facet of acquiring software self-reliance. Both in establishing software production structures and in manpower development, existing models in developed countries may not be automatically applicable to the developing countries.

The rest of the report develops these arguments in more detail. And based on the conclusions that arise out of these arguments, suggestions and recommendations are made. Wherever possible, recommendations are related to actual efforts that have been tried out or are being tried out. Advice based on actual experiences often carries more credibility than that based merely on theoretical analyses. Analysis and study of case histories are, therefore, of great value while discussing issues relating to socio-economic and technological development.

After an analysis of the software production scene in the developed countries in the next Section, Section 3 considers the software needs of developing countries. The four categories of needs are analysed and based on this analysis the varieties of software packages and services that developing countries require are worked out. Section 4 is concerned with the acquisition of self-reliance in software production by developing countries. Based on the several software categories that have to be produced and the different kinds of needs that have to be met, a variety of plausible production structures are identified. Initiatives that must be taken to promote software production are discussed. A prerequisite to acquiring self-reliance in software production is the availability of qualified manpower. Issues in manpower generation are discussed in Section 5 and some actual efforts tried out to develop manpower successfully are described. In Section 6, initiatives that are to be taken by the developing countries and by UN organizations are summarized. Annexures provide further illustrations of models that could serve as examples for adaptation.

2. SOFTWARE PRODUCED IN DEVELOPED COUNTRIES

2.1 Historical Background

Computers in the West had two distinct origins -- one concerned with the sorting and tabulation of census data and the other concerned with the computation of numerical tables. Hollerith in the USA and Babbage in England were the founding fathers of these two motivated efforts. The relevance and importance of Hollerith's ideas to commercial data processing applications were perceived fairly early on, and organizations like the National Cash Register Co. and, somewhat later, IBM were interested and involved in supporting and further developing these ideas into commercially viable products. Babbage's ideas and early attempts, on the other hand, were far ahead of his time, and had to be eventually abandoned without realizing a working model. But this motivation to build mechanical devices for computing numerical tables acquired fresh impetus in the early days of the Second World War and rapidly resulted in, first electromechanical, and later electronic computers.

These two streams of efforts continued to develop independently in the period 1940-1955: one punch-card-based and mostly electromechanical, concerned with business applications and market- and sales-oriented; the other paper-tape-based, using vacuum tubes, concerned with scientific computations and academic- and research-establishment-oriented. But in both these streams there was not much difference between hardware efforts and software efforts -- the same group of people were involved in both these areas, more or less. In many cases end-users formed part of these teams and helped in writing the application programs.

To-day, in less than three decades, the whole picture has completely changed. The hardware manufacturers, the software manufacturers, and the end-users have become three distinct groups. The end user does not depend on the hardware vendor to supply all the software, application packages, and services that he is in need of. The software manufacturer, analogously, does not necessarily write software exclusively for one hardware manufacturer, or one class of end-users. This state of affairs assists the hardware manufacturer also since it is not essential now for him to invest in software and application development in order to be able to sell his hardware to the end-users. No doubt this is somewhat of an idealized picture: this is mostly true in the case of microcomputers, and to a large extent true in the case of minicomputers. This is perhaps less true in the case of mainframes where a large part of the software effort -- certainly of the system software, and in many cases also of the application software -- still comes from the hardware manufacturer. But it is in the context of this devolution of market responsibilities that software development has acquired the characteristics of a self-contained industry. It has been claimed that in the USA the independent software product industry has developed into a $ 1 billion-plus business with 1400 vendors, 8000 products, and 30000 users [Welke 1980].

It is important to realize that the software industrial profile that we currently see in Western Europe and the USA has not come about based on any planned growth or systematic design. Various kinds of market pressures and demand pulls have provided impetus to the growth of the industry and the particular kind of structuring one finds there. In the very early stages the hardware manufacturer provided all the system software and much of the application software. All this software was not specifically costed but sought to be absorbed more or less in the total system price. Soon it became evident that software development and maintenance costs had become equal to, and were beginning to exceed, the hardware cost. It thus became essential to assign a commercial price to software and "unbundle" the sale of software and hardware. IBM's decision to adopt this strategy in 1969 contributed significantly to the creation of a separate software sector since, in principle, unbundled software did not have to be procured from the hardware manufacturer himself.

In the 1960s Defence and Space software needs in the USA opened up vast opportunities for independent software groups to undertake to develop the specialized software needed by these end-users on the basis of development contracts (mostly on a cost-plus arrangement). Much of this software was extremely large, complex, and had to meet stringent reliability and real-time constraints. Opportunities to work on these development contracts were of extreme value to the independent software contractors in evolving software production methodologies to manage large, complex software projects, and ensure quality and reliability of the products. These tested methodologies were then spun off to the commercial market to form standard industry practices.

Extremely rapid developments in hardware technology due to progress in microelectronics technology resulted in keen competition among hardware manufacturers to come out with newer and newer generations of hardware -- especially in the mini- and micro-computer areas. This widened the gap between hardware and software availability in the market. Not being able to cope with this problem these hardware manufacturers had no alternative except to accept and even actively encourage the emergence of independent software entrepreneurs. For these entrepreneurs the ever-widening information services market proved to be an ideal entry-level business because of its relatively low entry-cost, high rates of return and profit margins, and the great diversity of software requirements and the consequent impracticability of any one firm, or even a few firms, dominating the market. The software industry, as it exists now in the West, is thus a highly fragmented one spanning a very wide spectrum of products and services, and a very wide range of sizes of units and incomes: (See Table 1 for an indication of this diversity). We shall consider this state of affairs in more quantitative terms now.

2.2 Structure of the Software Industry

Welke, in the earlier referred to article, gives the following analysis of the software industry in the USA. "There are approximately 1400 vendors of largescale software

Table 1. Varieties of Software Products & Services

Software products and services range from analyzing a company's information processing requirements to programming, testing and installation of packaged software, integrated turnkey hardware-software systems to tackle specific applications, development of custom-tailored software, converting software to work in new environments, and performing all of a company's data processing on a service company computer (Facilities Management).

It is estimated that the number of business enterprises purchasing computer processing services, software products, professional services, or integrated systems will grow from 800,000 in 1983 to close to 1,200,000 in 1985 in the USA.

Communication Services

- Electronics Funds Transfer
- Electronic Mail
- Data Distribution
- Distributed Data Processing
- Computer Conferencing

Data Base Services

- Econometric
- Financial
- Legal
- Demographic
- News

Integrated Systems

- Claims Processing
- Distribution
- Legal
- Auto Supplies
- Medical
- Wholesale

Professional Services

- Analysis
- Design
- Programming
- Develop RFP
- Select hardware vendor
- Negotiate deal
- Hardware Maintenance
- Facilities Management
- Software Conversion
- Integrate System
- Custom Software
- Test, Install
- Audit System
- Evaluate System Performance
- Training
- Software Maintenance
- Documentation

Data Processing Services

- Interactive Computing
- Interactive Applications
- Batch
- Remote Batch
- Data Storage
- Back-up Computing
- Consulting

Software Packages

- System Software
- Support Software
- Applications Software (large varieties)
- Operating Systems
- Utilities
- Data Communication

(Source: World Communications, 1983)

products in the USA selling slightly more than 8000 products to more than 30000 users. These figures do not include or reflect the volume of business done by IBM or other mainframe manufacturers. In addition, some 2800 have software products available for minis (with or without hardware), or as a secondary line of business activity to whatever is their primary line."

"The typical firm in the software product business continues to be relatively small (approximately 30-40 employees) with annual sales of about $ 3 million. But the market place also has 50 or more large corporate firms in the software product business" with revenues amounting to several hundred million dollars each.

The profile of the software and computer services industry in Japan is, again, very similar to the one described above. The following summary giving the distribution of sales according to the size of the establishment brings this out quite clearly.

Japan in fiscal 1978
Total business : US $ 2.3 billions
Total workforce: 77313

Size of firm by number of employees	o/o of market (sales)	o/o of total workforce
1 - 4	0.8	0.7
5 - 9	2.5	2.3
10 - 29	11.5	12.5
30 - 49	13.0	12.5
49	72.2	72.0

(Source: Japan Computer News, Nov. 1981)

Continuing with the US software industry profile, Welke notes: "By far the majority of the software product firms specialize in a segment of the market, whether a hardware manufacturer, an industry discipline, or a type of application. Very few firms serve more than one industry unless they are in 'systems software' or have what in any way could be considered a full portfolio of product offerings".

"While there is much activity in the minicomputer market place, the majority of dollars are spent in the large-scale equipment marketplace ..... The minis have a smaller profit margin to work on with each sale, a lower price requirement, a less sophisticated user, and usually a local, rather than national, market."

