Abstract

Computer-aided software engineering (CASE) tools have generated much interest as potential means for easing the software development and maintenance bottleneck. To date, the picture regarding their contribution is incomplete and confused, particularly concerning the realization of productivity and quality gains. An in-depth study of one company's experiences with the introduction of CASE is described. Quantitative data is available to allow objective comparison of changes in productivity and IS quality consequent upon the CASE introduction. Questionnaires were used to determine the perceptions of both developers and their customers to the new methodology and tools. The importance to the successful introduction of CASE of the human resource, technical, and managerial infrastructural factors are also investigated.

Introduction

Over the last five years a number of firms have turned to Computer-Aided Software Engineering (CASE) methodologies as a means to enhance the effectiveness and efficiency of information systems (IS) development (for example, Banker and Kauffman, 1991). CASE tools have generated much interest among researchers and practitioners as potential means for easing the software development and maintenance bottleneck (Orlikowski, 1993). To date, the picture regarding the contribution of CASE is incomplete and confused, particularly concerning the realization of productivity gains in systems development.¹

This paper seeks a greater understanding of this contribution. It describes the results of an in-depth study of the introduction of CASE into one large British manufacturing company. It records the productivity and quality gains achieved in the company's IS development and the perceptions of participants to the initiative. Additionally, the organizational factors that impact on success are identified.

The paper is structured as follows. First the company in which the empirical study took place is described. Then the aims of the research are stated, previous literature is reviewed, and the research questions are posed. Next, the data sources including the characteristics of the questionnaire respondents are described. Then the detailed findings including tangible systems outcomes, the perceptions of these tangible outcomes, and perceptions of the associated, less tangible outcomes, are offered, and the "infrastructure" within which the systems development process takes place is discussed. Finally, the additional findings that emerged from the study and the summary and conclusions are presented.
Company Background

The British firm studied for this project is engaged in advanced material, engineering, and manufacturing technologies, and its operations, customers, and suppliers extend to more than 50 countries. The supporting processes are complex and highly integrated, and thus, place challenging demands on the supporting IS that had been developed using several modern systems development methodologies and programming languages. The most prevalent methodology was the Structured Systems and Design Methodology (see, for example, Ashworth and Goodland, 1990), with PL/1 the major language used.

In 1987 the company had embarked on a major review of its existing IS portfolio, focusing on major strategic areas such as computer-aided engineering and manufacturing, product logistics, and financial control. The review showed that deficiencies in the traditional IS development approaches used in the company had resulted in low productivity from the IS function, poor systems quality, long delivery times, and projects that extended their time and money budgets. Additionally, the IS using these methodologies tended to be inflexible and was supported by manual documentation that was difficult to keep up-to-date.

These deficiencies and the new and increasing demands on the IS function stimulated, in 1988, the search for a new and comprehensive IS development methodology. A suitable methodology had to be based on a structured approach, cover all phases of systems development, and be supported by an integrated toolset. The James Martin Information Engineering (IE) Methodology\(^2\) (Martin 1990a-c) was chosen. It was the only integrated CASE approach available in the UK at the time. Information Engineering Workbench from KnowledgeWare and the Information Engineering Facility (IEF) from Texas Instruments were the two complementary toolsets considered. IEF was adopted.

The total cost associated with the introduction of IE was in the region of $9 million. The company made investments in the toolset itself, in personal computers for each developer, enhanced mainframe capacity, and in training. Significant efforts were made to "win the hearts and minds" of both customers and developers. Through 1989 and 1990 all non-managerial customers and developers who might be involved in IE attended a one-day overview course; the customers and developers who would be working together attended on the same day. During this course the participants were informed of what the company was trying to achieve through the introduction of IE and how these achievements would be obtained. The cultural changes that each participant should expect were strongly stressed, particularly the way in which the customer-developer interaction would differ from past practice—especially the need for greater involvement and understanding by customers in the systems development process and developers in the business. Training was given to developers on an as-needed basis after the introductory course, linked to the phase of the IE project on which they were to be engaged.

Customer directors, and later those customer managers directly interfacing with the developers, also attended a one-day course. This course differed from the one given to non-managers—that the links to process engineering were emphasized rather than those to information engineering. The changed form of the customer-developer relationship was stressed and all personnel were made aware of the expected improvements in productivity, quality, and delivery times expected to result from the use of the IE methodology and tools: case studies from other organizations were cited in evidence.

The IE initiative was driven by two objectives. The first was to increase corporate effectiveness by ensuring that resources were deployed to achieve information strategies that supported business goals, particularly by linking business strategy planning with information strategy planning. The second objective was to make available an integrated IS platform that maintains a single consistent picture of the data. By so doing, the sequencing of projects would be improved, interfaces between systems would be better defined, and bridging effort would be reduced. A (unquantified) reduction in maintenance costs was also expected. Although maintenance costs have been estimated to comprise 50-80 percent of IS departmental costs (Banker, et al., 1991), the firm's maintenance costs were never more than 30 percent of the overall budget. At the time of this analysis, the IS function employed some
Benefits of CASE

1,000 staff and had an annual budget of approximately $130 million.

Literature and Research Questions

Aim of the research study

The introduction of IE affected the systems development processes and outcomes, as shown in Figure 1.

