CURRENT PRACTICES OF INFORMATION SYSTEMS DEVELOPMENT TECHNOLOGY

Nancy Tsai[1], Russell K.H. Ching[2]

[1]California State University, Sacramento, College of Business Administration (sytsai@csus.edu)
[2]California State University, Sacramento, College of Business Administration (chingr@csus.edu)

Abstract

During the latter half of the millennium, many business organizations focused their efforts on applying information technology (IT) toward the development and implementation of different types of information systems (IS) to facilitate their business. The advent of newer technologies has led to the introduction of systems that possess expanded capabilities, incorporate more complex levels of IT, and open greater business opportunities. This paper examines current IS development practices, and presents the results of a national survey on current IS development practices. The results of this survey indicate that the major objectives of software development methods and tools focus on developing information systems that correctly met (with the fewest errors) the requirements of end users, and can be easily modified and maintained later. Therefore, many systems development practitioners will adopt methods and tools that help them meet these objectives while they examine IT for innovative ways that might provide a competitive or cost-saving advantage.

1. Introduction

During the latter half of the millennium, many business organizations focused their efforts on applying information technology (IT) toward the development and implementation of different types of information systems (IS) to facilitate their business. The advent of newer technologies has led to the introduction of systems that possess expanded capabilities, incorporate more complex levels of IT, and open greater business opportunities. Generally, these systems are grouped according to their progression into five categories: transaction processing systems (TPS), management information systems (MIS), decision support systems (DSS), expert and knowledge-based systems, and electronic commerce (e-commerce). The development of each category has presented different sets of problems, particularly those associated with harnessing increasing IT complexity and at the same time targeting windows of opportunities. This paper examines current IS development practices in light of these problems.

Whereas in past system development followed a regimented and methodological approach, systems of today must seize upon immediate business opportunities or face early obsolescence. In many cases, the continuous shifts in opportunities require systems to be completed within six or fewer months, any more and the opportunity will have passed. This has placed greater pressures on IT professionals for achieving the goals and objectives of the systems. Hence, the objectives of today's system development present an interesting dilemma that can be described as a double-edged sword: systems must be developed quickly yet thoroughly to accommodate and satisfy the organization's ever changing business needs.

2. Information Systems Background

For the last five decades, business organizations have been utilizing IT to implement different types of information systems to facilitate their business operations. Transaction processing systems (TPS) comprise the first type. The primary focus of these systems is to automate the manual and routine transaction processing tasks that are often characterized by simple yet procedural-based computations and the repetitive processing of voluminous inputs. The outputs tend to focus on detail and be directed toward operational-level managers. TPS implementations often target reductions in labor costs/content and increased efficiencies of business operations.

The second type is named management information systems (MIS). In contrast to TPS, MIS perform statistical analysis on the data collected and processed by the organization's TPS. MIS gear their output toward satisfying the information needs of middle management, particularly those engaged in monitoring, controlling and planning the immediate and structured short-term business activities aimed at improving operational efficiency, effectiveness, and productivity.

Decision support systems (DSS), the third type of information system, are characterized by their ability to support computer-aided problem solving in an interactive computing environment [1]. They are intended to help solve complex, unstructured and nonroutine business problems faced by upper-level managers in the organization. DSS allow decision-makers to analyze corporate databases with sensitivity analysis (i.e., what-if, goal seeking) and forecasting models to explore their various options when engaged in ad hoc and unusual decisions, and to gain strategic advantages and a competitive edge. Thus, they help identify business opportunities.

Expert and knowledge-base systems comprise the fourth type of information systems. These systems are designed to mimic human intelligence through artificial intelligence, and by retaining the knowledge and decision rules of problem domain experts. The interactions between the non-expert user, knowledge base and inference engine, allow the organization to achieve higher quality and consistent decisions.

Electronic commerce (e-commerce) falls into the latest type of information systems. These systems include several subcategories, such as electronic funds transfer, electronic data interchange and Web commerce. A common characteristic of e-commerce systems is their support of business transactions, such as marketing, selling and buying products and services, taking place via private Extranet and/or public global Internet. Therefore, e-commerce systems offer organizations the opportunity to replace many of their existing business practices with the newer ways of conducting business in hopes of gaining a competitive edge [10].