Two other dimensions of segmentation concern the nature of software products that are marketed: (1) either they are basic commercial application systems (payroll, personnel, inventory, project management, financial planning and profit analysis, etc.), or more sophisticated systems programming software; (2) either the software products are custom-tailored to meet individual end-user needs, or are standard software packages.

Along the first dimension, the major part of the software effort in the US market at present would seem to be in the production of basic commercial application software. Along the second dimension, the market for packaged software would seem to be rapidly growing. It has been estimated, for example, that in 1978 more than half of all software used by US banks was provided in packaged form. The market for packaged software in USA in 1979 is estimated to have been $ 2.5 billion [Withington 1980]. Packaged software can be both application packages and system programs. Availability of easy to use packaged software has been an extremely important factor in the large-scale acceptance of mini- and micro-computers by a wide spectrum of users.

Table 2 summarizes the various sources of software products as we have been discussing them.

2.3 Strategies, Policies & Future Trends

The following factors have been identified as contributing to the phenomenal success of the software market in the USA [Welke 1980]:

1. Leasing instead of selling
   software is leased or licenced and not sold.
2. Pricing maintenance separately
   continued maintenance of a package is separately priced from the basic one-time sale price -- it could be anywhere from 10-15°/o of the basic sale price.
3. Long-term vendor-customer relationship
   an end-user expects a long-term commitment from the software vendor to keep the package updated and contemporary.
4. Designed to minimize installation costs
   software package design must have built-in features to minimize installation costs (which

Table 2. Sources of Software Products

1. Computer Manufacturer

2. User Groups (Co-ops.)

3. Software Houses

   Many still concentrate on contract programming to produce customized products to fill end-user needs. But increasingly production of packaged software meant for multiple-user market is becoming popular.

4. Software Brokers

   These obtain products from software developers who may lack marketing skills and/or inclination, and market them widely. Some brokers have staff to install, maintain, and support the software they sell.

5. Turn-key System Suppliers

   These supply total engineered systems (hardware + software) to meet a specific need of a customer. The customer must be knowledgeable to be able to specify his needs adequately, to monitor the reliability of the subsystems used to build the total system, to be able to test the total system performance before acceptance. In case the supplier disappears from the scene after system acceptance, the customer must have enough documentation (at the source-level) and know-how to be able to maintain the system, modify and/or augment it as needed, and so on.

6. Computer Stores

   Applies especially for micro- and personal-computer software.

normally involve some special tailoring to match the end-user's special constraints and needs).

5. Avoiding over-designing & over-selling reliable operation and continued support are more important for a successful market than producing an over-designed product or mounting a hard-sell campaign.

The development and growth of the software industry simultaneously prompted the development of protection measures through, essentially, three basic methods: trade secrecy, patent law, and copyright law. A detailed summary of the provisions of these protective measures as available at present in the USA and Western Europe may be found in [UNIDO 1983b].

The more important provisions are in the copyright law. The amended law (Dec.1980) defines a computer program as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result". By this definition protection is extended both to source code and object code. But the copyright law has many deficiencies and does not guarantee fool-proof protection. For instance, it makes "copying" an offense but not unauthorized use of a program. In view of this many owners of computer software have turned to using proprietary markings and nondisclosure agreements as more practical methods of protecting software. But these agreements are not always easy to monitor, especially on a world-wide scale.

More recent alternatives that are being explored to prevent unauthorized transporting of proprietary software involve the use of special micro-programmed chips that have to be plugged into the computer to provide sequencing signals without which the software would not run. The chips are sold along with the leased software. Incorporating software routines as micro-programmed codes in a proprietary chip is perhaps the most effective protection yet available. This is likely to become a general vehicle for software transfer because of the increasing ease with which customized microprogrammed chips can be fabricated. A forecast predicts that while microcoded application packages have so far been confined to household devices, personal computers, hand-held devices, and application-dedicated terminals, "application chips" will enter general use in future [Withington 1980].

It is predicted that the software market is bound to grow at a rapid rate on a continuing basis for the following reasons [Welke 1980]:

1. The market is nowhere near saturation. New applications are all the time becoming economical as the hardware and communication technology advances and prices fall. The volume of computing is continuing to grow.

2. The business office environment is rapidly becoming information-processing-oriented with the increased use of intelligent office equipment and tele-informatics.

3. The use of data bases, both centralized and distributed, is increasing in business and industry.

4. Analogous to office automation, factory automation is also advancing at a fast pace. Robotics and the use of intelligent tools are likely to become norms in factories of the future.

5. Integrated digital communication services (voice+picture+text+data) would widen the base for software products and services.

While one does not see any limits to the demand for software products and services in the foreseeable future, the shortage in manpower supply to produce all this software is already a live issue in most of the developed countries. It is here that developing countries, with their high quality surplus manpower resources, could have an advantage provided adequate training and transfer of know-how are arranged, and appropriate trading channels are created.

3. SOFTWARE NEEDS OF DEVELOPING COUNTRIES

3.1 Historical Background to Present Pattern of Use

In an early effort at an analysis of the problems and prospects of the application of computers to development by developing countries, a UN Expert Group suggested a four-level classification of countries according to their computer usage. The levels were termed: Initial, Basic, Operational, and Advanced. Table 3 summarizes the characterizing features of these different levels.

In so far as computer usage has not had an indigenous origin in any of the developing countries, all developing countries -- whatever be their current levels of computer usage -- must have started out from the 'initial' stage. This means that their very early introduction to computers must have been at the initiative of computer salesmen -- for the most part of one or other of the multinational mainframe manufacturers. The typical scenario in most cases would have been something like this: The first computers would have been imported by foreign-owned corporations, or sold to subsidiaries of foreign commercial companies. The first applications of computers would almost always have been in the areas of routine commercial clerical work -- book-keeping, accounting, invoicing, etc. In most cases these establishments would very likely have already mechanized many of these activities through the use of electromechanical Unit Record systems -- often also belonging

Table 3. Levels of Computer Usage

Level	Characteristics
Initial	There are no operational computers in the country. A few nationals have had contact with computing. The only local sources of information are computer salesmen.
Basic	There is some understanding of computers in government (and private) decision centres. A few computer installations are to be found. There are some nationals involved in computer operations. There is some education and training in computer technology in the country. Computers are used in basic government operations.
Operational	There is extensive understanding of computers in government (and private) decision centres. Among the numerous computer installations there are some very large machines. There are centres for education and training in computer technology and some are of excellent quality. They offer degree programmes in computer or information science. There is design and production of software and some manufacture of hardware. Computers are affecting many disciplines, particularly science, engineering, and medicine.
Advanced	Most government and administrative work is carried out by computers. There are well established professional activities and national meetings on computers. There is a complete range of quality education and training programmes. The number of computers, of all sizes, is increasing rapidly. Time-sharing, teleprocessing, and remote job entry are common. There is design and production of both hardware and software. Many technologies have been changed or are in the course of being changed. New applications of computers are found regularly. There is strong participation in and contribution to international activities.

(Source: United Nations 1971)

to the same mainframe manufacturer. The early computers imported into the country would have been systems withdrawn from use in the more advanced Western countries and refurbished, either after import into the country or prior to import, for sale within the developing country.

This scenario would apply to most developing countries whose introduction to computers was in the late 1950s or early 1960s -- that is, whose introduction was to the first generation computer systems. In the last two decades computer hardware has gone through three more generations of development. The introduction of mini- and micro-computers, and super-minis, has brought about major changes in the computer marketing scene in developing countries.

In addition to multinational mainframe manufacturers and their subsidiaries, very large numbers of local agents -- with little or no background in computing or computer technology -- have set themselves up in trading business selling mini- and micro-computers and their associated peripherals earning as commission some fixed percentage of each sale. Operating in what is essentially a seller's market, there is very little incentive for these middlemen to invest in providing training and after-sales service to their customers. Thus, developing countries -- especially those in the initial and basic level -- are increasingly facing the danger of having computers, peripherals, and software dumped on them with no adequate information or training concerning their use, and no support service to maintain them in operation for any length of time. These dangers raise important policy issues relating to the import and sale of computer systems in developing countries which these countries should analyse and cope with through appropriate monitoring and regulation.

Most developing countries in the initial and basic level would, however, find that -- whether computing systems are sold to them by mainframe manufacturers or local agents -- the first usage of computers in these countries is almost always in routine commercial applications, as earlier discussed. The reasons for this are readily understandable:

1. Clerical operations in commercial establishments are easiest to mechanize;

2. They are self-contained operations which, therefore, lend themselves readily to computerization using an in-house computing facility;

3. In Western countries, computer applications in business started out with these routine clerical tasks;

4. Software application packages developed for use in these countries could be readily transported and offered for use by commercial establishments doing similar work in developing countries.

