THE BLACKWELL ENCYCLOPEDIA OF MANAGEMENT

About the Editors

Cary L. Cooper is Professor of Organizational Psychology at the Manchester School of Management (UMIST), UK. He has also been appointed Pro-Vice-Chancellor at the University of Manchester Institute of Science and Technology (UMIST). He is the author of over 80 books, has written over 250 scholarly articles and is editor of *The Journal of Organizational Behavior*. He is also the Founding President of the British Academy of Management.

Chris Argyris is the James B. Conant Professor of Education and Organizational Behavior at the Graduate School of Business, Harvard University. He has written many books and received numerous awards, including the Irwin Award by the Academy of Management for lifetime contributions to the disciplines of management. Recently, the Chris Argyris Chair in Social Psychology of Organizations has been established at Yale University.

About the Volume Editor

Gordon B. Davis is the Honeywell Professor of Management Information Systems at the Carlson School of Management, University of Minnesota. He is recognized internationally as one of the principal founders and intellectual architects of the academic field of Management Information Systems. He has authored 21 books and over 150 scholarly articles. He has been involved in establishing academic journals, scholarly conferences, and model curricula for the field of information systems.

The Blackwell Encyclopedic Dictionary of Management Information Systems

Edited by Gordon B. Davis

Carlson School of Management

BLACKWELL
Business

— C —

C and C++ C is a high-level programming language that takes into account machine-level considerations. There are a number of high level instructions to simplify programming. It has been used in writing UNIX operating systems and for applications to be run on microcomputers and workstations. Programs in C for one computer tend to be easily transported to another computer.

C++ is a variation of the C language that incorporates principles of object-oriented programming. It has the advantage of being similar to the C language but the disadvantage of not meeting all of the conditions of an object-oriented programming language. *See also* PROGRAMMING LANGUAGES.

careers in information systems Information systems (IS) professionals are people who acquire, develop, manage, and maintain hardware, software, and telecommunication networks, and offer computing and information services to users. There is a wide spectrum of information systems professionals who play various roles in the management and development of information systems. The major categories of IS careers are:

1. *Chief information officers* (CIOs). CHIEF INFORMATION OFFICERS are top corporate officers responsible for the overall IS functions in organizations. CIOs offer leadership in managing the information resources of the firm, and in directing the power of information technology toward the strategic objectives of the firm.
2. *Data center IS professionals*. These include computer operations managers; database administrators; network specialists; systems programmers; and computer operators who are involved in computer capacity planning and management; disaster recovery; hardware and systems software maintenance; and production or job scheduling.
3. *Software development and maintenance teams*. These comprise application programmers who develop software using programming languages and software tools; systems analysts who determine user requirements and design the systems specifications; and project managers who oversee and coordinate teams of programmers and systems analysts in developing specific application systems.
4. *End-user IS professionals*. These are responsible for the acquisition, maintenance, and help-desk support of personal computer use in the organization.

Career Orientations of IS Professionals

Igbaria et al. (1991) found that IS professionals are diverse in their career orientations, i.e. in their interests, self-perceived talents, values, and motives that shape their career decisions (see also Ginzberg & Baroudi, 1988). Technical and managerial orientations are two dominant themes among IS professionals. People in technical jobs such as systems programmers, application programmers, and systems engineers are more technically oriented, while those in managerial jobs such as systems analysts, project leaders, and managers are more managerially oriented. The study by Igbaria et al. (1991) found that the match between job type and career orientation of IS professionals is important because such a match leads to higher job satisfaction, stronger organizational commitment and lower intentions to leave the organization. The implication of this study is that management should take into account differences in employee interests and orientations and provide job opportunities that match employee needs.

Future Directions for IS Professionals

Two important factors influence the evolution of IS work: the expansion of end-user computing and the outsourcing of IS services. With the advent of client-server technologies, powerful personal computers, and software languages that are easy to use, end-users have been increasingly developing and managing their own computing applications, rather than relying on the organization's IS employees. This trend has contributed to the downsizing of the IS group within organizations. It will also affect the nature of IS work. Rather than providing complete information system services to the organization, IS professionals may provide support for end-user developed systems or may concentrate on developing and managing only major, company-wide applications.

The organization of IS work is also changing. Traditionally, firms that require a specific skills set will employ a worker under a long-term employment contract where the worker works all year round at the employer's place of business, except for vacations and holidays. Unless they resign or have their services terminated, employees are assumed to remain with the employer until death or retirement. From the legal perspective, both parties have rights and responsibilities accorded to them by both common law and employment statutes governing the employer–employee relationship. Accordingly, it is not uncommon for IS workers to remain attached to single employers during their entire careers.