3. Information System Development

IT has created a new world for organizations to carry out their business. The continuous development of less expensive but more powerful computers has enabled many organizations, to implement different types of information systems to increase its productivity, efficiency and competitive power. Moreover, the birth of the information superhighway has forced organizations to continually examine and reengineer their business processes to benefit from innovative uses of information technology, and to meet and exceed new business requirement(s) in this fast moving environment between the businesses. The shorten timeframe under which these systems must be completed (usually between three to six months) leads to a developer's paradox, a double-edge sword: *quick but thorough* development.

Although the development and implementation of an information system usually poses its own set of (unique) problems, many user and management complaints dwell on a common set of problems. Many experienced managers have cited the following as their most frequently encountered problems:

1. The final system did not adequately satisfy end-user's expectation and/or need.
2. The completion date of the final system exceeded the agreed dead line.
3. The total costs of the final system were much higher than the estimated budget.
4. The benefits generated from the final systems did not justify the total investment.
5. The inflexibility of the final system to meet new government regulation and requirements of end-user resulted high maintenance costs.

These problems have led information system researchers and data processing professionals to seek better methods and techniques to not only overcome the problems, but to *ensure the right information system delivers the right information to the right people at the right time*. Current system development technologies include a collection of traditional and/or newly developed structured and non-structured approaches and tools. These technologies have been used in industry to make the systems development process more efficient and effective. For this reason, many of these technologies have been included in undergraduate and graduate information systems. However, the fast pace at which business now moves has rendered some these methods and techniques obsolete. Thus, it is important to identify the current practices of industry to help ensure the currency of topic coverage.

4. A Survey of Development Practices

4.1 Methodology

A four-page questionnaire was developed for this study. Its primary focus was to identify the current use of information systems development approaches and tools, and the benefits reaped from automated development tools. The questionnaire also gathered information about the organizations, such as the methods they relied upon or adopted for developing application software packages and the minimum educational background they sought in new application development personnel.

The names of the organizations and executive/managers drawn from the Directory of Top Computer Executives for 1997. The two selection, organizations with 50 or more information systems employees and organizations with major investments in computing resources, narrowed the list of survey recipients to large businesses, non-profit organizations and government agencies. Over 1,100 questionnaires were mailed to organizations throughout the U.S. A cover letter explaining the survey's purpose and appealing for their participation has accompanied each questionnaire. Only 48 questionnaires were returned, of which 46were used in this study.

4.2 Results of the Survey

4.2.1 Development Methods

This section of the survey attempted to identify the direction different information systems development practices have taken over the past decade, and any emerging trends. Table 1 summarizes the survey responses to the percentage of software applications developed by an organization using one of the listed methods. In-house development appears to be the foremost method used. This is not surprising since the surveyed organizations have committed large investments toward their computing resources. The second most frequently used method involves acquired software packages. Organizations can now purchase several *read-to-use* or *off-the-shelf* packages that perform fairly standardized routine operations, such as payroll, accounts payable and accounts receivable. This allows them to direct more of their resources toward developing applications that target satisfying specific or unique information needs. Approximately nineteen percent of systems are developed through outsourcing. In many cases this allows the organization to capture the expertise it does not currently possess within its ranks to either customize acquired software packages and/or quickly develop systems that may provide a competitive advantage.

Methods	***Percentage (Mean/Median)***
In-House	48.1/45
Outsourcing	19.5/10
Acquiring Package	38.4/27.5
Other (Please Describe):	7.0/10

Table 1. Estimated percentage of application software development performed using one of the methods

4.2.2 Minimum Education for New Hired Systems Development Personnel

The survey sought to identify the most frequently applied selection criteria for hiring IT/IS application development candidates, particularly their education background and relevant experience. Table 2 provides a summary of the results for programmers, technical support personnel and systems analysts.

More than two-thirds of the respondents indicated that the relevant experience of a candidate is the most important factor they considered across the third categories. This result implies that an educational degree is not a substitute for practical experience in the field. This was particularly true for technical support personnel where more than 82 percent of the respondents placed relevant experience as their top criterion. Often, experience requires a person to apply his/her knowledge to different situations. As the person gains experience, s/he expands her/his mental library of solutions, induces generalizations, and forms rules that guide her/him to quickly create solutions to resolve problems. Experience also means reduced training costs and acquiring expertise that can be used immediately.

