

A STUDY ON THE BENEFITS AND PROBLEMS OF THE OBJECT-ORIENTED PARADIGM

Patrik Paetau

Swedish School of Economics and Business Administration, Vaasa, Finland
patrik.paetau@hanken.fi

Abstract. Several researchers are of the opinion that there are many benefits and problems in using the object-oriented paradigm in information systems development. The objective of the following study was to investigate and to gain more understanding of the benefits and problems of the object-oriented paradigm. A review of previous studies was made and several benefits and problems were established. Further a survey was made in order to get some knowledge on what benefits and problems with the object-oriented paradigm Finnish software companies had experienced. 104 companies answered the survey that was sent to all Finnish software companies with 5 or more employees. The major finding was that Finnish software companies were exceptionally positive towards the object-oriented information systems development and had experienced very few of the proposed problems.

1 INTRODUCTION

There are few available studies on the benefits and problems of the object-oriented paradigm (Pomberger & Blaschek, 1996, p. 282), of which the studies by Johnson (2000) and Pickering (1996) are probably the most worthy of note. Furthermore in the book by Cockburn (1998, pp. 23-30) several benefits of the object-oriented paradigm are presented; these benefits are based on comprehensive interviews with project leaders that Cockburn has made, discussions with consultants and experts also made by Cockburn, and on information from project reports that Cockburn has read.

Because knowledge of the benefits and problems of the object-oriented paradigm might be a significant success factor for software companies as well as for other companies, more awareness of this issue is therefore needed. Verity & Schwartz (1991) propose that in an era when hardware is a commodity, software will be the most important competitive factor, and the software companies, the traditional industry and service companies, and the computer manufacturers that exploit object-oriented software development the best are likely to succeed in the computer and software industry itself.

1.1 Purpose and scientific methodology

The purpose of this study is to investigate and gain some understanding of what *benefits and problems* there are with the *object-oriented paradigm* and what *benefits and problems Finnish software companies have experienced*.

In this work underlining is used in order to improve readability.

Literature review

During the literature review several (about 300) scientific articles, books and conference papers were studied. The findings from the review of previous studies could be presented in a chapter of its own but are situated in the appendix and are presented in the Theory - Studies part of the questions in the survey. When this review was made the author of this study searched for benefits and problems with the object-oriented paradigm.

Empirical research and the selection of a research method

After considering different research methods, the following conclusion has been reached: the best solution and the overall empirical research method will be a *a survey*. An important argument for a survey is presented by Verschoor & Low (1994) who argue that “as with any study investigating general ‘state-of-practice’, a survey is a feasible means of providing data with sufficient external validity”. Because the empirical part of this study is concerned with treatment of the ‘state-of-practice’ of experienced benefits and problems with the object-oriented

paradigm among Finnish software companies, a survey seems to be an appropriate research method.

In the survey of this study only Finnish software companies with five or more employees were selected, the subjective reasoning behind this decision is as follows:

- Very small Finnish software companies with 1-4 employees probably have limited experience of object-oriented information systems development.
- They are unlikely to answer the survey.
- They are numerous and would make the selected population too large to manage in an efficient way.

The population of Finnish software companies for the survey in this study became 808 software companies. Note that software companies with an unknown number of employees were excluded from the selected population.

For the survey a questionnaire was developed and based on the research questions in this study. The questionnaire was first developed and written in English and after that was translated into Finnish.

The survey in this study is based on a mail survey.

The survey with the questions, results, conclusions and a discussion are presented in the appendix.

A total of 130 answers were received, of these:

- 104 were valid responses.
- 24 were received as 'return to sender' presumably due to the following reasons; inaccuracies in the mailing list, companies had gone bankrupt or companies had moved, etc. Some information regarding this issue was obtained from the Finnish Post who returned the mail.
- 2 surveys received were incomplete.

Thus, a total of 104 valid answers were obtained. The 104 valid surveys out of 788 surveys sent, reflected a valid response rate of 13,2%. This was considered sufficient for this study because of the large number of answers (104) and because the questionnaire had been sent to all valid companies in the population (total survey).

The general quality of the answers received further increased the satisfactoriness of the responses. This quality was reflected in a review of the job titles of the respondents, which suggested that over 70% of the surveys were completed by managers, consultants and system analysts, etc. (Question IV).