Three overarching sets of factors were identified that were thought to have a strong influence on successful systems development: the human resources, technical, and managerial domains. These "infrastructural" factors provided the environment within which developers and customers engage in the systems development process that produces both the tangible systems outcomes and the intangible process outcomes (such as the developers enhancing their knowledge of the business). The incorporation of these two sets of outcomes is based on the model for group decision support systems research proposed by Pinsonneault and Kraemer (1990). Learning theory informs us that behavior is repeated and constant improvements are sought because of positive reinforcement (Luthans and Kreitner, 1985). This motivated the view that perceptions of the systems development outcomes would provide feedback into the systems development process. The developers' personal perceptions of the systems development process provide additional feedback.

The study had five aims strongly motivated by the elements in Figure 1: (1) to determine the tangible systems outcomes associated with the introduction of IE; (2) to gauge the perceptions of the tangible systems outcomes of both customers and developers; (3) to determine their perceptions of the intangible process outcomes; (4) to investigate the personal perceptions of those most affected by the change to IE—the individual developer; and (5) to identify the importance to success of the "infrastructural" factors. The sets of research questions associated with the first four aims are detailed next. The broad findings on the infrastructural factors are discussed later.

This paper compares the situation with and without the application of IE. Outcomes achieved without the use of IE are those associated with IS developed using the types of systems development methodologies and languages indicated above.

Tangible systems outcomes

Three quantitative goals were established for IE by the company. These were to increase the productivity of the systems development process by 300 percent,1 to reduce systems delivery times by 50 percent, and to improve IS quality in order to eliminate category 1 defects4 in delivered systems and to ensure that delivered systems met the customers' needs.

Productivity improvements were expected through the automation of some systems development tasks and the elimination of others. They were not sought from reuse (as in Banker and Kauffman, 1991) because the company had no form of object management, and the version of the IEF toolset it was using did not support an object-oriented approach. It was expected that productivity improvements would be reflected in improved delivery times. This led us to ask questions 1a and 1b.

1a: Has the introduction of IE resulted in productivity improvements in systems development?

1b: Has the introduction of IE resulted in a reduction in systems delivery times (and increased delivery rates)?

Improvements in systems quality were expected to come from customer involvement in all stages of the development process, helping to ensure the elimination of many potential defects. Enhanced involvement was also expected to reduce the number of post-implementation enhancements: it has been reported that the average growth in functionality for CASE-assisted projects was around two percent as opposed to 12 percent for non-CASE assisted projects (Freeman, 1992). We then asked question 1c.

1c: Has the introduction of IE resulted in improvements in systems quality?

It was expected that in the initial period there would be a significant learning curve, which would mean productivity losses until the required skills had been achieved. The association of a
learning curve with IS developments was noted by Banker and Kauffman (1991) and described by Kemerer (1992). This led us to ask question 1d.

Perceptions of the tangible systems outcomes

Nelson (1991) argues that organizations should not devote all of their resources to technological research and development but should focus instead on finding ways in which to make
employees more productive with IT. Much of Figure 1 was motivated by the importance of feedback in providing the incentive for further improvements; specifically that improvements are sought by both customers and developers because of positive reinforcement (Luthans and Kreitner, 1985). Nelson (1991) quotes Lawler (1966) that at high levels of ability (as was the case with both developers and customers in this study) motivation can cause even higher performance levels to be obtained. To heighten motivation it is therefore important that realized gains are reported to and recognized by those who were responsible for them. This led us to ask question 2a.

2a: Do the perceived improvements in systems development outcomes that IE has delivered accord with objective assessments?

The experience of the questionnaire designers was that actual improvements would be better known to development staff than to customers. It was suspected that improvements were not being effectively communicated by customer management to their staff. As discussed later, the reasons include the productivity measure used and the commitment of customer management to IE. We thus asked question 2b.

2b: Do developers and customers differ in their perceptions of the improvements that IE has delivered?

Perceptions of the intangible process outcomes

A key process assumption underpinning the introduction of IE was that it would lead to more and "better" involvement of customers and developers. Involvement was understood in the company to mean that a customer was participating to the appropriate level in systems development or that a developer was participating to the appropriate level in the business: too much participation would be considered just as inappropriate as would too little.

The rationale for this assumption was that IE placed emphasis on business processes rather than simply on the computerization of current systems (which tended to be the case with more traditional systems analysis) and on greater team-work and customer-developer interaction than was traditional.

The empirical evidence to support this view of the value of user involvement is somewhat weak. A review of the evidence in 1984 concluded that the methodological weaknesses of the studies into the link between systems success and user participation in systems development prevented definitive conclusions being drawn (Ives and Olson, 1984). More recent and methodologically sound studies indicate only a weak correlation between user involvement and perceived systems usefulness (Baroudi, et al., 1986 and Franz and Robey, 1986). These findings have been explained by linking user involvement to whether the user considers the system to be important and personally relevant rather than to a set of activities performed by the user (Barki and Hartwick, 1989).

Recent research indicates a high correlation between the perceived user representativeness and user satisfaction for user-led IS developments (Lawrence and Low, 1993). User-led development occurs when user representatives have clear control over a project: including such matters as defining system requirements, overseeing systems testing, and managing the overall project. The company was practicing this form of user involvement.

No study has considered involvement and consequent systems usefulness where CASE is involved. This led us to ask questions 3a and 3b.

3a: Has the introduction of IE led to a perception that developers have increased their involvement in the business?

3b: Has the introduction of IE led to a perception that customers have increased their involvement in the systems development process?

A second key process assumption was that the introduction of IE would lead to customers and developers having a greater understanding of each others' activities. Understanding was viewed in the company to mean that a customer was knowledgeable about the methods used by the developer and of his/her own place in the systems development process. For the developer, understanding was viewed to mean that he/she took a company-wide perspective on the
Benefits of CASE

business processes he/she was analyzing. The terms "organizational savvy" and "business literate" have been used to describe this (Nelson, 1991). This led us to ask questions 3c and 3d.