These routine commercial data processing activities, however, seldom mature into more significant management applications such as, for example, planning, project management and control, modelling, simulation, operations research applications, and so on. These management tasks can be computerized only through building up enough internal know-how in an organization for systems analysis and software development. From the viewpoint of developing countries these management applications, and also applications in transaction processing, real-time process monitoring and control, engineering design, text- and word-processing, information storage in a data base, query, and retrieval, etc., are by far more important than the purely commercial clerical ones earlier discussed. Multinational vendors and local agents seldom concern themselves with such application areas because of the enormous preparatory and training efforts needed to successfully computerize such applications, and also because ready-made software packages are seldom usable in these application areas without substantial modifications.

In the rest of this section we shall analyse more carefully what the real application needs of developing countries are and what kinds of software development know-how they imply. This would enable us to identify the sorts of organizations that would need to be set up and efforts that would have to be made to acquire and grow these kinds of software development know-how.

3.2 Four Categories of Needs

We saw earlier in Section 1 that the computational needs of developing countries can be broadly divided into four categories:

1. Applications concerned with basic needs and the agricultural sector;
2. Applications in the government sector;
3. Applications concerned with productivity increase in the industrial and service sector;
4. Software operations relating to trade in software packages and services in the international market.

We shall analyse the characteristics of each of these categories in more detail now. Our objective is not to produce an exhaustive list of applications in each category but to identify the nature of applications in each category in broad terms so that we can evaluate the software capability needed to design and produce packages to serve these application areas.

3.2.1 Meeting Basic Needs & the Agricultural Sector

The priority concern of a developing country is meeting the basic needs of that section of its population which, for various reasons, is poor, underprivileged, and economically unable to compete under open-market conditions to better its lot. In most developing countries this section of the population tends to live in rural areas, in small, mostly economically unviable and geographically isolated communities. Tradition-bound, lacking in formal education and marketable skills, this population tends to eke out a subsistence living off marginally productive lands.

Developing countries committed to developing the human potentials of this section of the population to the fullest extent possible and enabling them to contribute effectively to the economy of their countries have an extremely complex problem to tackle. These countries have to plan and implement a coordinated programme of field-level activities that simultaneously address the following issues as they relate to this depressed section of the population:

provision of primary health-care and essential education;

provision of housing and water for drinking and cultivation;

improving the economic skills and nutrition;

provision of capital infrastructure to launch individuals and families on economically remunerative activities;

integrating the communities with the institutionalized financial and market structures of the country;

and so on. While addressing these immediate and local issues, integrated rural development plans must be oriented towards long-term improvement of the non-urban areas in order to bring them into the mainstream of economic activities and make them productive and self-sustaining.

The implementation of integrated rural development programmes of this kind assume great complexity because of the following factors:

1. the large size of the population to be serviced;

2. the multiplicity of agencies involved in providing the services;

3. the close coordination between the agencies that has to be maintained to ensure effective results;

4. the long time periods over which these integrated activities have to be maintained and operated;

5. the necessity for ensuring proper temporal sequencing of the varieties of sub-activities involved in the projects;

6. the great geographical distances that have to be covered to bring into the planned action a sizeable population.

It is precisely in coping with these complexities that computers could play on extremely effective role. As instrument_ of management at the rural level, they could provide an integrated framework of assistance to the government to plan, formulate, administer, monitor, and control the varieties of programmes that make up the integrated rural development projects. Management information systems that can play this role have to be tailor-made to meet the requirements of each country and, possibly, each subregion in a country. By integrating computing and communication requirements of management in one framework, such management information systems for the rural sector could radically transform the style of governmental functioning in developing countries. Clearly this application area is specific to developing countries and models cannot be imported from developed countries since analogous problem areas do not exist there. Expertise in system analysis and software development for coping with this application area must necessarily be indigenously grown within the developing countries themselves.

In the more developed agricultural sector, a variety of computer applications arise: in land use planning, water management and irrigation control, weather monitoring and forecasting for farmers, and so on. Several implemented accounts of these applications were discussed in a recent international conference [Kalman & Martinez 1981]. A farm information system for assisting small farmers discussed in this Conference is of particular interest in confirming and supporting our arguments earlier for the relevance to developing countries of management information systems in the rural sector.

This system, called SCAPA for 'System for Computer-aided Agricultural Planning and Action', is being run as a pilot project in Malaysia for the Rubber Industry

Smallholders Development Authority. The computerized system is based on the following premise [Robson 1981]: "If farmers could time their crop operations correctly, use the correct input quantities of seeds, fertilizer, spray chemicals, and irrigation water at the right time in recommended amounts, the crop production on small farms, in developing countries, could be increased without any additional expense by the farmers. If this were to be done, extra services of extension workers would be required, people who are already overburdened as it is. Any factor that could release trained advisory workers would help to make such a scheme a success. The use of a computer-based information system would fit perfectly into this development environment. It would not only give immediate returns in production, but also provide a basis for longer-term imaginative teaching and extension activities, leading to a full economic and agricultural growth situation."

3.2.2 Government Sector Applications

Every government requires systematic maintenance of records -- of property holdings, of transactions of all kinds, of productions and consumptions, exports and imports, and all kinds of status indicators. Foreign reserves, trade balances, private and public debt, consumer spending, price indexes, educational statistics, and similar information are essential to monitor and control the performance of the various ministries and departments of a government.

Most of these applications lend themselves readily to computerization. They involve tabulation, storage, and retrieval of vast quantities of data; indexing, classifying, and sorting; and statistical computations of various kinds -- regression, correlation, time series analyses, and so on. More sophisticated applications such as modelling, planning and simulation would call for more complex software production and computer usage. Graphics capabilities of various sorts would be needed in surveying, mapmaking, land-use planning and analysis, etc. Budget planning and control is another vital area where computers could be of great help.

3.2.3 Industrial and Service Sector Applications

Table 4 gives an overview of the primary and secondary production sectors in a fully developed economy. Table 5 gives an analogous overview of the service sector. In practically each of these subareas of production and service activity computers can be used to increase productivity. Applications in these areas are more or less well-developed in technologically advanced countries. Where the technology and the organizational set up supporting these activities in a developing country are identical to that in advanced countries, application packages should be directly transportable from the developed to developing countries. Where the technology and/or organizational set up differ, existing packages may have to be modified more or less extensively. For developing countries, systems analysis

Level 0	Level 1	Level 2	Level 3	Level 4
ENERGY Wood Biogas Wind Oil Electricity Thermal Hydel Nuclear Fusion Solar	FISHERY	FOOD . BEVERAGES . TOBACCO	CHEMICALS PETROLEUM & COAL PRODUCTS • Basic chemicals Fertilizer • Paints, Inks Explosives Pesticides, Soaps, Pharmaceuticals, Cosmetics • Liquid Fuels, Lubricants	VEHICLE MANUFACTURE • Road • Rail • Sea • Waterways • Air
	AGRICULTURE Food Crops Cash Crops Livestock meat poultry dairy	TEXTILES . CLOTHING FOOTWEAR	WOOD & METAL CRAFT WORK	ELECTRICAL GOODS Components Equipments
	FORESTRY	WOOD . WOOD PRODUCTS	PRINTING & PUBLISHING	ELECTRONICS GOODS Components Equipments
	MINING	PAPER . PAPER PRODUCTS	FABRICATED METAL PRODUCTS Metal Containers, Tools, Nuts & Bolts Spring & Wire products	ELECTRONIC SYSTEMS Communications Computers
	QUARRYING	IRON & STEEL NONFERROUS METAL	RUBBER & PLASTIC PRODUCTS	
		NON METALLIC MINERAL PRODUCTS glass, glass products, bricks, Ceramics Cement, Concrete Plaster, Stoneware	PRODUCTION TECHNOLOGY GOODS Machine Tools — Manual — NC CAM ROBOTICS	

Table 4. Primary and Secondary Production Sectors

CONSTRUCTION Buildings Water Supply Sewerage Roads Rail Tracks Canals & Waterways	**MANAGEMENT** Planning & Research Administration Marketing — Advertisement — Other Promotion Sales — Wholesale — Retail	**FINANCE** Banks Insurance Real Estate Stock Exchange
TRANSPORTATION & STORAGE Rail, Road, Sea, Air Inland Waterways Vehicle — Operation — Maintenance Warehousing		**COMMUNICATIONS** Post, Telegraph, Telex Telephones, Digital Data Equipment — Operation — Maintenance
RECREATION & TOURISM Cinema, TV, Radio Audio, Sports, Racing Clubs, Hotels, Restaurants Travel Agencies	**SOCIAL OVERHEADS** Government National.State.Local — City — Ward — Village Law & Order Police — Prevention — Detection Justice Defense	**COMMUNITY SERVICES** Education & Basic Research Health : Human, Veterinary — Paramedical — Testing & Diagnosis — Hospital Care Libraries, Museums, Galleries Weather Services Personal Artisan Services

Table 5. Services

and modelling is, therefore, unlikely to be the main issue in promoting computer applications in these areas. Identifying priority areas for computerization is likely to be a more urgent and serious problem: (see in this context [Foster 1982]).