For candidates lacking experience, most organizations indicated a preference for MIS and CIS baccalaureate graduates when placing them in programming (20 percent) and systems analysts (22.7 percent) positions. In contrast, educational background was not considered a major criterion for selecting technical support personnel. Apparently, advanced degrees (i.e., masters) where not considered significant indicators in any of the three categories. It is obvious that employees with advanced degrees should be assigned more challenging responsibilities than constructing information systems.

A desire for MIS and CIS graduates over computer science graduates suggests that most IT/IS personnel should possess some knowledge of business functions. Whereas computer science graduates have a more intricate understanding of the inner workings of technology, MIS and CIS graduates can produce business solutions, a task that requires knowing *how* to apply IT as a means for overcoming an organizational problem.

Requirement	***Programmer*** n = 45	***Technical Support*** (e.g., telecommunications) n = 46	***Systems Analyst*** n = 44
Relevant Experience	**30 (66.7%)**	**38 (82.6%)**	**30 (68.2%**)
Bachelor's Degree, MIS/CIS Major	9 (20.0%)	3 (6.5%)	10 (22.7%)
Bachelor's Degree, Computer Science	4 (8.9%)	3 (6.5%)	2 (4.5%)
Master's Degree, MIS/CIS Concentration	0	1 (2.2%)	1 (2.3%0
Master's Degree, Computer Science	0	0	0
Other Bachelor's Degree (Please Specify)	1 (2.2%)	1 (2.2%)	1 (2.3%)
Other Master's Degree (Please Specify):	0	0	0
Other (Please Describe):	1 (2.2%)	0	0

Table 2. Minimum education requirement of new application development personnel

Overall, organizations are focused on hiring candidates for systems development positions with previous relevant experience. Educational background preferences are made to those holding a bachelor's degree in the areas of MIS and CIS. Therefore, in the future, personnel assigned to information systems development, maintenance and technical support will need to take useful courses or possess a degree in the MIS and CIS disciplines. The dramatically increasing college enrollment in IT/IS programs during recent years supports this trend.

4.2.3 Current Systems Development Approaches

The survey also attempted to determine which of the development approaches were being widely used by organizations. Table 3 summarizes the responses. It appears that many organizations are committed to developing systems through prototyping (34) and the systems development life cycle (30). It is not surprising that prototyping is the most popular systems development approach. Systems analysts working with end-users can quickly build, refine and modify a prototype using fourth generation programming languages. Therefore, prototyping is not only a cost-effective method but a means for producing an end product that is acceptable to end-users. Surprisingly, the traditional systems development approach, system development life cycle was ranked second. This implies that many organizations still prefer to divide the systems development process into predefined phases to effectively control the limited resource and manage specific activities within each phase.

Rapid application development (RAD) and structured development appear to be moderately used. RAD segments a system into subsystems or functional components where *model-critique-refine* processes are performed. Developers use an integrated software package to enhance and extend the initial version until it is suitable for operational use. This is another cost-effective approach since it creates applications that are easier to maintain and modify. The structured development approach emphasizes top down analysis, design, programming, and implementation. It breaks an information system into components, or modules to enhance the understanding of analysts, designers and end-users. Hence, it is a valuable communication and documentation methodology in developing an information system [2].

Two newer information systems development approaches, end-user development and object-oriented analysis/design, are used in half of the responding organizations. The factors contributing to their growing popularity in end-user computing include inexpensive but powerful microcomputers, the availability of systems development software with graphical user interface, and computer literate employees in the business functional areas. Moreover, end-users can develop their own applications faster and more accurately since they understand their business processes and needs better than IT/IS staff. This form of development will inevitably become pervasive, especially for small application projects. Object-oriented (OO) analysis/design treats a system as a set of cooperating objects with encapsulated data and methods. The interaction between objects is done through messages. OO supports temporal and spatial changes of the objects, and is more suitable for the design of network applications that need multimedia presentation and on line transaction processing, and involve document management [3,7]. This approach will be increasingly popular due to its suitability for building e-commerce applications. Its reusability feature helps reduce development cost and increase productivity.

Only nine responding organizations reported using the data centered approach. With this approach, the enterprise's data are treated as the most important resource and form the foundation for information systems. Its low usage might be due to its time consuming nature and the difficulty involved with managing a single database for the entire organization.