3c: Has the introduction of IE led to a perception that developers have increased their understanding of the business?

3d: Has the introduction of IE led to a perception that customers have increased their understanding of the systems development process?

Personal perceptions of systems development staff

Conflicting reactions to the introduction of CASE tools were reported by Orlikowski (1993). In one organization the developers reacted positively to the CASE tools, but in a second organization those developers who saw their careers as IS specialists viewed the move to CASE as a threat to their hard-earned skills and experience. All the developers in our study saw their careers as IS specialists.

Because of its structured approach and the emphasis it places on design, management expected that the introduction of IE would result in the developers feeling they needed more programming and design skills. This led us to ask question 4a.

4a: Has the introduction of IE altered the developer's perception of his/her skill requirements?

Management also thought that the power of the IE toolset would give the developers the feeling that they had become more effective and that their job was more interesting because of the enlarged scope of their work and the closer business links. This led us to ask questions 4b and 4c.

4b: Has the introduction of IE made the individual developer feel that he/she is more effective?

4c: Has the introduction of IE made the individual developer feel that his/her job is more interesting?

Data Sources

The on-going systems controls associated with the IE toolset together with established quality assurance procedures allowed quantitative data to be obtained systematically on productivity, delivery rates/times, and quality for both IE and non-IE projects. Additionally, questionnaires were developed to determine the views of developers and customers through interviews with six IS managers, six customer managers, and two employees of the IE vendor. The steps taken to validate the questionnaire are described in the Appendix.

Every customer (32) and developer (56) associated with an IE project was sent a questionnaire. Twenty-six completed questionnaires were returned by customers; 52 by developers. Of the 10 questionnaires not returned, only four came from the same area of the company, so coverage was good. Ten follow-up interviews were carried out with respondents who indicated they would like to discuss the research further. These interviews provided a richer picture than that determined by the questionnaire responses alone.

In any comparative experiment it is important that potentially significant confounding be eliminated. A detailed examination of the IE projects did not indicate that they were significantly different from the comparative non-IE projects. The projects involved a similarly wide range of different business issues. The development teams contained the same skills mix except that personnel with high technical ability were picked for the very first IE project.

The project life cycle used by the company had been adapted from one originally developed by Texas Instruments (Texas Instruments 1990; Martin, 1990a). The stages of this life cycle are business strategy planning, information strategy planning, business area analysis, business systems design, technical design, construction and verification, and post-implementation. The company has modified this life cycle in light of its experience. The two major changes were that the business area analysis stage was split into two and the business systems design, technical design, and construction stages were rolled into a single stage called development.
The profile of the developers in terms of job title, role, and experience in IS development are shown in Table 1.5

The low number of responses from staff having experience with information systems planning is explained by a reliance on a specialist staff for this activity. The low number of respondents who had taken part in post-implementations reflects the smaller size of such teams and the current situation with few implemented systems.6 The customer respondents consisted of 21 customer representatives, three customer managers, and two sponsors.

Table 1. Profile of the Developers in the Survey

<table>
<thead>
<tr>
<th>Job Titles</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Managers</td>
<td>14 Analysts</td>
</tr>
<tr>
<td>10 Programmers</td>
<td>18 Analyst/programmers</td>
</tr>
<tr>
<td>4 “Others”</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Roles*</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Managers</td>
<td>17 Project leaders</td>
</tr>
<tr>
<td>7 Lead analysts</td>
<td>6 Lead programmers</td>
</tr>
<tr>
<td>14 Analysts</td>
<td>10 Programmers</td>
</tr>
<tr>
<td>21 Team members</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experience in Information Systems Development</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3 had less than 1 years experience</td>
<td></td>
</tr>
<tr>
<td>19 had between 1-3 years experience</td>
<td></td>
</tr>
<tr>
<td>14 had between 3-5 years experience</td>
<td></td>
</tr>
<tr>
<td>7 had between 5-10 years experience</td>
<td></td>
</tr>
<tr>
<td>9 had over 10 years experience</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experience of the Stages of IE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>15 in Information Strategy Planning</td>
<td></td>
</tr>
<tr>
<td>26 in Business Area Analysis stage 1</td>
<td></td>
</tr>
<tr>
<td>16 in Business Area Analysis stage 2</td>
<td></td>
</tr>
<tr>
<td>30 in Development</td>
<td></td>
</tr>
<tr>
<td>10 in Post-implementation</td>
<td></td>
</tr>
</tbody>
</table>

*The numbers associated with roles sums to more than the 52 respondents because past as well as current roles are included.

Detailed Findings

Tangible outcomes

One conventional way to measure productivity in IS development is through the use of function points (for example, Albrecht and Gaffney, 1983; Banker and Kauffman, 1991; Low and Jeffery, 1990; Sprouls, 1990). The function point value of a system is independent of the way the system has been designed and implemented and thus provides a common means of measurement.

The function point counting methodology described by the International Function Point Users Group (1992) was employed in this study. IFPUG claims the method to be reliable within 5-10 percent. The count was taken during the post-implementation stage by either the lead analyst or the lead programmer following a rigorously laid-down procedure. For all projects a 1-in-10 sample of the count was taken, and in the one case where discrepancies were found a recount was made. The agreed counts were then fed into a database where they were matched with the corresponding times spent by developers (although not customers) to determine development productivity.