3.2.4 Export Sector Applications

The application areas, here, would be determined by the markets being served. The computational sophistication could range from simple data preparation in computer readable form at the low end, to systems analysis, software design and program development, testing and documentation at the high end. The main incentive for developing countries to invest in the export sector is to take advantage of the fact that software design and production is a labour-intensive task. Increasingly software production -- both system and application software -- is being contracted out in the advanced countries. And, as we saw earlier, human resources there are in short supply and consequently very expensive. The high quality surplus human resource available with the developing countries could be made to yield handsome returns if provided with appropriate training to produce software to meet the requirements of the advanced countries. Establishing viable trading channels may be a big hurdle and initiatives are needed to tackle this problem as we shall see later.

3.3 Software Products and Services Required

A detailed analysis of the application areas covered in the four categories discussed in Sec. 3.2 would show that the software packages that serve these application areas can be classified into ten broad groups. These are listed below giving for each group some indication of

1. the application areas served,
2. the nature of the software items in this group from the viewpoint of software engineering,
3. the expertise and background needed to produce such software.

We shall consider in more detail in the next section, based on this analysis, the institutionalized structures and efforts needed to build the kind of software products listed here.

1. System Software

This includes operating systems, compilers and utilities like text editors, cross-assemblers, pre-processors, etc. Basic software for specialised areas such as database management, graphics etc. is also included in this category. A characteristic feature of such software is their high degree of dependence on the specific hardware for which they are designed. Though some of the more recent compilers and utilities are designed to reduce the machine dependent code to a great extent, lack of portability is an important feature of this category. The design and development of these systems calls for a sound background in computer science; roughly of the level of B.Tech./M.Tech.(Computer Science).

Most of these systems are bundled with hardware and fresh design/development efforts are needed only within captive software groups in manufacturers' organizations. However periodic update/maintenance capabilities are essential even for leased systems and these capabilities would arise from people with the same background as outlined above. Specialized software may also be handled by consultancy groups.

2. Computer Communication

Computer communication software for voice (telephone), message, telex and data switching is included in this group. Similar software is also required to be supplied by manufacturers of computer systems for inter-connecting computers. Considering the rapidity with which computing and communications are coming together, a variety of specialized systems for use in different application areas will need to incorporate communication software. Network capabilities will, more and more, be incorporated into the operating systems of all computer systems in the medium and large categories. Software of this type can be produced only by well-organized teams consisting of highly trained staff. Postgraduate training in computer science and communications will be required for those working at the design level.

3. Systems for Real-Time Transactions Processing

Examples in this category are on-line systems for train/airline reservations, banking etc. The emphasis here is largely on the specific application areas, especially in the emerging context of the availability of infrastructural support with simplified interfaces for communications, database management and transactions management. There is considerable scope for highly portable systems of this kind, especially after these interfaces have achieved some degree of standardization. A computer science background is desirable for any work in these areas; however, knowledge in the applications areas is undoubtedly the major requirement for system analysis and design.

4. Real-Time Embedded Systems including Command and Control

Process control, defence and simulation systems are some of the examples of this group. These are highly specialized software systems that are usually packaged as integral parts of other equipment. Maintenance of such systems is a highly specialized task and, most often, the degree of integration with the equipment is so large that the software is not easily adaptable to other systems. The background required for productive work in this area is usually specific to the equipment under consideration. However, knowledge about systems level programming is always required.

5. Commercial Data Processing Software

A large chunk of the software developed in many countries falls in this category, including application software for commercial use, such as accounting, payroll, sales accounting, inventory management, invoicing etc. Less expertise is demanded of software personnel involved in this area, particularly relating to computer science. The most essential requirement is a good grasp of high level languages like COBOL and Basic. Knowledge of database management, transaction processing and even business graphics techniques will progressively become important. Though heavily customized systems are being produced today, there is considerable scope in this area for the use of specialised software tools that could develop customized end-user-systems from parameterizable modules. Program generators are increasing in popularity. Portability is important. Fresh graduates with hands-on training for 6-12 months should be able to undertake the development of application systems, though the design and development of basic tools may call for the expertise of system software personnel. The end user will normally maintain such systems, though packages fully supported by specialized software houses may also be envisaged.

6. Scientific Software

Included here are basic scientific software libraries like IMSL (International Mathematical and Statistical Library), SPSS (Statistical Package for the Social Sciences), the NAG (Numerical Analysis Group) Library and the Scientific Subroutine Package (SSP of IBM). Very often some basic scientific library package is included in the bundled software supplied with the basic system; it is also common for user groups to augment such software with locally produced routines. Augmentation of this kind requiring only some knowledge of FORTRAN is trivial; but the production of a library of the kind listed above is a highly specialised job requiring expertise of a high order in numerical analysis as well as in library packaging. Since thoroughly tested packages are available for lease on most systems at reasonable rates, leasing such systems should perhaps be preferred to ab-initio development within the country. Adaptability and maintenance are usually easy.

7. Engineering Software

Some examples in this area are structural analysis packages, pipe-design systems, CAD/CAM systems and other general design packages. Design and development of these systems can be undertaken only by specialist groups who also have a good degree of expertise in high level languages and an awareness of advanced techniques in large-scale software development. Knowledge in special topics like computer graphics and CAD and database management will be useful. In view of the high cost of such software when purchased from abroad, where reasonably large numbers of engineering consultants (with background in software) are available in the country, it would be desirable to consider development of these systems indigenously. High portability across manufacturers and types of machines is a feature of these systems. Maintenance including upgrading is best done by the system developers.

8. Planning and Decision Support Systems

Systems included in this group are information systems that range from simple routine reporting systems to those of all kinds which make use of quantitative methods in Operations Research like mathematical programming, PERT/CPM and simulation methods. Generalized packages for quantitative methods in OR fall roughly in the category of engineering packages, with the difference that the developers of these systems require a good background in management science and software. Management Information Systems (MIS) which also fall under this category, do not necessarily demand extensive technical background of management science but some of the requirements outlined in the context of commercial systems apply rather closely here. There is good scope for developing portable systems by specialised groups.

9. Office Support Systems

These include the emerging areas of electronic mail, conferencing, word processing, document production and so on. There will be heavy emphasis towards micro-processor-based software in this area. The expertise required is largely similar to what is applicable for system software, with the difference that work in this area will be mostly in high level languages like Pascal, C, or other implementation languages. Portability is an important consideration and purchase or lease of systems from abroad is not recommended since a concerted effort in the development of such systems would be highly cost-effective in view of the large number of systems that could be sold within the country, especially if implemented on microprocessor systems. Maintenance of such systems should be the responsibility of identified specialist groups.

10. Specialised Software

This is essentially a catch-all category for software that does not fit into the above categories. Examples of such software are Computer Aided Instruction (CAI) packages, graphic arts design systems, computer assisted printing and publishing systems, etc. The expertise required in these areas varies from system to system and the purchase-lease-or-develop decision will also have to be taken on a case-by-case basis.

4. ACQUIRING SELF-RELIANCE IN SOFTWARE PRODUCTION

4.1 Present Situation and Lessons to Learn

Our earlier analysis of the software industrial sector in the advanced countries, and our analysis above of the historical background to the current state of information processing activities in the developing countries on the one hand, and the real information processing needs of these countries on the other hand, should enable us to draw some lessons. It is worth listing these first so that we can use this as a starting point to determine the kinds of structures and efforts needed to acquire self-reliance in software production in the developing countries.

Table 6 gives an idea of the value-added components in the various segments of the information processing industry. It is evident from the table that software and people are the two ubiquitous value-added components in the information industry. And since software production itself is a very labour-intensive effort, the single most important factor in the information industry is qualified manpower.

As we saw earlier, the software industry in the USA and many European countries has not taken shape through any organized efforts and systematic planning. It has grown through market pressures and demand pulls of various kinds. However, opportunities to work on contracts to meet defense, space, and similar national needs have been of indispensable value in the growth and professional maturation of the software industry. Even now the major part of the production

Product or Service	Hardware	Software	Communications	People	Data Base
Consulting				X	
Education				X	
OEM Distributors	X	X		X	
Turnkey vendors	X	X	may be	X	may be
Software product		X		may be	
Professional Services		may be		X	
Information Processing Services Batch	X	X		X	may be
Remote Batch	X	X	X	X	may be
Interactive	X	X	X	may be	may be
On-site	X	X	may be	X	may be
Hardware manufacturers	X	X		may be	

(Source: Welke 1980)

Table 6. Value-added components

effort is concerned with meeting the internal demands of these countries.