One can argue that the validity of the survey was appropriate because Statistics Finland provided the population and the survey was made for all software companies in the population. A company is defined as a software company by Statistics Finland if the company defines itself as a software company. The questions in the questionnaire were gathered from the review of previous studies and theory.

The reliability of the survey was adequate because the number of answers was sufficient and the quality of the answers was high.

Because the questionnaire was sent to all adequate software companies in the population and the number of answers was as high as 104, one can profess that one can generalise about all (more than 4 employees) Finnish software companies. No systematic drop out of software companies among the responses was found (for example, the questionnaires received after a remind message had been sent, were compared with the questionnaires received earlier), and one can argue that the software companies that participated in the survey most likely reflect a good sample of all (more than 4 employees) software companies in Finland involved in *software development*. However, one has to be aware of the danger that software companies involved in object-oriented information systems development were more willing to answer the survey than companies not involved.

The results cannot be generalized to populations outside Finland. However, the sample represented a wide variety of information systems developers. Respondents varied from those only slightly familiar with the object-oriented paradigm to those who were very experienced with it. There was a broad spectrum of jobs including executive chiefs, managers, analysts, programmers, consultants and even a 'share holder'.

1.2 Boundaries

The focus of this study is on the areas of the object-oriented paradigm specified above. This study is neither concerned with object-oriented information systems development analysis *methods* nor with design *methods*; if the reader is interested in these issues the work by Wieringa (1998) is recommended.

This study does not investigate benefits and problems at the programming level as many programming problems are often tied to a specific programming language (Webster, 1995, p. 191), or database management level.

One also has to consider the possibility that individuals who were more favourably disposed to the object-oriented paradigm were more likely to respond to the survey, thereby biasing the results in favour of the object-oriented paradigm. However, great care was taken in the wording of the cover letter, survey instructions and survey items to avoid any bias for or against the object-oriented paradigm.

1.3 Research problems

According to several researchers there are many benefits in using the object-oriented paradigm in information systems development (Booch, 1994, pp. 3-25; de Champeaux et al., 1993, p. xiv; Henderson-Sellers & Edwards, 1990; Jacobson et al., 1995, pp. 45-48; Smith & McKeen, 1996; Winblad et al., 1990, pp. 43-51, etc.). For example, according to Taylor (1990, pp. 103-107) there are the following potential benefits: faster development, higher quality, easier maintenance, reduced cost, increased scalability, better information structures and increased adaptability.

There are also of course problems with the object-oriented paradigm. However, there is still little knowledge on how companies have experienced the benefits and problems when using the object-oriented paradigm (Miah, 1997). Maring (1996) proposes that companies know little about how to use the object-oriented paradigm with predictable results. However, in a study by Villeneuve & Fedorowicz (1996) with 218 practitioners it was found that perceived benefits of the object-oriented paradigm depend on the size of the software development project and the scope of use of the object-oriented paradigm through the systems development life cycle. In the study by Johnson (2000) the question of benefits and problems with object-oriented systems development was also studied and it was found that the benefits are recognized but the problems are virtually nonexistent.

Nevertheless, there is still a lack of comprehensive studies on how to develop object-oriented information systems by utilising the benefits and avoiding the problems. One can actually argue that there is a need for knowledge on this issue in the information systems development community (McGregor, 1996).

When the review of previous studies was made several benefits and problems with the object-oriented paradigm were found. The research problems have been developed out of these benefits and problems. This reference to literature is recommended by Eisenhardt (1989) and by Yin (1994, p. 9).

The research problems for this study are the following:

RP1: What are the *benefits* experienced with the object-oriented paradigm in information systems development?

Have the information systems development projects, for example, been faster or easier? Has the reuse concept been useful?

RP2: What are the *problems* experienced with the object-oriented paradigm in information systems development?

For example, has the object-oriented paradigm been considered immature? Has the object-oriented paradigm been considered difficult or complex?

2 RESULTS AND DISCUSSION

This chapter first summarises the major findings regarding the benefits and problems with the object-oriented paradigm. Then the major empirical findings are discussed with a focus on the explanation of the results.

Benefits. The empirical study showed that most of the Finnish software companies are very positive towards the object-oriented paradigm. One has, however, to take into consideration the risk that companies that are more positive towards the object-oriented paradigm also might be more willing to answer the survey. The response rate of 13,2% was not very high.