Table 2 shows the development productivity for all the systems implemented since the first IE project was completed early in 1990; they include 11 systems developed using IE and 43 developed using non-IE methodologies. The small number of IE projects means that care must be taken in drawing conclusions from the data.

The mean values across the total period are 20 function points per person-accounting period for IE projects and 11 for non-IE projects. This is a statistically significant increase in productivity of around 85 percent. The introduction of IE has apparently increased productivity (question la) for this organization.

On average, 78 function points per elapsed accounting period10 were delivered for IE projects and 26 for non-IE projects. This represents a decrease in delivery times of around 70 percent. Thus, there is statistically significant evidence that IE has improved system delivery (question 1b).

Two measures of systems quality were available for comparison purposes: the number of system failures11 and the number of requests for
Benefits of CASE

Table 2. Results of the Introduction of IE

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Productivity</strong>&lt;br&gt;(Function Points per person = accounting period)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IE</td>
<td>11</td>
<td>*</td>
<td>22</td>
<td>21</td>
<td>23</td>
<td>22</td>
<td>20</td>
</tr>
<tr>
<td>Non-IE</td>
<td>6</td>
<td>11</td>
<td>7</td>
<td>9</td>
<td>14</td>
<td>17</td>
<td>11</td>
</tr>
<tr>
<td><strong>Delivery Rates</strong>&lt;br&gt;(Function points delivered per elapsed accounting period)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IE</td>
<td>113</td>
<td>*</td>
<td>56</td>
<td>95</td>
<td>60</td>
<td>65</td>
<td>78</td>
</tr>
<tr>
<td>Non-IE</td>
<td>20</td>
<td>46</td>
<td>29</td>
<td>12</td>
<td>20</td>
<td>49</td>
<td>26</td>
</tr>
<tr>
<td><strong>System Quality</strong>&lt;br&gt;(Per project)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Failures</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IE</td>
<td>13</td>
<td>16</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>Non-IE</td>
<td>17</td>
<td>22</td>
<td>18</td>
<td>23</td>
<td>21</td>
<td>23</td>
<td>21</td>
</tr>
<tr>
<td>Requests for Change</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IE</td>
<td>5.4</td>
<td>4.5</td>
<td>3.8</td>
<td>3.4</td>
<td>3.5</td>
<td>4.3</td>
<td>4.1</td>
</tr>
<tr>
<td>Non-IE</td>
<td>5.1</td>
<td>5.2</td>
<td>3.3</td>
<td>3.9</td>
<td>4.2</td>
<td>5.1</td>
<td>4.5</td>
</tr>
</tbody>
</table>

* = no IE projects were started in this period.

changes. Both are measured within the six-month warranty period that the IS department gives to its users. (In practice very few failures or requests for changes arise after this period.) Table 2 shows that there are twice the failures associated with non-IE systems as with IE systems. This difference is statistically significant.

Requests for changes result either from a system not doing all the customer envisaged it would do or simply because the customer asks for additional functionality. There is an insignificant difference between the requests for changes called for in IE projects and those in non-IE projects.

We can conclude, therefore, that an improvement in quality has been attained through the use of IE (question 1c) as measured by system failure.

During interviews, several systems managers stated they believed that there was an underlying trend of continuous improvement in tangible systems outputs. Leaving aside the atypical first period associated with the very first IE projects, the figures in Table 2 do not support this contention (question 1d).

**Perceptions of the tangible systems outcomes**

Table 3 shows the scores for the three questions in the questionnaire covering the perceived improvements in productivity, delivery, and quality. A score of zero indicated that the respondent thought no improvement had been achieved; a score of 10 meant that the respondent thought the target had been fully achieved.

The productivity improvement associated with the introduction of IE is 85 percent in this firm, considerably less (around one-third) than the target of 300 percent. The mean scores of between 2.8 and 3.7 suggest that both developers and customers had a reasonable view concerning the improvement in productivity that had been achieved. The improvement in delivery times is
Table 3. Perception of Delivered Improvements

<table>
<thead>
<tr>
<th></th>
<th>Mean Scores</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Customers</td>
<td>Developers</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Range</td>
</tr>
<tr>
<td>To what extent has a 300% improvement in productivity been achieved? 3.1; 5.1</td>
<td>2.8</td>
<td>2-5</td>
</tr>
<tr>
<td>To what extent has a 50% improvement in delivery times been achieved? 3.2; 5.2</td>
<td>2.8</td>
<td>0-5</td>
</tr>
<tr>
<td>To what extent have improvements in quality been achieved? 3.3; 5.3</td>
<td>3.5</td>
<td>2-7</td>
</tr>
</tbody>
</table>

Notes: There were 26 responses from customers, 52 responses from developers. The small numbers after the question refer to the question numbers in the customer and developer questionnaires respectively.

70 percent, but neither developers nor customers believe strongly that the target of 50 percent had been achieved. The customers' perception of systems quality was that some improvement had occurred, while developers considered that achievement was half of the target. Overall, one can say that both customers and developers had a reasonable view of productivity and quality improvements but that improvements in delivery times were not well recognized.

In all cases, customers had a lower awareness of the achieved benefits than did the developers (question 2b). The reason for this was not hard to find. Productivity information in terms of function points is put forward at "process improvement" meetings to all development managers and is subsequently communicated to their staff. Customer management is similarly informed, but the function point measure does not mean much to them: they are concerned with achievement against contract and with adverse budget variances. Since projected productivity gains are included in these budgets, the realization of these gains is not noteworthy. These two factors militated against effective communication of improvements to customers. The lack of customer management commitment appeared to be another reason, as is discussed below.