The software market is highly fragmented both in terms of the varieties of groups that service the market and also in terms of the varieties of products that are offered for sale -- prepackaged as well as custom-tailored. There have been various attempts at standardization of language specifications, interfaces, protocols, and so forth. These efforts at standardization still continue. Nevertheless, on the whole, standards for production and product interfacing are as yet non-existent.

The trend is towards increased software ownership protection and production of portable products that are

1. easy to instal in multiple installations with minimum installation and maintenance costs,

2. easy to augment and update,

and 3. easy to use.

Office automation, robotics and factory automation, tele-processing, custom-programmed chips, and digital communication technology are all bound to have major impacts on the software industry in determining the products and services that would dominate the market in the coming years.

In contrast to this picture, the current picture in the developing countries, as we saw, may be summarized as follows:

1. To a very large extent the needs of the information processing sector are met through imports;

2. Both hardware and software are imported without a systematic analysis of the needs of the countries and an identification of the right systems to meet these needs;

3. The import and sale of systems (hardware and software) are done at the initiative of multinationals, their subsidiaries, or their agents. Quite often systems are over-sold, remain under-utilized or unutilized due to lack of proper training, and after-sales service to maintain these systems in satisfactory operation.

4. There are very few software companies or other third-party consultancy agencies providing effective assistance to the end-users;

5. Standard application packages tend to be imported without adequate understanding of their logic and operation, and without adequate organizational preparation for their use;

6. There is usually very little coordination in the import of hardware and software into

the country. The same software packages are imported repeatedly by a variety of organizations without any horizontal transfer of know-how. Again, each system tends to be imported on the basis of ad-hoc negotiations thus resulting in the deployment of a wide variety of systems in the country which are difficult and expensive to maintain;

7. The real needs of the country are seldom analysed and systematic local efforts mounted to solve them through the development of adequate software packages;

8. There is little coordination between the academic institutions that train computer and software engineers and system specialists, and the industries and other end-users who ultimately have to employ them;

9. Bright, young persons highly trained at great public expense in the computer field in the educational institutions either leave the country to work in the advanced countries of the West, or become marketing and sales agents of foreign companies.

Of course, not all developing countries suffer from all the inadequacies listed above. The situation would vary from country to country depending on its level of informatics development. Many of the inadequacies listed above are likely to be true of countries at the initial or basic level of informatics development. Some of the larger developing countries have been conscious of many of these issues and have been trying to enforce monitoring and regulatory measures to ensure a planned growth of informatics activities.

Three basic lessons that developing countries must learn from the history and current trends of the informatics field in the advanced countries are the following.

1. In acquiring hardware, as well as software, for production purposes, the determining consideration in choice-making must be maximization of the added-value. Turn-key solutions, however attractive in the short-term, are almost always detrimental from the long-term viewpoint.

2. A viable software industry can be built up only by developing an informed and competitive internal market.

3. To remain competitive, whether in the internal or international market, the goal must be to attain methodological sophistication in software design and production. Modularity, maintainability, documentation are important factors in the production of reliable software. Appropriate production tools and environments must be created to ensure professionalism in software production.

Directly arising out of the above lessons there are three basic requirements that have to be met to ensure self-reliant software production in the developing countries. These are the following.

1. Access to appropriate hardware systems to develop and use software;

2. Identification of the real software needs and creation of teams to produce the needed software;

3. Education and training of manpower, both to produce software and to deploy and use the application packages produced.

We shall consider now in some detail the structures and efforts needed to meet these three requirements.

4.2 Suggested Structures and Strategies

In discussing the structures and strategies for self-reliant software production we shall, as stated in the Introduction, formulate them as they apply to developing countries at the high-level of technological competence. This is done to make the list of suggestions as comprehensive as possible. Subsets of these suggestions would apply to countries at other levels of competence: it should be fairly clear how to select these subsets for each country depending on its informatics competence and level of preparation.

4.2.1 Structures for software production to meet software needs

We discussed at some depth earlier in Sec. 3.2 four categories of applications of importance to developing countries: 1) basic needs and the agricultural sector, 2) government informatics, 3) industrial sector (goods and services), and 4) export market in software. In Sec. 3.3 we analyzed the major software categories that cover these application areas.

It will be seen from our earlier discussions and analysis that some software packages are very hardware-specific; intimate knowledge of a particular hardware is needed to produce these. Some others are application-area-specific; wide and deep knowledge of the application areas is needed to produce these. Some require a good formal background in computer science for their design and implementation. Some others require a good theoretical background in the concerned application areas. Some need access at production time to the hardware in which they are to be ultimately implemented; some others can be readily transported from machine to machine. Some require large team efforts for their creation; some could be produced by very small groups or even by single individuals. There is scope for all sizes of software industrial groups -- from the very small to the very large -- to coexist economically. Software industry is preeminently suited for exploitation by small entrepreneur groups with specialized application-area knowledge. We saw that the present situation in the advanced countries reflect well this variety and built-in segmentation of the software industry. These observations provide a clue to a plausible structuring of the software industry in developing countries. This is shown in Table 7.

The profile in Table 7 shows a natural segmentation of the software production and service sector into four major groupings. The commercial (profit-making) section would be made up of a very large number of small software consultancy, service, and product groups; a limited number of large software and engineering consultancy houses; and the software groups belonging to the hardware manufacturers.

Then there is need for a certain number of resource groups to play a variety of developmental and promotional roles. For example:

1. technology development and spin-off,
2. education and training to absorb new technology and to ensure professional growth,
3. operation of design environments for specialized applications and providing promotional and consultancy services (e.g., CAD centres with integrated design, drafting and documentation facilities),
4. operation of simulation and/or developmental centres for dedicated applications and providing assistance in tailoring, updating, and augmenting packages for the concerned

Table 7. Structures for Software Production

Commercial (profit-making)

1. Small software consultancy, service and product groups

2. Large consultancy groups

   2.1 Software Bureaus
   2.2 Engineering consultants

3. Computer Manufacturers' Software Groups (Committed hardware)

Non-commercial (not-for-profit but self-supporting)

4. Resource groups with appropriate computer facilities

   4.1 Linked to teaching/training institutes
   4.2 Dedicated to specialized application software
   4.3 Supporting software development and service activities
   4.4 Dedicated primarily to technology development and spin-off
   4.5 Dedicated to individual organizations (in-house)

Government

5. Service and support for Government informatics

   5.1 Central Government Ministries and Departments
   5.2 State-level centres
   5.3 City and Municipal Level centres

Data network operations

6. Public data network utility

7. Special user-group data network utilities.

applications (e.g., railways, airlines, energy, other real-time and transaction processing applications).

All these resource groups would function around dedicated computing facilities and are best operated as non-commercial (i.e., not-for-profit) but self-supporting organizations, i.e., earning enough to meet their operational expenses and possibly producing some amount of surplus to finance their growth in a limited way. Clearly a certain geographical distribution would be necessary to service the needs in the various parts of the country. In addition to these promotional and technology-spin-off resource groups, large corporations, agencies, etc., would require in-house resource groups to service their own internal needs.

Next, computer centres and associated software groups are needed to service the government informatics requirements at the central, state, and municipal levels.

Lastly, for a variety of applications (e.g., banking, transport, tourism, etc.) data communication infrastructure is of essential importance. Therefore, agencies must be created to design, construct, and operate data transmission networks and provide the associated network utilities.

4.2.2 Initiatives needed to promote software production and usage

The marketplace and commercial considerations would determine the growth and functioning of large software bureaus and consultancy firms providing software services. However, it is essential to take positive steps to promote the growth of software consultancy and service infrastructure in identified, high priority sectors, for example, energy, primary industries, engineering industries, transport, construction, etc. A certain number of groups with appropriate backgrounds in these application areas must be identified and consciously groomed to support the software needs of these sectors. An analogous strategy is needed to establish simulation and/or development centres for dedicated application areas.

For the growth and viability of small entrepreneurial groups the single most important need is access to an appropriate software development environment. A workable solution that has been successfully adopted in some developing countries is the creation of national and/or regional computer centres. These operate large computer systems with the latest technological capabilities offering excellent program development environments. Computer services are made available to entrepreneurial groups on a cost-plus basis. The staff of the centre could, in addition, offer consultancy in programming and short-term training courses in software design and production, publish self-teaching manuals, and offer other assistance.

For meeting the government informatics needs, perhaps the single best solution is the operation of government computing centres along the lines indicated above. The customers here would be exclusively government departments and other government agencies. The training of government end-user groups, and assistance in systems analysis, consultancy, and software production would form an integral and essential responsibility of the computer centre staff. The emphasis should be in developing generalised software systems useful across departments, wherever possible, so that investment in software production benefits governmental units at many levels -- Centre, State, City/municipal -- simultaneously.