A substantial majority of the Finnish software companies had experienced the following benefits. The percentage figure in parentheses is the “yes” category.

1. The object-oriented paradigm is useful when developing large-scale and complex information systems (94%).
2. Reuse is beneficial (92%).
3. The quality of object-oriented systems is better than the quality of traditional systems (70%).
4. Object-oriented information systems development is more productive than traditional information systems development (68%).
5. Maintenance of object-oriented information systems is easier than maintenance of traditional information systems (64%).

The Finnish software companies had not experienced the following proposed benefits with the object-oriented paradigm:

1. The companies had not experienced a better and more 'natural' communication between information systems developers and end users due to the use of the object-oriented paradigm. Only 22% of the companies had experienced a more natural communication, when as many as 57% of the companies had NOT experienced this.
2. The companies had not experienced that the object-oriented system development process could be seen as a uniform 'one model' from problem domain to code and maintenance. 42% of the companies had experienced a uniform one model whereas the same number (42%) had NOT experienced this.

The two exceptions above reflect unexpected empirical evidence when compared with the findings in the review of previous studies.

Problems. It was also much unexpected that the Finnish software companies had experienced so few of the proposed problems.

For the problems, the following were most commonly agreed:

1. It has been difficult to find experienced object-oriented software developers and systems analysts (**65%**).
2. Companies have experienced computer efficiency problems in object-oriented information systems development projects (**49%**).

When analysing the results regarding experienced problems, there are several problems sighted in the review of previous studies that the Finnish software companies have not experienced to a significant degree. The following are the most important:

1. The object-oriented paradigm is still immature (28% 'Yes', **66%** 'No' and 6% 'Not sure').
2. Testing object-oriented information systems has been difficult (29% 'Yes', **62%** 'No' and 9% 'Not sure').
3. There has been a problem with reuse for the reason that software developers do not want to reuse a component, because they claim that it does not work (28% 'Yes', **61%** 'No' and 11% 'Not sure').
4. The object-oriented paradigm is considered complex (35% 'Yes', **58%** 'No' and 7% 'Not sure').

The exceptions above are interesting. These empirical results differ from the findings in the previous studies. This discrepancy with mainstream studies on the object-oriented paradigm seems, however, to be in agreement with recent findings

by other researchers (Johnson, 2000; Johnson, 2002), and may have to do with the fact that tools and experience have changed in ten years.

As mentioned earlier in this study the empirical results differ from the findings in the previous studies. This divergence with mainstream studies on the object-oriented paradigm seems, however, to be in agreement with recent findings by other researchers (Johnson, 2000; Johnson, 2002).

However, though the results of this study are in agreement with the results of the studies by Johnson (2000) and Johnson (2002) there are differences in how this study was carried out and how the other two studies were conducted.

Some interesting differences are presented in Table III.

Table III: Differences in studies on object-orientation

| | Present study | Johnson 2000 | Johnson 2002 |
|----------------------|---|--|-------------------------------------|
| Topic | Benefits and problems with the object-oriented paradigm | Benefits and problems with object-oriented systems development | Object-oriented analysis and design |
| Type of survey | Mail | Internet | Study of 12 empirical studies |
| Number of answers | 104 | 150 | - |
| Country | Finland | The US | - |
| Source – Respondents | Statistics Finland | US subscribers to Communications and OOPS Messenger plus registrants at recent OOPSLA conferences (in the US). | Miscellaneous |
| Selection | All in the population | Randomly selected sample | - |

As can be read from the table above there are some interesting differences in how this study was carried out in comparison with the two other studies. The study by Johnson (2002) was very different from this study because it included no pure empirical study, but a study of some empirical studies, and it was focused mainly on object-oriented analysis and design. The study by Johnson (2000) is rather similar to this study although some significant differences exist:

- This study can be generalised over all Finnish software companies (with the exception mentioned earlier in this study) because the population was

all companies. The study by Johnson (2000) can only be generalised over the rather limited selected population.

- This study was a mail survey. The study by Johnson (2000) was Internet based.
- In this study all companies in the population were selected. In the study by Johnson (2000) a sample was selected from the population.
- This study is concerned with software companies in Finland and the study by Johnson (2000) is concerned with companies in the US.

