Association for Information Systems

AIS Electronic Library (AISeL)

AMCIS 1996 Proceedings

Americas Conference on Information Systems (AMCIS)

8-16-1996

Dimensions of Computer Aided Software Engineering (CASE) Technology

Srinarayan Sharma
Southern Illinois University

Arun Rai
Southern Illinois University

Dimensions of ComputerAided Software Engineering (CASE) Technology

Srinarayan Sharma, Southern Illinois University, Carbondale, IL 62901
Arun Rai, Southern Illinois University, Carbondale, IL 62901

Introduction

Computeraided software engineering (CASE), a relatively recent technological innovation, is viewed by both researchers and practitioners as a potential means to increase the productivity of information systems development activities and ease the software development and maintenance burden threatening to overwhelm ISDs. A number of studies have examined the effect of CASE usage on systems work. While some studies have reported productivity gains (or perception of such gains) from the use of CASE tools (Banker and Kauffman, 1991; Norman and Nunamaker, 1988; Swanson, et al., 1991), many others have found that the expected productivity gains are elusive (Card, et al., 1987; Yellen, 1990), or marred by inadequate training and experience, developer resistance, and increased design and testing time (Norman, et al., 1989; Orlikowski, 1988, 1989; Vessey, et al., 1992).

These contradictory experiences with CASE technology appear puzzling and difficult to interpret. Many reasons have been put forth to account for these inconsistencies. Most important of these seems the lack of a clear definition of CASE usage (Henderson and Cooprider, 1990).

The need to better define and measure CASE usage behavior suggests a need to develop a model of CASE that corresponds more closely to key designer behavior. That is, rather than define this technology in economic terms (e.g., costs), technology terms (e.g., PCbased or networked), or other more general terms (e.g., having an embedded design language or structured code compiler), we must develop a model of CASE technology that is functionality oriented. Such a model would then provide a way to directly relate usage of a CASE tool to design team performance.

Henderson and Cooprider (1990) have developed such a model for IS planning and design aids in general. According to this model, design aid technology has three general dimensions: production, coordination, and organization. Production dimension is defined as functionality that directly impacts the capacity of an individual(s) to generate planning or design decisions and subsequent artifacts or products. Coordination technology is defined as functionality that enables or supports the interactions of multiple agents in the execution of a planning or design task. Organizational dimension is defined as functionality and associated policy or procedures that determine the environment in which production and coordination dimensions will be applied to the planning and design process. Each of these subdimensions is composed of a number of subdimensions. Production consists of representation, analysis, and transformation. Coordination consists of control and cooperative subdimensions. Organizational dimension consists of support and infrastructure. Henderson and Cooprider (1990) used a Qsort technique to map specific functionalities, 98 in all which were generated by interviewing leading designers, onto these components and their related dimensions.

Our study empirically tests the above model by using a randomized mail survey in the context of CASE technology.

This Study

CASE is defined as tools and methods to support an engineering approach to software development (Forte and Norman, 1992). We have taken 22 functionalities of CASE tools (see Tables 1, 2, and 3 below) corresponding to all dimensions and subdimensions in the Henderson and Cooprider (1990) model. The limited length of the questionnaire did not permit the inclusion of all the functionalities.

The questionnaire was validated in two stages. In the first stage, three IS doctoral students reviewed the survey instrument for the content coverage of the domain of different constructs. One POM doctoral student reviewed the instrument for the clarity of the questions asked in the questionnaire. After incorporating their feedback, in the second stage, the instrument was reviewed by four faculty members of a large Midwestern university and four IS executives from four different organizations for both content coverage and clarity of the questions asked. Their reviews suggested minor changes, primarily in the wording of questions. These changes were incorporated in the survey instrument. To reduce method bias due to proximity of items measuring the same construct, the questions associated with the operationalization of various constructs were randomly scattered in the questionnaire.

The sample for the study was selected from a database called "Directory of Top Computer Executives" which is maintained by Applied Computer Research Inc., Phoenix, Arizona. The survey was sent out to 1582 top IS executives using first class mail in the second week of August, 1995. A follow-up mailing was done four weeks later. A total of 350 usable questionnaires was returned, representing a response rate of 23.08%. Out of these 350 usable responses, 245 never used CASE, 59 had considered using it at one point in time, but did not use it, while 46 were using at the time survey was administered.

Data Analysis and Results

The purchased sample data came with its own industry classification that included commercial banking, diversified finance, federal government, health service, insurance, local government, manufacturing, retail, state government, transportation, utilities, and others. These thirteen industry categories were consolidated in three: manufacturing, government, and service for ease in data analysis. Many past studies have used these industry categories. A chi-square test revealed that the composition of the sample and responses were not significantly different ($p > 0.05$), indicating that there was no systematic bias among the respondents of various industries.