As sophistication in computer usage develops within the government sectors, networks could be created with front-end mini- or micro-computers in the departments to serve as in-house facilities, linked to state- or national-level central systems operating large/very large mainframes. This arrangement offers great potential for maintaining centralised files with access facility provided to more than one ministry or department thus promoting interministerial and interdepartmental cooperation and coordination in information usage for purposes of control, promotion and planning. The National Informatics Centre, a national-level government computing centre located at New Delhi, is successfully experimenting with this methodology.

Networking is also the best solution to the systematization and coordination of information acquisition, dissemination, and use across the various hierarchic levels of the government (ward- city- district- state- nation).

Taking advantage of the cost-reductions made possible by microelectronics technology, micro- and mini-computers must be liberally deployed at the various levels. Compatible information and data structures must be carefully designed to meet the functional needs at the various levels so that repeated raw-data inputting could be avoided. Ultimately all these computers should be linked by data communication networks so that computer-mailing and other conferencing possibilities that the technology affords could be exploited. But till the data communication infrastructure is developed to a satisfactory level, information could be manually transported across levels in suitable computer-readable media -- floppies, disc packs, tapes, etc. Bulky printouts should be avoided except where functionally needed.

In dealing with government informatics one other aspect that ought to be emphasized is this. The traditional approach to dealing with quantitative information in governments -- this is especially true of governments in developing countries -- is to collect and compile data and perform various kinds of statistical analyses on them: for example, time series analysis, correlation analysis, etc. These are extremely important activities to monitor and assess performance at the macro-level for purposes of planning etc. However, in developing countries, while usually much attention is paid to planning of programmes, very little attention, in general, is paid to their implementation. At this level what needs to be monitored and controlled are actions (or activities) and their coordination, sequencing, scheduling, etc. The usual governmental machinery has very little understanding or training to deploy information technology for purposes of <u>event monitoring and control</u>. It is here, as we discussed earlier in Sec. 3.2.1, that computers in combination with communication and mailing capabilities can play a very significant role in transforming the style of government operations and modernizing the entire administrative outlook of government departments and agencies.

Strategies for software export promotion are more difficult to enunciate on a universally applicable basis. Export markets in the advanced countries are difficult to penetrate without an adequate marketing presence in these countries. This must be backed up by technological credibility to design packages and deliver services of acceptable quality at competitive prices. Geographical distances between the developed and developing countries compound the problem of effective and continuous interaction between customers in the former and software groups in the latter. One solution that has worked in some instances is to use a software group in the developed country itself as a collaborating partner. This group would take the initiative in identifying the customers and their needs and in providing continued interaction with them. The job execution could be shared by the collaborating groups based on what could be carried out at a distance, and what needs to be handled close to the customer. Small group entrepreneurs may find it difficult to establish such collaborative partnerships on their own. Governments or public sector promotional corporations could provide assistance in promoting and underwriting such partnerships.

Another way of effectively providing software development services from a distance is through the use of computer networks. Software could be developed on customers' own systems provided access to them via a network is possible. In many instances the different time zones in which the customers (in developed countries) and the software providers (in the developing countries) live make this a very feasible proposition since the idle night time of the computers could be productively used. This possibility is, of course, predicated on the availability of data communication facilities -- via satellites, mostly -- to the developing countries.

5. ISSUES IN MANPOWER DEVELOPMENT

5.1 General Considerations

The successful development of a software industry is ultimately dependent on the availability in requisite numbers of qualified manpower. The software industry needs not only persons with computer science knowledge, but persons with knowledge of application areas together with the requisite capability in systems analysis and programming. Also a very large number of persons who could function as members of programming teams is needed, as also persons with documentation and marketing capability. The formal qualifications minimally needed span a wide spectrum -- from secondary school education to Ph.D. in computer science and/or in the relevant application areas.

"A Modular Curriculum in Computer Science with Guidelines for its Implementation in Developing Countries" has been in preparation under a UNESCO contract by a Working Group set up by the Technical Committee for Education (TC3) of the International Federation for Information Processing (IFIP) [WCCE 81]. This effort addresses the training of the following categories of computer specialists.

"1. Programmers, with the ability to analyse a data processing problem, to design or select appropriate algorithms and to construct a well-structured computer program.

2. System Programmers, with special knowledge and ability to develop the software required

to make computer systems effective i.e., operating systems, compilers, interpreters, communication software ...

3. Application Programmers, with knowledge of specific fields of applications and of data processing techniques, and ability to construct substantial programs and packaged software for users.

4. Coders (Assistant Programmers), with detailed knowledge of appropriate programming languages and who can, under supervision of 1), 2), or 3), produce working programs for computer systems.

5. Information System Specialists, able to analyse the information requirements of organizations, to design appropriate systems within which computers may play a part, and to implement these systems with the assistance of programmers and computer operations staff.

6. Computer Operations Personnel, (up to the level of computing service managers) able to control the handling of data, the operation of computer hardware and software, and the scheduling of work through the computer system.

7. Data Processing Managers, with knowledge and experience of the computing needs of an

organization, the ability to manage a computer service which may include staff of all the previous categories, for the benefit of all its users.

8. Computer Scientists and Researchers, whose main concern is with the further development of the principles and techniques of computing or informatics."

This classification omits, as the Working Group points out, other skilled persons needed in the computer services area, e.g., hardware engineers, maintenance engineers, maintenance technicians, etc.

While training on the job and training through part-time courses could, and should, be provided as we shall discuss in detail presently, developing countries must work out a phased programme to make computer literacy a part of regular education at all levels. Certainly all universities, colleges, and possibly many schools (especially polytechnics and vocational training schools) should have access to computing resources. Providing effective training in computer literacy and vocational programming should be feasible through the use of low cost microcomputers and small minicomputers. More elaborate facilities and teaching programmes would be needed to train sophisticated computer professionals, of course. Since software technology is still going through a rapid development phase, it is essential to provide means for software professionals to update their formal background on a continuing basis. Systematized schemes for refresher courses for continuing education based on Workshops, seminars, and short-term advanced courses, are of extreme importance.

Considering these issues in somewhat greater detail, we can say that the manpower generation problem in the software sector for developing countries in the Initial, Basic, and Operational levels of informatics development consists of the following specialized components.

1. Generation of fully qualified software scientists and engineers.

   These persons would go through a formal educational programme in Computer Science and Technology, up to the bachelor's, master's, or doctoral level. They would work as senior software engineers, senior system analysts, system programmers, or R&D scientists in software technology.

2. Retraining of qualified application area specialists and first degree holders.

   Very large numbers of students who graduate with a first degree in science, commerce, and other subjects with a reasonable exposure to mathematics, can be retrained to be efficient programmers. Analogously, persons with degrees in specialized engineering and technology areas can be retrained to be qualified application area analysts and programmers. Such persons tend to drift into the software industrial market as entry-level programmers and coders. With systematized training in the formal aspects of programming methodology and software engineering, they could be moulded into capable application area specialists so that their specialized formal backgrounds in the concerned application areas could be put to productive use.

3. Vocational training and extension-training in programming.

   Students who opt for vocational training at the secondary school level, and carefully selected secondary school dropouts, can be trained to be coders, computer operators, data-preparation personnel, and also maintenance technicians.

4. Training of public- and government-sector end-user groups.

   Most of the development-related activities in developing countries -- for example, basic needs programmes, agricultural extension activities, transportation, energy generation and distribution, communication, and so on -- are in the government and public sectors.

The training of end-user groups in these sectors in computer usage assumes great importance to facilitate efficient deployment of computers to support development-catalyzing activities.

5. Public education in informatics awareness.
Public education in computer literacy and informatic awareness is of very great importance to developing countries. A prerequisite to effective computer usage in development-catalyzing applications is the awareness that information is an economic commodity. Bringing about this awareness both within the governmental sector and among the general public is an essential requirement for development. [Narasimhan 1982].

5.2. Some Specific Efforts

UN and its agencies like UNIDO and UNESCO have been involved in promoting informatics for development. The first detailed report on "The Applications of Computer Technology for Development", was published by the UN Secretary-General's office in 1971: (a second report was issued in 1973). UNESCO has been involved in promoting informatics for development in terms of many detailed programmes [UNESCO 1980]. Manpower development for informatics forms an important component of these efforts. The curriculum development project earlier referred to is part of this initiative. UNIDO activities related to informatics are undertaken as part of its efforts to strengthen the technological capabilities of developing countries in the fields of emerging technological advances. Several non-governmental organizations have also been involved in programmes for promoting informatics for development - education and training forming an important part of these efforts, again. Many international workshops, seminars, and conferences have been held in the last ten years on computer education at various levels. Perhaps the largest and the most important of these was the World Conference on Computer Education (WCCE) held in Lausanne in 1981 organized by IFIP. Proceedings of all these conferences contain useful descriptions of efforts underway in various developing countries to promote computer education and deal with manpower development in this area. These should be consulted for suitable models to adopt/adapt: (see also in this context EDINFO-82 conference record). We shall briefly outline here a selected set of efforts which have succeeded very well in meeting their objectives and which could serve as models for adoption by developing countries on a larger scale.