One could compare the studies regarding validity, reliability, generalisation, etc. nevertheless, no such comparison will be made. However, one can conclude from the comparison in Table 3 that this study most probably is an adequate study and in any case not of inferior quality to the study by Johnson (2000).

As a summary, one can state that a major finding from the empirical part of this study is the following:

It is interesting to note how positive the Finnish software companies that took part in the empirical part of this study are towards the object-oriented paradigm. Most of the companies had experienced many of the proposed benefits but had not experienced many of the proposed problems. Out of these results, one can argue that nowadays (2007) the object-oriented paradigm is a leading information systems development paradigm among Finnish software companies, and that the Finnish software companies do not experience significant problems with it.

One could further argue that many of the benefits of the object-oriented paradigm are not always so strong and obvious. One could also further conclude that many of these problems have been solved, or at least partly.

The results of this study will hopefully be used for improving the understanding of how the benefits with the object-oriented paradigm can be realised, and correspondingly, how the problems with the object-oriented paradigm can be avoided in object-oriented information system development in the future.

REFERENCES

- Agarwal, R., De, P., Sinha, A. & Tanniru, M. (2000): On the Usability of OO representations. *Communications of the ACM*, Vol. 43, No. 10, pp. 83-89.
- Basili, V., Briand, L. & Melo, W. (1996): How Reuse Influences Productivity in Object-Oriented Systems. *Communications of the ACM*, No. 10, pp. 105-116.
- Berg, W., Cline, M. & Girou, M. (1995): Lessons Learned from the OS/400 OO Project. *Communications of the ACM*, Vol. 38, No. 10, pp. 55-64.
- Bhattacharjee, A. & Gerlach, J. (1998): Understanding and Managing OOT Adoption. *IEEE Software*, Vol. 15, No. 3, May/June 1998, pp. 91-96.
- Booch, G. (1994): *Object-oriented Analysis and Design with Applications*. Redwood City, California: The Benjamin/Cummings Publishing Company, Inc.
- Caliò, A., Autiero, M. & Bux, G. (2000): Software Process Improvement by Object Technology. 22nd International Conference on Software Engineering (ICSE), University of Limerick, Ireland, 4-11 June 2000. Proceedings of ACM and SigSoft, pp. 641-647.
- Champeaux de, D., Lea, D. & Faure, P. (1993): *Object-Oriented System Development*. Reading, MA: Addison-Wesley Publishing Company.
- Coad, P., North, D. & Mayfield, M. (1995): *OBJECT MODELS Strategies, Patterns & Applications*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.
- Coad, P. & Yourdon, E. (1991): *Object-Oriented Design*. Englewood Cliffs, New Jersey, US: Prentice-Hall, Inc.
- Cockburn, A. (1998): *Surviving Object-Oriented Projects: a Manager's Guide*. Reading, Massachusetts, US: Addison-Wesley Longman, Inc.
- Davis, J. & Morgan, T. (1993): Object-Oriented Development at Brooklyn Union Gas. *IEEE Software*, Vol. 10, No. 1, January 1993, pp. 67-74.
- Eisenhardt, K. (1989): Building Theories from Case Study Research. *Academy of Management Review*, Vol. 14, No. 4, pp. 532-550.
- Gillach, J. & Deyo, N. (1993): Empowering the IT-business relationship with objects. *Object Magazine*, Vol. 3, No. 3, September-October, pp.
- Hatton, L. (1998): Does OO Sync with How We Think? *IEEE Software*, Vol. 15, No. 3, May/June 1998, pp. 46-54.
- Henderson-Sellers, B. & Edwards, J. (1990): Object-Oriented Systems Life Cycle. *Communications of the ACM*, No. 9, pp. 143-159.
- Henderson-Sellers, B. & Edwards, J. (1994): *BOOK TWO of Object-Oriented Knowledge: The Working Object. Object-Oriented Software Engineering: Methods and Management*. Englewood Cliffs, New Jersey, US: Prentice Hall, Inc.
- Jacobson, I., Ericsson, M. & Jacobson, A. (1995): *The Object Advantage: Business Process Reengineering with Object Technology*. Reading, Massachusetts US: Addison-Wesley, Publishing Company. ACM Press.