Approach	***Check If Used***
System Development Life Cycle (SDLC)	30
Rapid Application Development	24
Structured Development	22
Prototyping	34
Object-Oriented Analysis/Design	14
End-User Development	15
Data Centered Approach	9
Other (Please Specify)	3

Table 3. System development approach

4.2.4 System Development Tools

Another section of the survey asked respondents to identify the system development tools that were most often used. The frequency of their use ranges from never (used) to use often. Table 4 summarizes the responses.

Data dictionary, data flow diagram (DFD), entity-relationship diagrams (ERD) and system flow charts appear to be the most frequently used systems development tools. A data dictionary defines every data and its structure within the scope of the proposed system. While a DFD shows the passage and process of data from sources to destinations, a system flowchart identifies relationships between input, output, and processing components [2]. ERDs describe the structure of data and their relationships in a database [4]. These four system development tools are used for either decomposing a complex system into simple and workable related components, or defining and documenting the entire information system. Their usage is highly associated with system development life cycle approach that was rated as the second most popular development approach.

The hierarchy chart, decision table and structure chart cited as used sometimes. A hierarchy chart specifies the architecture and functions of an application program without making note its logic or the exchange of data between its components [9]. In contrast, a structure chart defines an application's basic processing components and their relationship, including data flowing between its components [9]. A decision table gives a tabular view of the decision-making logic associated with complex combinations of conditions and actions in a program [8]. It is interesting to note that these three development tools are mostly used to build computer programs, either to diagram overall structures or to show the conditions, logic, and actions.

State transition diagrams, dialogue design diagrams, structured English, and action diagrams are the four least used by the responding organizations. A state transition diagram illustrates all possible changes of a state for a complex processing while a dialogue design diagram depicts different transition states between screens and response actions. Structure English uses narrative notation of a natural language to present program logic [8]. Action diagrams define detailed program logic of a process with short English phrases and simple graphic notations [8]. The results suggest that these classic systems analysis, design and programming tools have been replaced by newer ones.

Tool	*Frequency Of Use*		
	Never	*Sometimes*	*Often*
Action Diagram (n = 37)	**16 (43.2%)**	14 (37.8%)	7 (18.9%)
Data Dictionary (n = 42)	5 (11.9%)	12 (28.6%)	**25 (59.5%)**
Data Flow Diagrams (DFD) (n = 42)	5 (11.9%)	13 (31.0%)	**24 (57.1%)**
Decision Table (n = 37)	10 (27.0%)	**20 (54.1%)**	7 (18.9%)
Dialogue Design Diagram (n = 32)	**21 (60.0%)**	6 (18.8%)	5 (15.6%)
Entity-Relationship (ER) (n = 43)	4 (9.3%)	16 (37.2%)	**23 (53.5%)**
Hierarchy Chart (n = 38)	7 (18.4%)	**24 (63.2%)**	7 (18.4%)
Stage Transition Diagram (n = 32)	**24 (75.0%)**	5 (15.6%)	3 (9.4%)
Structure Chart (n = 38)	10 (26.3%)	**19 (50.0%)**	9 (23.7%)
Structured English (n = 33)	15 (45.5%)	**12 (36.4%)**	6 (18.2%)
System Flow Chart (n = 41)	3 (7.3%)	16 (39.0%)	**22 (53.7%)**
Other (n = 2)	0	0	**2 (100%)**

Table 4. System Development Tools

4.2.5 Benefits of Current System Development Approaches

Lastly, the survey attempted to gather information about the improvements and benefits organizations were reaping from their current systems development approaches and tools. Table 5 lists several benefits and the responses to them. It appears every organization has experienced some improvements.

These findings suggest that all organizations through the selection of their development methods and/or tools have been able to (1) significantly reduce the errors in their systems, (2) better meet their user requirements, and (3) reasonably communicate the system's design to users. These results should please researchers, developers and practitioners since they indicate we have been able to overcome a major obstacle, poor communication between end users, analysts and designers during the definition of the systems requirements.

Other benefits that were cited include systems were developed on time, programming time has been reduced, systems are relatively easy to maintain, systems are easy to modify, communications are satisfactory with management, and communications are satisfactory with technical personal. These findings should also be encouraging to the champions of these approaches and tools.

To a lesser extent, some organizations indicated that they were able to moderately achieve greater consistency and completeness among their (development) documents. It is not surprising that more organizations did not realize greater improvement in this area for the following reasons. First, most software development personnel do not have time to complete an up-to-date document. Secondly, this is not a primary objective of the methods or tools the organization has selected. However, complete and consistent documentation is very important, and not only affects a system's future operations, maintenance, modification and recovery, but serves as a valuable tool for training and educating software personnel in reducing the impact of key personnel turnover. Therefore, organizations should assign a technical writer to oversee the preparation, management and control of all documents.