Perceptions of the intangible process outcomes

Four questions addressed the achievement of developer and customer involvement and understanding. A score of 10 would indicate fully appropriate involvement, or total understanding; a score of 0 would indicate a totally inappropriate level of involvement, or zero understanding. Before the results were obtained it had been agreed between researchers and management that a score of 5 or more would be deemed acceptable.

The first two columns of data in Table 4 indicate that both customers and developers thought that understanding and involvement had been achieved to an acceptable level. These views were consistent both through intra-group ratings (for example, by developers of developers) and by inter-group ratings. Thus, one of the main motivators for the introduction of IE had been reasonably well accomplished (questions 3a-d).

The developers perceived a higher achievement of involvement and understanding than did the customers. There was also a difference between how the achievement was seen by experienced developers (those with more than 3 years of systems development experience) and those with less experience. These perceptions are shown in the final two columns of Table 4. The more experienced developers associated IE with lesser increases in involvement and understanding. In subsequent discussions, developer management explained this as occurring because all developers were now interacting with customers; the more experienced staff were seeing this as a reduction in their power and influence while the less experienced (and normally younger) were seeing it as a good opportunity to increase their
### Table 4. Achievement of Understanding and Involvement

<table>
<thead>
<tr>
<th></th>
<th>Customers</th>
<th></th>
<th>Developers</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Range</td>
<td>Mean</td>
<td>Range</td>
<td>Less</td>
<td>More</td>
</tr>
<tr>
<td>Increased developer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>involvement with the business</td>
<td>5.3</td>
<td>3-8</td>
<td>6.2</td>
<td>4-8</td>
<td>6.8</td>
<td>5.8</td>
</tr>
<tr>
<td>Increased developer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>understanding of the business</td>
<td>5.2</td>
<td>4-8</td>
<td>6.3</td>
<td>4-8</td>
<td>6.9</td>
<td>5.8</td>
</tr>
<tr>
<td>Increase in customer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>involvement in projects</td>
<td>5.5</td>
<td>3-7</td>
<td>5.9</td>
<td>2-9</td>
<td>6.9</td>
<td>5.4</td>
</tr>
<tr>
<td>Increase in customer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>understanding about projects</td>
<td>5.2</td>
<td>3-7</td>
<td>5.5</td>
<td>2-9</td>
<td>6.1</td>
<td>5.0</td>
</tr>
</tbody>
</table>

**Notes:** High mean scores indicate a perception of high achievement. There were 26 responses from customers; 52 responses from developers. The small numbers refer to the question numbers in the customer and developer questionnaires respectively.

Business awareness. Additionally, the more experienced developers now had to revert to programming—something they thought they had left behind.

**Personal perceptions of systems development staff**

Out of 52 replies, 40 developers believed that the introduction of IE had made some change to the skills they required to do their jobs (question 4a). Twenty-one developers thought they needed more business analysis skills, 20 thought they needed more systems design skills, and 21 thought they needed more programming skills.

There was a relationship between the stages of a project that developers had worked on and the additional skills they thought they required. Eighty percent who had worked on the information strategy planning stage and who believed that the introduction of IE had changed the skills they required believed they needed more business analysis skills. This compares with the development stage where approximately 95 percent who believed IE had changed their skills requirement believed they needed more programming and/or design skills; only five percent believed they needed more analysis skills. All respondents who had participated in the post-implementation stage believed IE had changed their skills requirement, split equally among analysis, programming, and design.

Twenty-five respondents thought IE had made them more effective, 26 thought it had made no difference, and one person thought it had made him/her less effective. Overall, IE is seen as having had a positive impact on developer effectiveness, but over 50 percent remained unconvinced (question 4b). There was a difference between the experienced and less experienced developers. For every experienced developer who thought that IE had made them more effective, there were two who felt it had made no difference. The reverse was the case with the less experienced developers: 14 replied that the use of IE had made them more effective, while only seven said it had made no difference.

Sixty-five percent of the respondents believed IE had made their job more interesting, and 35 percent believed it had made no difference. In general, therefore, IE had increased the developers’ job interest (question 4c). This does
not confirm previous findings that technically oriented systems developers are threatened by the introduction of IE (Orlikowski, 1993); indeed, the situation was quite the reverse—developers who were initially not involved with IE believed they were being left behind, with less marketable skills than those who were involved. Again, there was a difference between the experienced and inexperienced developers. Roughly 80 percent of the inexperienced developers said that the use of IE had made their jobs more interesting, while only 50 percent of the experienced developers thought this.

**"Infrastructural" Aspects**

Table 5 summarizes the responses covering the importance of the "infrastructural" issues. In each subsection of the questionnaire the respondents were asked to rank the items in