- Johnson, R.A. (2000): The Ups and Downs of Object-Oriented Systems Development. *Communications of the ACM*, Vol. 43, No. 10, pp. 69-73.
- Johnson, R.A. (2002): An Examination of Empirical Research in Object-Oriented Analysis and Design. *Journal of Computer Information Systems*, Vol. 42, No. 3, Spring 2002, pp. 11-15.
- Kung, D., Gao, J., Hsia, P., Toyoshima, Y, Chen, C., Kim, Y. & Song, Y. (1995): Developing an Object-Oriented Software Testing and Maintenance Environment. *Communications of the ACM*, Vol. 38, No. 10, pp. 75-87.
- Malan, R., Coleman, D. & Letsinger, R. (1995): Lessons from the Experiences of Leading-Edge Object Technology Projects in Hewlett-Packard. Conference on Object-Oriented Programming Systems, Languages & Applications, (OOPSLA)'95, Austin, Texas, US. Proceedings, pp. 33-46.
- Maring, B. (1996): Object-Oriented Development of Large Applications. *IEEE Software*, Vol. 13, No. 3, May 1996, pp. 33-40.
- McGregor, M. (1996): Too Many Cooks could spoil the Broth. Object World UK 1996 Report and directory. EvolveIT.net. Available from: http://markmcgregor.com/art_cook.htm [Accessed 11 October 2002].
- Miah, S. (1997): Critique of the Object-oriented Paradigm: Beyond Object-Orientation. Part of a book. Available from: <http://members.aol.com/shaz7862/critique.htm> [Accessed 11 October 2002].
- Mili, A., Yacoub, S., Addy, E. & Mili, H. (1999): Toward an Engineering Discipline of Software Reuse. *IEEE Software*, Vol. 16, No. 5, September/October 1999, pp. 22-30.
- Noack, J. & Schienmann, B. (1999): Introducing OO Development in a Large Banking Organization. *IEEE Software*, Vol. 16, No. 3, May/June 1999, pp. 71-81.
- Nokso-Koivisto, M. (1995): Oppi tulee kantapään kautta. *Tietoviikko*, No. 35, P. 15. In Finnish.
- Nowicki, A. & Kosiak, M. (1996): Premises of Object-Oriented Approach Adoption in Information Systems. Research Experiences. Wrycza, S. & Zupancic, J. (Eds.). Proceedings of the Fifth International Conference on Information Systems Development, Gdansk/Poland, September 24-26, 1996.
- Pancake, C. (1995): The Promise and the Cost of Object Technology: A Five-Year Forecast. *Communications of the ACM*, Vol. 38, No. 10, pp. 33-49.
- Pickering, C. (1996): *Survey of Advanced Technology 1996*. Overland Park, US: Systems Development, Inc.
- Pomberger, G. & Blaschek, G. (1996): *Object-Oriented and Prototyping in Software Engineering*. Hemel, Hempstead, Hertfordshire, UK: Prentice Hall Europe.
- Radding, A. (1999): Fast track to app success. *InformationWeek*, Manhasset, July 26.
- Radin, G. (1996): Object technology in perspective. *IBM Systems Journal*, Vol. 35, No. 2, pp. 124-127.
- Reeves, C. & Bednar, D. (1994): Defining Quality: Alternatives and Implications. *Academy of Management Review*, Vol. 19, No. 3, pp. 419-445.