Careers built upon long-term employment relationships with single organizations work well in situations where the skills sets required by the firm are relatively stable over time. However, in information systems, firms are finding that alternative employment arrangements, such as contract work, are becoming more important and attractive because of the increasingly rapid evolution of technology (see Slaughter & Ang, 1995). Cutting-edge technologies typically enjoy lifespans of only two years. Skills of IS personnel therefore erode very rapidly. Operating in short windows of stable technological environments, IS organizations with a stable and static workforce anchored in traditional employment relationships continually face the problem of needing to upgrade the skills of the workforce. In many cases, organizations may feel that commitment to training the internal workforce is self-defeating. Because technologies move so rapidly, by the time an organization invests in and trains its IS staff in a certain technology, that technology may already have become obsolete.

Accordingly, the number of organizations using contract workers for IS work is growing dramatically, particularly with the rapid diffusion of IS outsourcing where organizations are contracting out the services of the entire IS department to independent contractors or service-providers. Consequently, IS careers no longer take place in single organizations. Rather, as contract workers, IS professionals are not attached to any single organization for a long period of time. Instead, they are independent and self-employed, hired on a fixed-term basis for a specific skill through an agreed-upon contract. The contract may provide a fixed duration of service or may operate on a job-by-job basis. From the worker's point of view, contract work provides an opportunity to establish a special expertise or professional status within an industry. In fact, it is often regarded as a way for workers to focus on the aspects of their profession they most enjoy (e.g. programming instead of managing software projects) without having to deal with corporate politics or pressures to move up the expected career ladder.

The trends toward outsourcing and careers based on contract work arrangements imply an increasing interorganizational division of IS labor in the future, as work formerly conducted within organizational boundaries and under the administrative control of a single enterprise is parceled out to more specialized individuals or organizational entities. The implication for IS professionals is that they can no longer solely rely on building careers by moving upwards in single organizations. Rather, IS professionals must consciously plan to upgrade and reskill themselves in light of competence-destroying technologies. They must also be cognizant of new career opportunities offered by outsourcing arrangements. For example, ideal IS professionals in outsourcing must possess not only a combination of technical and practical knowledge, skills, and abilities, but also negotiation and bargaining skills to sustain a flexible partnership that demands intense relationship-building and continual recommitment from top to bottom of both client organizations and service-providers.

Bibliography

Ginzberg, M. H. & Baroudi, J. J. (1988). MIS careers – a theoretical perspective. *Communications of the ACM*, 31 (5), 586–94.

Igbaria, M., Greenhaus, J. H. & Parasuraman, S. (1991). Career orientations of MIS employees: an empirical analysis. *MIS Quarterly*, 15 (2), 151–70.

Slaughter, S. A. & Ang, S. (1995). Information systems employment structures in the USA and Singapore: a cross-cultural comparison. *Information Technology and People*, 8 (2), 17–36.

Soon Ang and Sandra Slaughter

CASE: computer-aided software/system engineering The term "computer-aided software or system engineering" (CASE) applies to tools that assist developers to analyze, design, and construct software applications or information systems. A software system is developed to provide a solution to a problem. This requires the problem domain to be analyzed before a solution is proposed and a solution to be designed before a software system is constructed. Once constructed, the system is maintained. After use, it may evolve and change. This software system life-cycle is not necessarily sequential, nor are all phases in the cycle exercised every time. There are different approaches to the development (analysis, design, and construction) of software systems, as well as different formalisms and notations to represent requirements and design during software life-cycle phase.

There are two broad categories of CASE tools – upper and lower. Upper CASE tools are used in the analysis and design, while lower CASE tools are used in the construction of information systems. A repository is used to pass analysis and design representations, often called models, to a lower CASE tool. A lower CASE tool is also called an application or code generator. It has a language to add more implementation details to a design and provides a facility to generate program instruction. All artifacts in the development process are stored in a repository. In this way, they can be used to maintain and evolve a software system from the analysis and design perspective, providing higher-quality software. In addition, they can be reused in other projects to increase software development productivity.

A CASE tool is, itself, a data-intensive software system. It consists of a repository of analysis, design, and construction data about another software system, and a graphical interface that looks at the repository data through the eyes of a specific notation (formalism or language). The primary concept that distinguishes a CASE tool from a Database management system is its meta-model. A meta-model is a repository schema, which determines what kind of data is going to be kept in the repository about systems which are built with the help of the CASE tool.

One of the important features of CASE tools is support for modeling of a problem domain and modeling information system components. One of the most-used notations to analyze a problem domain is a data flow diagram (DFD). DFD has four concepts with graphical representations: external entities, business processes, data flows, and data stores. External entities provide specific entry and exit points to or from an organization unit being analyzed. Business processes transform, move, or store organizational data represented as data flows or data stores. A process relies on organizational resources to accomplish its tasks. If a process at hand is complex, it can be decomposed into lower-level processes that are graphically represented in another diagram. This top-down graphical decomposition of processes is a significant feature of the DFD formalism.

In an information system application, there are four major components: business data, user applications (interfaces), business policies (rules), and business tasks (services). In general, each component is represented and designed using a different formalism. A CASE tool may support only one of them, or an integrated combination of them. In object-oriented analysis and design tools, the components are integrated through a common model of classes (entities) and their relationships.