<table>
<thead>
<tr>
<th>Table 5. Importance of &quot;Infrastructural&quot; Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scores</strong></td>
</tr>
<tr>
<td><strong>Customer</strong></td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>Human Resource Issues</td>
</tr>
<tr>
<td>Education of developers in cultural change needed 2A.1; 4A.1</td>
</tr>
<tr>
<td>Education of customers in cultural change expected 2A.5; 4A.7</td>
</tr>
<tr>
<td>Appropriate Analysis skills of developers 4A.3</td>
</tr>
<tr>
<td>Appropriate programming skills of developers 4A.4</td>
</tr>
<tr>
<td>Training of customers in the technical aspects of IE 2A.6; 4A.8</td>
</tr>
<tr>
<td>Appropriate people skills of developers 2A.2</td>
</tr>
<tr>
<td>Appropriate business skills of developers 2A.3</td>
</tr>
<tr>
<td>Appropriate technical skills of developers 2A.4</td>
</tr>
<tr>
<td>Technical Issues (asked only of the developers)</td>
</tr>
<tr>
<td>Internal technical support available 4B.1</td>
</tr>
<tr>
<td>External consultancy available 4B.2</td>
</tr>
<tr>
<td>Stability of the IE toolset 4B.6</td>
</tr>
<tr>
<td>Mainframe capacity 4B.8</td>
</tr>
<tr>
<td>Logical model management 4B.3</td>
</tr>
<tr>
<td>Physical model management 4B.4</td>
</tr>
<tr>
<td>Development of standards and procedures 4B.5</td>
</tr>
<tr>
<td>Management Issues</td>
</tr>
<tr>
<td>Commitment of developer management to IE 2C.2; 4C.2</td>
</tr>
<tr>
<td>Commitment of senior customer management to IE 2C.3; 4C.3</td>
</tr>
<tr>
<td>Quality controls 2C.6; 4C.6</td>
</tr>
<tr>
<td>Ability to plan and control projects 2C.7</td>
</tr>
</tbody>
</table>

*Questions not asked of this group.

**Notes:** Low mean scores indicate high importance; high scores indicate low importance. There were 26 responses from customers, 52 responses from developers. The small numbers refer to the question numbers in the customer and developer questionnaires respectively.
Benefits of CASE

order of importance: 1 signifying the most important, 2 the second most important, and so on.

Human resource issues

There is a distinction between learning how to use the toolset and learning the underlying methodology (Kemerer, 1992). When introduced to the firm, IE was a different methodology than those currently in use; thus, it was to be expected that the developers would be deficient in some of the attributes needed to best utilize IE: in particular the use of IE demanded a cultural change to systems development. The cultural change expected was that both customers and developers would work as a more integrated team, that systems development would now be process-oriented, and that there should be no defending organizational boundaries. At the operational level the IE approach called for enhanced people skills including listening, adopting a suitable questioning approach, and understanding potential resistance to change.

It was also expected that the skills possessed by the developers would not be sufficient. The relationship between learning and job performance is well-recognized, in particular the general rule that, holding motivation constant, performance will improve with increased ability (Nelson, 1991).

Table 5 shows that customers generally ranked the importance of human resource skills higher than did the developers and those pertaining to developers higher than their own. They ranked the developers’ people skills as the most important human resource issue, closely followed by the developers’ technical skills. Business skills ranked third in importance, followed by the issue of the cultural change required of the developers.

Developers considered their own skills as the most important human resource issues. Cultural change was considered to be of the highest importance, perhaps reflecting that developers had been called upon to alter their side of the customer-developer interaction more radically than had the customers. They regarded analysis skills as more important than programming skills.

There was general consensus between developers and customers that it was not important for customers to understand the technical aspects of IE. This view was not shared by their managers.

Technical issues

Questions regarding the technical infrastructural issues were posed only to developers. We expected that the success of IE would depend on the effectiveness of the technical infrastructure and of the systems management practices. As shown in Table 5, internal technical support was considered the most important technical issue impacting the success of IE. Having external consultancy assistance available was rated the least important. However, these simple results do not tell the whole story. In the early days of IE adoption it was very important for support to be available from the IE and IEF vendors. As experienced was developed and stability attained, the use of external consultants diminished as sound internal technical support became available. The questionnaire responses were relevant to an organization with several years experience of using IE.

The stability of the IE toolset was considered the second most important technical issue. In the initial period of the IE introduction, frequent changes to the toolset had had an unsettling effect on the developers. The lack of sufficient mainframe capacity ranked third in importance. Even though upgrading of computer facilities was undertaken, the sheer success of IE caused computational demands to outstrip capability.

Model management techniques and development standards and procedures were not considered of particular importance. However, from an analysis of the data and from subsequent discussions with respondents it would appear that developers on later projects attached more importance to the existence of these techniques and standards and procedures than did those on earlier projects.

Managerial issues

Many studies have shown the vital contribution that management commitment makes to change management in general and to the introduction of IS in particular (see, for example, the seven case studies in which the effectiveness of senior executive support for IS development is described by Jarvenpaa and Ives (1991)). The data in Table 5 show that this view is held by both developers and, albeit a little less strongly, by
customers. The head of the IS function was very committed to IE, insisting that IE be used with all projects unless a very good case could be made for doing otherwise. Most developer managers regard the introduction of IE as a very valuable and progressive step taken by the IS department. However, customer managers were not nearly so committed. They saw the company as an engineering and design company, not as a knowledge company.

The introduction of quality control procedures is recognized as an important contributory factor in successful IS development, and this applies equally to the use of CASE (Gibson, 1989). Customers considered the ability of its management to plan and control projects as the second most important management issue, with the establishment of quality controls the third. Quality control was considered the least important issue by the developers, probably because the use of the toolset was enforcing the necessary quality standards.

### Other Findings

One important lesson drawn from the introduction of IE has been the importance of having multi-skilled individuals on project teams. This was one consequence of using an integrated CASE methodology and an integrated toolset since these covered all stages of systems development. Multi-skilling encouraged continuity of personnel on a project, thus enhancing teamwork. The company strove to train all staff to an adequate level in both analysis and programming, and it emphasized that specialist skills (such as data analysis and data base design) can be developed to a valuable level in non-specialists with very specialized support being provided by the center.