A principal component factor analysis with varimax rotation was used to find out underlying subdimensions of production, coordination, and organization dimensions. (Limited size of the sample did not permit factor analysis of the 22 functionalities from all the dimensions together). A factor analysis of 9 production functionalities resulted in three factors, explaining 75.1 percent of the total variance (see Table 1). Functionality 9 had loadings in excess of 0.35 on two factors, and hence was dropped. A factor analysis of the remaining functionalities resulted in three simple and interpretable factors. Factor 1 represents analysis and representation subdimensions of CASE. Users do not seem to make a distinction between analysis and representation subdimensions of CASE. However, this result should be interpreted with caution as only one functionality from each of representation and analysis subdimensions was included in the questionnaire. The remaining functionalities (3-8) loaded on two distinct factors. One (factor 2) represents front-end transformation activities (automation of planning or design tasks, data base code/schema generation, procedural code generation, etc), while the other (factor 3) represents back-end transformation activities (automatic restructuring of program code, analysis of program structure, test data generation, etc). Thus, users seem to make a distinction between the front-end and back-end transformation activities.

Coordination dimension does not seem to have two subdimensions as suggested by Henderson and Cooprider (1990). All the functionalities of this dimensions loaded on one factor, explaining 58.9 percent of the variance (Table 2). Users do not seem to make a distinction between control and cooperation subdimensions as they are very closely related. Both seem to require adherence to rules set out by software development team and use of communication among members of the team.

However, organizational dimension seems to have two subdimensions as suggested by Henderson and Cooprider (1990) support and infrastructure. A factor analysis of the functionalities of this subdimension resulted in two distinct factors, explaining 71.8 percent of the total variance. Functionalities (1-4) of support subdimension help users understand and use CASE tools effectively, while functionalities (5-7) of infrastructure subdimension represent standards that enable potability of skills, knowledge, procedures, and methods across planning or design processes.

Our study did not evaluate the capabilities of different

CASE tools used by the respondents' organizations. However, we did find out about the usage behavior of the respondents. 34.8% of respondents used frontend CASE tools, 6.5% used backend CASE tools, and 39.1% used full lifecycle CASE tools. 10.9% used a combination of frontend, backend, and full lifecycle CASE tools, and 6.5% used a combination of frontend and backend CASE tools. The remaining 2.2% used a full lifecycle CASE tool in conjunction with some other CASE tools which they did not identify. The users who used only fulllife cycle CASE tools had the highest usage of production functionalities. However, users who used a combination of frontend, backend, and fulllife cycle CASE tools had the highest usage of coordination and organizational functionalities. Users using only frontend CASE tools had the lowest usage of both coordination and organizational functionalities. The production dimension was least used by users who used a full lifecycle CASE tools in conjunction with other tools. These results indicate that a combination of CASE tools may have more coordination and organizational functionalities compared to standalone frontend, backend, or full lifecycle CASE tools, and usage of different functionalities may be dependent on the availability of those functionalities in the CASE toolset. However, it is also quite likely that users may not exploit all the functionalities of CASE toolset in spite of availability of those functionalities.

In conclusion, our study provides some confirmation of Henderson and Cooprider (1990) model in the context of CASE technology. Future studies should try to empirically test the model using all the functionalities. Future studies may also closely look at the relation between availability of CASE tool functionalities and their use by users in terms of the three dimensions.

References

Available upon request.

Table 1: Production Functionalities of CASE

CASE Tool Functionalities	Factor 1	Factor 2	Factor 3
1. Representation of objects, relationships, or processes	0.8547	.	.
2. Analysis of objects, relationships, or processes	0.9173	.	.
3. Automation of planning or design tasks	.	0.7374	.
4. Data base code/schema (e.g. IDMS) generation	.	0.7735	.
5. Procedural (e.g. COBOL) code generation	.	0.8129	.
6. Test data generation	.	.	0.7765
7. Analysis of program structure	.	.	0.7951
8. Automatic restructuring of program code	.	.	0.7958
9. Analysis of data base structure (dropped)	0.6661	.	0.4201
Eigenvalue	4.0686	1.6089	1.0851
Variance explained	45.2%	17.9%	12.1%

Table 2: Coordination Functionalities of CASE

CASE Tool Functionalities	Factor 1
1. Enforcement of rules, policies, or priorities governing activities of the systems development process	0.7465

2. Resource management: budgeting, scheduling, and tracking	0.7065
3. Access control: auditing, configuration control, and authorization management	0.8654
4. Messaging and electronic communication	0.8668
5. Attaching notes electronically to objects	0.6881
6. Group interaction support (brainstorming, nominal group techniques, etc)	0.7095
Eigenvalue	3.5335
Variance explained	58.9%

Table 3: Organization Functionalities of CASE

CASE Tool Functionalities	Factor 1	Factor 2
1. On-line help for specified commands/features	0.8197	.
2. Templates for tutorials/demos	0.8729	.
3. Explanation facility for recommended actions	0.6896	.
4. Use of domain knowledge to diagnose user problems and recommend appropriate action	0.8209	.
5. Standardized structures to represent designs	.	0.7832
6. Consistency of data definition storage structures	.	0.8961
7. Project repository	.	0.8782
Eigenvalue	3.4650	1.5593
Variance explained	49.5%	22.3%