Results Being Achieved	Mean	***Not At All***				***Very Well***
User Requirements Are Being Met (n = 45)	3.71	0	3 (6.7%)	13 (28.9%)	**23 (51.1%)**	6 (13.3%)
Systems Are Developed On Time (n = 45)	3.13	0	10 (22.2%)	**21 (46.7%)**	12 (26.7%)	2 (4.4%)
Systems Are Developed Within Budget (n = 45)	3.38	1 (2.2%)	5 (11.1%)	**19 (42.2%)**	16 (35.6%)	4 (8.9%)
Programming Time Has Been Reduced (n = 44)	2.98	5 (11.4%)	10 (22.7%)	**13 (29.5%)**	**13 (29.5%)**	3 (6.8%)
Systems Are Relatively Error Free (n = 45)	3.40	1 (2.2%)	7 (15.6%)	11 (24.4%)	**25 (55.6%)**	1 (2.2%)
Systems Are Relatively Easy To Maintain (n = 45)	3.27	1 (2.2%)	8 (17.8%)	16 (35.6%)	**18 (40.0%)**	2 (4.4%)
Systems Are Relatively Easy To Modify (n = 45)	3.31	1 (2.2%)	7 (15.6%)	**17 (37.8%)**	**17 (37.8%)**	3 (6.7%)
Documentation Is Consistent and Complete (n = 45)	2.40	6 (13.3%)	**18 (40.0%)**	**18 (40.0%)**	3 (6.7%)	0
Communications Are Satisfactory With: Management (n = 44)	3.41	0	4 (9.1%)	**21 (47.7%)**	16 (36.4%)	3 (6.8%)
Users (n = 44)	3.55	0	3 (6.8%)	16 (36.4%)	**23 (52.3%)**	2 (4.5%)
Technical Personnel (n = 44)	3.55	0	4 (9.1%)	16 (36.4%)	**20 (45.5%)**	4 (9.1%)
Other (n = 2)	4.0	0	0	0	2	0

Table 5. Benefits of Current System Development Approach

5. Conclusion

The results of this survey indicate that the major objectives of software development technology focus on developing information systems that correctly met (with the fewest errors) the requirements of end users, and could be easily modified and maintained later. Therefore, many systems development practitioners will adopt methods and tools that help them meet these objectives while they examine IT for innovative ways that might provide a competitive or cost-saving advantage.

There are several significant implications of the findings from this study for college level MIS and CIS programs. A curriculum should not only *instruct* students on the use of system development methods and tools that have been commonly adopted by businesses, it should emphasize their importance in the context of the objectives. With this in mind, students should understand the effects of each (tool and approach) on the management of a development project, and when its use is most appropriate.

Further research needs to be conduct to determine how the IT/IS development process can be improved, particularly with e-commerce applications and systems, and how end-user computing can be effectively supported in the new millennium. Future surveys need to be continuously carried out in order to assess the currency and appropriateness of IS curricula to the professional practice of software development in business.

References

[1] Bidgoli, *Decision Support Systems: Principles & Practice*, South-Western Publishing Co., 1996.

[2] Burch and Grudnitshi, *Information Systems: Theory and Practice*, John Wiley & Sons, 1986.

[3] Coad and Yourdon, *Object-Oriented Analysis*, Prentice Hall, 1991.

[4] Connolly, Begg, and Strachan, *Database Systems: A Practical Approach to Design, Implementation and Management*, Addiso-Wesley, 1999.

[5] Dewitz, *Systems Analysis and Design and the Transition to Objects*, McGraw-Hill, 1996.

[6] Hershey and Kizzier, *Planning and Implementing End-User Information Systems: Office and End-User Systems Management*, South-Western Publishing Co., 1992.

[7] Lorenz, *Object-Oriented Software Development: A Practice Guide*, Prentice Hall, 1993.

[8] Martin, *Recommended Diagramming Standards for Analyst & Programmers: A Basis for Automation*, Prentice Hall, 1987.

[9] Martin and McClure, *Structured Techniques: The Basis for CASE*, Prentice Hall, 1988.

[10] Schneider and Perry, *Electronic Commerce*, Course Technology, 2000.