One approach is the use of SWAT teams (Staff With Advanced Tools). In such teams each individual, while expected to do all of the tasks in the project life cycle, would not be expected to be a specialist in all. Instead, SWAT teams of two or three people would have one member who was highly skilled at analysis, one skilled in programming, and perhaps one who was skilled at database design; each would then lead and assist the others in the skill in which they were most proficient. Such teams require training that encourages the development of multiple skills and a recruitment program that actively looks for staff capable of performing at each stage of the project life cycle.

Development managers report that those customer managers who benefited from business reengineering were most satisfied with IE; those that only got an enhanced IS without any reengineering tended to be less satisfied. One senior customer manager commented that one of the by-products of his IE project was that the staff had realized for the first time that they did the same job and that they would benefit if they had a single functional head. Therefore, at the same time as introducing the IS, this particular business area also reengineered its business processes.

### Summary and Conclusions

This paper is concerned with the introduction of IE into a large company. The study is one of the first to demonstrate that a form of CASE and its associated toolset can deliver considerable benefits both to an organization and to the individual developer. In particular, it adds to our knowledge of how practitioners might be affected by the introduction of a CASE methodology and its toolset.

For this firm the introduction of IE has led to statistically significant productivity gains of 85 percent, system delivery rate increases of around 200 percent, and quality improvements. But learning seemed to stall early, with little additional improvement over time. Neither the developers nor the customers were fully aware of the improvements in tangible systems outputs that had been achieved, so some of the hoped-for feedback to build on tangible success did not occur. The less tangible process outcomes of improved involvement and understanding were reasonably well achieved. The developers viewed IE as making their job more interesting and on balance making them more effective, with the less experienced developers thinking this most strongly. The developers also believed that their traditional IS development skills needed augmentation.

Customers considered the enhancement of the developers' skills, particularly their people and
Benefits of CASE

technical skills, as the most important human resource issue. Developers considered their own skills as the most important human resource issue, with cultural change of greatest importance. This perhaps reflected the need they perceived to alter their side of the customer-developer interaction more radically than did the customers. The major technical issues were the need for technical support, a stable toolset, and adequate mainframe capability.

Management commitment was considered the most important managerial issue. Other major management issues of concern were project planning and control and the involvement of customers in the IE process.

We see several implications for systems development practice. Figure 1 is of practical applicability and should encourage both customer and developer management to consider more fully the “softer” issues in IS development, particularly the human activity feedback system. Unless a sustained effort is made, neither customers nor developers will be fully aware of the benefits associated with IE; without such knowledge the reinforcing feedback will be weakened. The changes in working practices should not be underestimated; there should be an appropriate infrastructure required to support the introduction of IE. Senior management support is vital, especially in providing an adequate level of technical support in the form of a stable toolset, adequate computational power, and either internal or external consultancy support. Finally, reengineering is likely to leave the customer better satisfied.

There are three main implications for systems development research. First, the research reported in this paper concerns one organization with its own particular culture and background introducing one form of CASE. The respondents constituted around 90 percent of all the people who had had experience of IE and covered all types of developer and many levels of customer. All respondents were of high intellectual quality. The range of experience and age of the respondents was wide. With the validation methods applied, the results have high internal validity. However, there is a need to extend the generalizability of the results by carrying out similar detailed research in other organizations and where other forms of CASE are used. Second, to enable the benefits to be fed back to customers, there is a need to develop a measurement method that fits with their needs. This can come either from educating customers in the meaning of function point methods or by devising a more appropriate method of measurement. Third, the infrastructural elements need to be examined in more detail: for example, the form that senior management support should take in order to optimize their input.

Acknowledgements

We would like to thank the senior editor of this journal for the very great help he has given us in improving the style and readability of this paper.

Endnotes

1 For example, no gains through the use of CASE were reported by McGaff (1989), 20-30 percent were claimed by Sperling, et al. (1989) and by Freeman (1992), and 60 percent were reported by Gabel (1989), albeit in a pilot project.
2 Now JMA Information Engineering.
3 Unlike the tangible systems outcomes, the intangible process outcomes could not be measured directly. Thus, only the developers’ and customers’ perceptions of the intangible process outcomes are discussed.
4 The organization categorized defects into three categories. Category 1 defects are those where the system fails, stops, or is unusable, or where incorrect data has been accepted, or where incorrect output has been produced.
5 All staff associated with systems development (customers and developers) were graduates—overwhelmingly from British universities—and thus, in broad educational terms, lying in the top 15 percent of the population (HMSO, 1993). For this prestigious company, however, their ability to recruit the best applicants suggests that the staff in this survey were from the top 5 percent.
6 Jones (1991) provides a comprehensive list of the factors in IS experimentation that may need to be considered.
7 These changes were introduced to provide greater project control and have not changed the overall tasks of an IE project from those in the Texas Instruments approach. An executive summary of the IE methodology is given by Richmond (1992).
8 Four respondents had worked in teams of one or two members, 15 had worked in teams of three-five, 27 had worked in teams of six-10, and eight had worked in teams of more than 10.
9 T-tests were performed to test for statistically significant differences between means. Regression analysis was performed to test for statistically significant trends. The level of significance used was 0.05.
10 Productivity per person will generally be related to the delivery rates: the higher the productivity, all things being equal, the higher the delivery rate. However, higher productivity could be used to reduce the head count and keep the
Benefits of CASE

delivery rates the same. Thus, the two measures reflect the trade-off the company has made between delivering IS faster and reducing headcount and the disposition of developers over projects.