5.2.1 The Threshold Scheme

This scheme [Penny 1982] has three features distinguishing it from other training programmes for school-leavers:

1. it is not dependent on school examination rules - selection is based on specially designed written tests and interviews,

2. though lasting one academic year
   it is split up into 4 alternating periods,
   two in college and two in industry,

3. while aiming primarily at computer programming,
   it also provides a broad base of computing
   and business.

It is managed by the National Computer Centre (NCC), UK, with funds provided by the Manpower Services Commission (MSC), UK. In 1982, some 60 Colleges of Further Education and several thousand employers were participating in the scheme in providing tuition and industrial training. About 1500 students were going through the scheme in that year. The scheme, apart from providing students with a readily marketable skill (i.e., programming), has benefited both the participating colleges and the employers.

The employers came to see that their efforts in providing supervision could give an immediate return with the trainees undertaking useful projects for which the regular staff could never be spared. And they realized that, in the longer term, the industrial period was providing a prolonged selection interview. Some of the trainees after their first period of industrial training were absorbed by the participating employers as regular staff even before completing the full course.

Many colleges, which previously had only minimal computing staff and equipment, and little or no industrial contact, have now become a centre of computer training and a friend and a natural source of new recruits to a large number of data processing managers within their catchment area.

During the third year of the scheme, the Business Education Council (BEC) and the Technician Education Council (TEC) formed a joint committee which, starting with the Threshold syllabus, evolved a syllabus for a new BEC/TEC National Certificate in Computer Studies. The Threshold Scheme has now adopted this syllabus so as to conform to a nationally accepted standard.

All the costs of the scheme are borne by the Manpower Services Commission. In 1981/82 the net cost per trainee to the tax-payer amounted to £2200.

5.2.2 The NCSDCT Part-time Course in Software Technology

This course was developed and is provided by the National Centre for Software Development and Computing Techniques (NCSDCT) in Bombay. The course is principally aimed at retraining graduates already employed as programmers but who have not had a formal training in computing. The syllabus -- made up of individual, self-contained modules of duration two-months each -- emphasizes programming methodology and provides familiarity with widely used programming tools and common applications in engineering and management. Pascal is introduced first and is used as a working language. Other common languages such as FORTRAN and COBOL are also taught.

Some of the novel features of the course which have contributed to its success and popularity are the following.

1. Modularity of the course allows the students to complete the full set of modules at a pace that matches their other professional commitments. Each module is taught by a practising scientist who specializes in that topic.

2. The course is heavily self-study oriented. One or more books are prescribed for each module selected out of the best currently available texts in the international market. Each student is given a full set of books to cover all the modules he/she has registered for. The cost of the books is included in the fee collected from the students. Only 30°/o of the contact sessions (2 hours in the evening each week day) involve formal lectures which mostly introduce the students to the relevant concepts and motivate them to study further on their own. Apart from self-study, the modules are paced by assignments and tests. During the remaining 70°/o of the contact sessions consultants are available to students to assist them with their self-study and assignments. The assignments require

programming, and liberal computer time is made available to the students via a time-sharing computing environment.

3. Entrance to the course is through a 3-hour long objective test. Two introductory books on computers and computing are prescribed. The entrance test evaluates the applicant's ability to read and absorb the concepts introduced in the books; the test also evaluates the usual scholastic aptitudes of the applicants. The formal qualification prescribed is a bachelor's degree in any field but with an exposure to college-level mathematics.

4. The topics covered are: Introduction to programming using Pascal; Data Structures; Computer system organization and System programming; FORTRAN & COBOL; Data Base management techniques.

The part-time course is in its sixth year at present. Each year 65 students are admitted selected out of over 1000 who take the entrance test. About two-third complete all the modules. The fee for each student is, at present, Rs.2000/- ($200).

5.2.3 Training and Education in Computer Applications at the Asian Institute of Technology, Bangkok

The Regional Computer Centre (RCC) founded by the Asian Institute of Technology (AIT) currently operates an IBM 3031 model 6 with a variety of peripherals. The RCC was the outcome of a multigovernment/multiagency cooperation. Its main purpose is to provide computing services to AIT and also to provide education and training in computing to meet the needs of the region. RCC maintains a large and growing library of software including data base and information retrieval systems, mathematical, statistical, and simulation packages, planning aids, and application packages covering several important areas of computer usage.

RCC started its training programmes called 'Programmes in Computer Application Development (PCAD)' in 1976. PCAD provides tailor-made, practical instruction in the use of computer as a tool in problem-solving. There are three categories of programmes:

A. Training to adapt the use of specific software packages to develop applications to meet specific needs;

B. Training to use advanced application technologies to develop specific programmes for use when the trainees return to their countries;

C. General training in the development of computer applications.

Programmes A, B, can vary in length from several weeks to several months depending upon the application to be developed. Programme C is of 15 weeks duration with 8-9 weeks of formal instruction and 6-7 weeks of practice. The formal instruction in C is offered three times a year, in January, May and September. The contents vary according to the needs of the participants. The backgrounds of the participants span a wide spectrum -- from novices to very experienced computer users. As of December 1981, PCAD had trained about 200 participants from 13 countries.

AIT has now commenced a graduate programme in computer applications. This is run by the Division of Computer Applications (DCA). The programme concentrates on user-oriented education and on applied research, seeking practical ways in which computers can be utilized most effectively in the Asian environment.

RCC and DCA have pioneered the promotion of continuing education to computer personnel in the Asian region. Each year about 5 seminars, short courses, and conferences on various computer technology topics are organized.

5.2.4. International Education and Research for Application of Computer Technology (INTERACT)

It is estimated that during the 1980s about 1 % of the industrial investment in developing countries would be in dedicated computer-based systems for real-time and other applications (excluding mere data processing and data base applications). These dedicated systems would be needed to support vital development-catalyzing applications such as, communications, power system control and management, railway wagon management, weather monitoring and forecast, and so on. Currently most of these systems are supplied by a few industrialized countries on a turn-key basis often as 'black-boxes'. They are invariably inadequate for local needs and significant effort is required for adaptation, enhancement, etc., before they can be put to field use. In the absence of local know-how to carry out these adaptive modifications, this work too gets usually assigned to the vendor himself or to third party system engineering groups from developed countries at exorbitant costs.

Know-how needs to be developed among the developing countries themselves to undertake system engineering and software development for such real-time, on-line systems, to save close to 60 % of the foreign exchange cost and to become self-reliant in meeting internal needs. Recognizing this, a TCDC project INTERACT was started in 1981 in India with the support of UNFSSTD/UNDP. The identified applications for this project are: (1) power systems management, (2) railway freight management, and (3) meteorological early warning systems. Computer professionals from participating developing countries work in the areas of systems analysis, design, and system integration as co-professionals along with the project team. The completed systems are to be installed in the field and linked to the operational environments in the concerned sectors with the cooperation of the relevant Central/State government agencies.

Co-professionals who work on the project would become proficient to undertake similar assignments in their own countries. The design details of the specific prototype systems built in the project would be available for immediate adaptation and use in the participating developing countries. In addition, structured training programmes -- based on the field-implemented prototype systems -- are to be run for professionals from participating and other developing countries. The project funds would permit up to 200 professionals to be trained. Portable teaching material and training kits are to be developed so that the trainees, after returning home, could use these to train more of their country professionals. The training would emphasize system design of dedicated real-time computer systems, software-hardware interfacing, operating system design and development, system integration, testing, and so on. The training programme would cover both executives and field-level professionals. 11 co-professionals from 5 developing countries are participating in Project INTERACT. But some 18 developing countries have indicated their desire to send trainees to the training course.

6. SUMMARY OF INITIATIVES NEEDED

6.1 Preliminary Comments

Initiatives that need to be taken by developing countries and by UN organizations to strengthen technological capabilities of developing countries have been analysed, discussed, and formulated in a variety of intergovernmental and other forums. Among the UNIDO-sponsored efforts, perhaps the most comprehensive summaries of these formulations are to be found in the Dubrovnik Workshop Report referred to earlier in the Introduction, and in the document titled "Strengthening of Technological Capabilities of Developing Countries". Almost all the recommendations contained in these reports can be readily translated to apply to strengthening the capabilities of developing countries in the area of software technology also.