A systems failure is defined to occur when the system is unusable or when an online transaction or a batch program stops or is stopped during its execution. This is not the same as a defect: one defect may lead to several systems failures if several users are affected by the same defect. The mean number of IE systems failures (11) would have arisen from 1 or 2 defects.

The whole population of customers and developers who had had experience of using IE was targeted rather than simply a sample of it. Almost 90 percent of the questionnaires were completed. Thus, it was considered inappropriate to carry out statistical testing to infer population statistics from the samples.

Toward the end of the period of the study reported in this paper, the company launched a systems engineering (SE) initiative aimed at far-reaching business reengineering. It has since been realized that IE and SE should form an integrated approach, and this has resulted in an edict that no new IS be developed unless they are supporting the business reengineering arising from the SE process. In this way, the SE is forming the initial stages of the IE methodology.

References


Low, G.C. and Jeffrey, D.R. "Function Points in the Estimation and Evaluation of the Software
Benefits of CASE


About the Authors

Paul N. Finlay is professor of strategic information systems in the Loughborough University Business School, England. Earlier, he spent six years working in strategic operational research with the UK Ministry of Defence and five years engaged in tactical operational research studies and corporate planning within a large multinational organization. His present research interests are the impact of IT on the strategic standing of commercial organizations and the modelling and validation of decision support systems. He has written four books and has published over 50 academic articles.

Andrew C. Mitchell is a project manager with a large multinational organization based in the UK. He has had many years experience with several software development methodologies and was chosen to spearhead his present organization's move into information engineering.

Appendix

Instrument Validation

An initial set of research questions was drafted prior to the first pretest interview, and these questions were refined into the customer and developer questionnaires throughout the questionnaire development process. This development followed fairly closely the one detailed by Straub (1989), but without the technical validation being treated separately.

Pretest and Technical Validation

The six senior managers responsible for formulating the IS department's policies and strategies were interviewed. The interviews were predominantly semi-structured but progressed from an open-ended
discussion phase to a much more thorough examination of questionnaire detail. Concepts introduced by the interviewees were noted and were in their own language. The interview began with the history behind the introduction of IE and a discussion of the changes in methodology that had taken place in the preceding five years. The extent of the investment in IE was considered, including training and consultancy and the inevitable learning curve that comes with such a major change. The motivational forces behind the change were also reviewed.

These interviews also covered the current situation to discover the extent to which the organization had planned for and understood the changes taking place, the benefits that IE was achieving, and the problems being encountered. Finally, the managers' views on the future of IE within the organization were sought.

Structured interviews were also conducted with one senior customer manager from each of the company's six main business groups, and two additional interviews were conducted with the major customer for IE. The interviews followed the same format as with the developer managers.

Semi-structured interviews were also undertaken with two experienced employees of the IE vendor to establish their views on the impact IE had had on the company and how this compared with other organizations.

An instrument valid in content is one that has drawn representative questions from a universal pool (Cronbach, 1971; Kerlinger, 1964). It is considered that the contributions from all 16 interviewees, together with the academic input, ensured a high level of content validity in the questionnaires.

Construct Validity

There are four sets of research questions. Construct validity is less of an issue with some than with others. The research questions of set 1 concern "facts" not elicited from the questionnaires, and thus, the question of the construct validity of the questionnaire does not arise. The research questions of set 2 relate to perceptions of well-defined variables—productivity, delivery times/rates, and quality. Given how well-defined these terms are, the three questions asked provided a very high level of construct validity.

Research question set 3 is concerned with perceptions of understanding and involvement. There are potential problems in using such simple terms. However, the terms were in common use within the company, and the researchers are convinced that there was a common view of the meaning of these two terms. (These meanings are defined in the main text.) Research question set 4 is concerned with the perceptions of the developers to the introduction of IE as it affected themselves. Given that the concepts used were so simple (extra skills required, individual effectiveness, and job interest), it is considered that any lack of validity due to the use of single questions is likely to be minimal.

A problem of construct validity is indeed present with the use of some of the single questions in section 4 of the developers' questionnaire and the corresponding section of the customers' questionnaire. The scope of the study asked for by company management meant that an already large questionnaire would be further inflated by the need to ask multiple questions on each issue. The participants were a fairly homogeneous group of people with a strong, common view on the meaning of the terms, and thus, the questions contained little ambiguity for the participants (as seen during the pilot testing). Multiple questions were not considered of the highest importance, especially when set against the dangers of reducing the number of questionnaire responses and the problems of obtaining management support. Nevertheless, these problems did mean that these questions were less valid than those associated with the four sets of research questions. For reasons of research rigor the findings from this section have been considered less formally than the research questions—in the section "Infrastructural Aspects" with a summary in Table 5.
Benefits of CASE

Pilot Test
In order to ensure that the questionnaires were properly understood, they were piloted with four developers and two customers, each of whom was questioned on his/her understanding of the questions. Minor modifications were then made to produce the main study questionnaires.

Full-Scale Survey
Every question from each completed questionnaire was analyzed to assist in establishing construct validity. Wherever a respondent had answered more than four questions with a response differing from the mean response of all respondents by more than twice the standard deviation, then a follow-up interview was arranged with that person. The result of this was that five people were interviewed. Two of these five had misunderstood the ranking system used and once the true intention had been explained, they provided a corrected questionnaire.

Follow-Up Interviews
The first line on the questionnaire was optional. The respondents could provide their name if they were willing to be contacted individually to discuss their responses. Fifteen follow-up interviews were carried out in total, five as discussed above. The other 10 follow-up interviews provided a fuller view on the research questions under discussion.