This merely emphasizes the fact that software technology and software engineering have all the ramifications that other technologies and other engineering industries have and, hence, the types of initiatives needed to strengthen a country's base in the latter would apply mutatis mutandis to the former also. However, as we discussed in the Introduction, software engineering differs from other conventional engineering activities in being design intensive and, therefore, very critically dependent on the availability of appropriately trained, high quality manpower resources. In the body of this report we have also discussed in some detail the necessity to analyse the priority needs of a country in the information sector and then to deploy indigenous software efforts to meet these needs. The initiatives that must be taken by developing countries to acquire self-reliance in software production, and by UN organizations to promote such self-reliance, directly follow from both these two considerations. We shall briefly summarize the needed initiatives in this section. (For further elaboration of some of the points made here see [UNIDO 1983b].)

6.2 Initiatives by Developing Countries

To quote from the UNIDO document, Strengthening of Technological Capabilities of Developing Countries, "A framework for national action is based on the three pillars of policies, programmes, and institutions. Policies by themselves can only act like valves which channel or shut off the flow of natural resources and energies. The creation of such natural resources and energies has to be effected through specific programmes of action with emphasis from policies giving the lead. Institutions are instruments of implementation and can only be as effective as the policies and programmes that back them, although over a period they can themselves help to generate policies and programmes. What needs to be avoided is an excessive reliance on any one of the three factors at the expense of the other two".

6.2.1 Policies

1. The first and the most important policy initiative needed is to recognize that 'information'

is an economic commodity and deal with raising the level of 'information consciousness' in all sectors of the economy as a task of vital importance to development.

2. All developing countries need a well-thoughtout and well-articulated informatics plan. This need arises from two reasons: (i) Since most developing countries have to depend on imports for making computers available, in the absence of a planned procurement policy, precious foreign exchange would be frittered away in the uncoordinated purchase of computers; (ii) Purchase and deployment of computers must be underpinned by well-planned actions for ensuring: (a) their proper maintenance and operation, (b) availability of right kinds of system and application software, and (c) the availability of suitably trained manpower to use the systems.

3. In working out a national informatics plan the most important decision criterion is the maximization of the value-added component at all levels. Public purchase policies should be deliberately formulated to support this principle. Turn-key solutions should be avoided to the maximum extent possible. Even if computer systems have to be imported, their operation and maintenance should be locally handled. Even if software packages are imported, their maintenance, modification and

augmentation should be locally handled. From the longterm viewpoint, it is advisable to invest in manpower training to first use computers knowledgeably and, later, to design and develop software locally to meet local needs.

4. Policy initiatives are needed to promote small- and medium-scale entrepreneur groups to provide software products, services, and consultancy. Provision of venture capital, wide-based access to computing resources, tax benefits, are some of the measures that could assist in such a promotional campaign.

5. As the indigenous capability to provide software products and services builds up it would become important to provide legal protection to these industrial activities through patents, copyright laws, etc. Also promotional measures would be needed to help to penetrate the export market.

6. To derive the maximum benefits out of information technology, the integration of computing and communication is of the utmost importance. Most developing countries have extremely poorly developed communication infrastructures. Investment in modernizing the communication base to facilitate data communication between computers, and between users and computers, should be taken up with the highest priority.

6.2.2 Programmes

1. At present in the developing countries informatics serves the research and academic groups and big business and organized industrial sector who already understand the value of informatics in their work. If informatics is to support national development it must concern itself with the small- and medium-scale. Thus special attention has to be paid not only to the small business entrepreneurs but also to the farmers. Together they provide employment for over 60% of the less sophisticated, lowly skilled and mainly illiterate: (see [Goldsmith 1982]). The importance of deploying computers to support basic-needs programmes and the agricultural sector has been discussed in detail in Sec.3.2. Programmes to deploy mini- and micro-computers in these areas and also to support the small-scale sector should be formulated and promoted.

2. Programmes to increase informatics awareness at all levels must be formulated and promoted. Vocational training in computer programming, training of computer scientists at the tertiary and post-tertiary educational levels, and extension-training in informatics to managers and personnel in the government sector, are important to provide. It is also vital to ensure that skilled manpower trained through these activities is suitably made use of.

3. Self-reliance in software production can be achieved only through creating opportunities for software generation for use within the country. Programmes to encourage indigenous software production should be formulated and promoted.

6.2.3 Institutions

1. Three kinds of institutions need to be promoted for achieving self-reliance in software production. These are:

i) Resource Centres in specialized software technology areas. These will function as Centres of Excellence in software technology generation and spin-off in the form of products and services.

ii) Education and Training Centres at the vocational, tertiary and post-tertiary levels. These will be the principal manpower generation centres.

iii) Computer Centres for government informatics and for supporting small entrepreneur groups engaged in software production and consultancy.

It is of great importance to provide suitable horizontal linkages between centres within groups and also across groups.

6.3 Initiatives by UN Organizations

1. Assistance from the UN organizations such as UNIDO should in general be catalytic - it should help the developing countries to help themselves. Assistance to achieve self-reliance is the most important and also the most difficult to provide. The exact nature of this assistance would naturally depend on the current informatics level of the country and the transition it is seeking to achieve (i.e., initial - basic, basic-operational, operational - advanced). The assistance could span a wide spectrum - from assistance to formulate a viable national informatics plan, to assistance in structuring and operating a highly specialized software technology resource centre (e.g. for CAD-CAM applications).

2. Promoting regional cooperation among the developing countries is very valuable. In software training, and in the production of specialized software for development-catalyzing applications, there is considerable scope for regional collaboration. UNIDO and other concerned UN agencies could play an especially valuable role in creating frameworks for such cooperation - e.g., through networking of resource groups, creating regional software training and/or production centres, etc. Programmes like Project INTERACT (see Sec. 5.1.4) should be promoted on a wider scale.

3. Identification of successful institutional models which could be transported from one country to another is of very great value. UN agencies could facilitate this by creating forums for exchange of experience among developing countries. Exchange of resource persons among developing countries should also have considerable value in spinning off ideas, products, and services. Also, compilation of case studies should assist developing countries to learn from one another in choosing strategies for their own informatics industries.

4. Software is increasingly being treated as a strategic commodity and software export from many developed countries is beginning to be covered by strict export control regulations. Control over transfer of technology in this field is making it increasingly difficult for developing countries to acquire this technology for deployment even in legitimate development-catalyzing application areas. It is important for UN agencies to create impartial alternate channels for ensuring continued availability of expertise, advice, consultancy, and design know-how from developed to developing countries.

REFERENCES

1. Datamation (1980): "Picking and Perfecting the Packages"; December, pp.139-148.
2. EDINFO-82 (1982): An International Symposium on Education in Informatics, Extended Abstracts published by The Computer Society of India.
3. F.G. Foster (1982) Ed.: Informatics and Industrial Development, Proceedings of an International Conference on Policies for Information Processing for Developing Countries, 9-13 March, 1981, Trinity College, Dublin; Tycooly International Publ.Ltd., Dublin.
4. M. Goldsmith (1982): "Thoughts on Informatics and Small and Medium Scale Enterprises in the Third World"; in [3], pp. 388-391.
5. R.E. Kalman & J. Martinez (Eds.)(1982): Computer Applications in Food Production & Agricultural Engineering, Proc. Conference held in Cuba, 26-30 Oct., 1981; North-Holland.
6. B. Kernighan & P. Plauger (1976): Software Tools; Addison Wesley.
7. R. Narasimhan (1982): "The Socio-Economic Significance of Information Technology for Developing Countries"; in [3], pp. 27-38.
8. G. Penney (1982): "The Threshold Scheme"; in [2], pp. 134-137.

9. D.J. Reiffer & S. Trattner (1977): "A Glossary of Software Tools and Techniques"; Computer, IFEE Comp.Soc.Magazine, 9, pp. 52-60.

10. F. Robson (1982): "Advanced Technology in Developing Agricultural Economics"; in [5], pp. 267-282.

11. UNFSCO (1980): Informatics : A Vital Factor in Development.

12. UNIDO (1983a): Workshop on Institutional and Structural Responses of Developing Countries to Technological Advances, Dubrovnik, 31 May--4 June,1983.

13. UNIDO (1983b): Problems of Software Development in Developing Countries.

14. United Nations (1971): The Application of Computer Technology for Development.

15. WCCE (1981): Computers in Education, Lausanne Conf. Proc. published by North-Holland, 1982.

16. L. Welke (1980): "The Origins of Software"; Datamation, December, pp. 127-130.

17. F.G. Withington (1980): "The Golden Age of Packaged Software"; Datamation, December, 131-134.

18. World Communications : New Horizons, New Power, New Hope (1983); Le Monde Economique, International Publishers.

B-389

4.04.16

AD.85.03