- Sheetz, S. & Tegarden, D. (1996): Perceptual complexity of object-oriented systems: a student view. *Object-oriented Systems*, Vol. 3, December 1996, pp. 165-195.
- Smith, H. & McKeen, J. (1996): Object-Oriented Technology: Getting Beyond the Hype. *The DATA BASE for Advances in Information Systems*, Vol. 27, No. 2, Spring 1996, pp. 20-29.
- Taylor, D. (1990): *Object-Oriented Technology: A Manager's Guide*. Reading, Massachusetts, US: Addison-Wesley. Second Printing, January 1992.
- Verity, J. & Schwartz, E. (1991): SOFTWARE MADE SIMPLE; WILL OBJECT-ORIENTED PROGRAMMING TRANSFORM THE COMPUTER INDUSTRY? *Business Week*, September 30, 1991, pp. 58-63.
- Verschoor, R. & Low, G. (1994): Software Reusability in Australia. *The Australian Computer Journal*, Vol. 26, No. 4, November 1994, pp. 134-142.
- Villeneuve, A. & Fedorowicz, J. (1996): Usage Benefits of Object Orientation. Boston University, School of Management, Boston, US. Presented on the INFORMS Conference on Information Systems & Technology, Washington DC, US, May 5-8, 1996.
- Webster, B. (1995): *Pitfalls of Object-Oriented Development*. New York, New York: M&T Books.
- Wieringa, R. (1998): A Survey of Structured and Object-Oriented Software Specification Methods and Techniques. *ACM Computing Surveys*, Vol. 30, No 4, pp. 459-527.
- Wilde, N. & Huitt, R. (1992): Maintenance Support for Object-Oriented Programs. *IEEE Transactions on Software Engineering*, Vol. 18, No. 12, pp. 1038-1044.
- Wilde, N. & Matthews, P. (1993): Maintaining Object-Oriented Software. *IEEE Software*, Vol. 10, No. 1, January 1993, pp. 75-80.
- Winblad, A., Edwards, S. & King, D. (1990): *Object-Oriented Software*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.
- Yin, R. (1994): *Case Study Research. Design and Methods*. Second edition. Thousand Oaks, CA, US: Sage Publications, Inc.

Appendix – Survey results

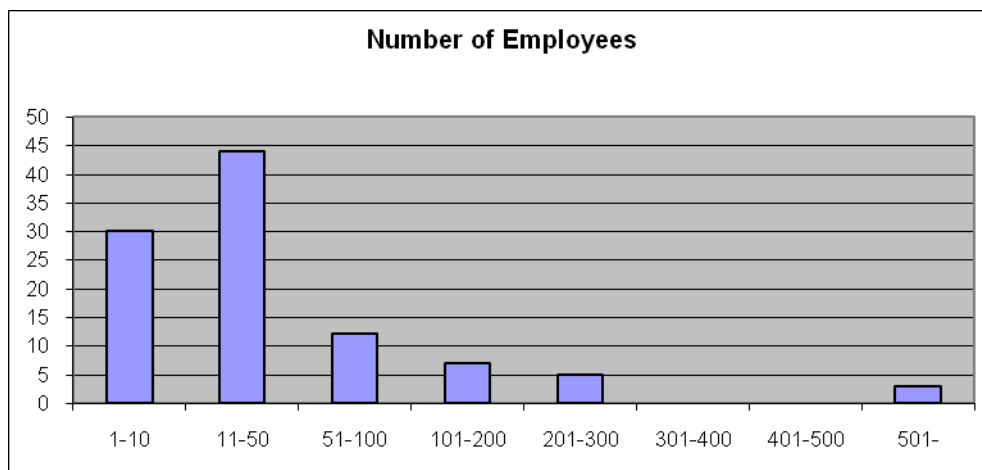
The results were presented and analysed in Section 2. The questions in the questionnaire regarding the company are, however, presented here.

When reading the results of each question one should note that the companies that did not answer the survey question are presented in a “no answer” category. The population in the survey is therefore always the 89 companies that use the object-oriented paradigm in information systems development.

I. Approximate number of employees in your company:

The results are presented in Figure 1.

Figure 1: Number of employees



II. In what field are most of your clients?

The results are presented in Table I.

Table I: Main clientele in the survey

| | Number of answers |
|-------------------------------|-------------------|
| Industry | 15 |
| Telecommunications | 15 |
| Several | 13 |
| Public Sector | 7 |
| Information Technology | 4 |

There were additionally 9 fields with 3 answers and 9 fields with 2 answers and 30 fields with 1 answer.

IV. What is your position in the company?

The results are presented in Table II. There were also 2 positions with 3 answers and 4 positions with 2 answers and 23 fields with 1 answer.

Table II: Position of respondent in the survey

| | Number of answers |
|--------------------------------|-------------------|
| Executive Chief | 16 |
| Production Manager | 8 |
| Software Development Manager | 8 |
| Technology Manager | 7 |
| Manager | 5 |
| Product Development Manager | 5 |
| Project Manager | 5 |
| CTO (Chief Technology Officer) | 4 |
| Program Analyst | 4 |

The research questions will be analysed from a theoretical and an empirical view. The main concern is to compare the empirical findings with the theoretical statements found in the review of previous studies.

In some questions a comparison with traditional information system development is made. With traditional information system development is meant functional information systems development with programming languages like C and Fortran and with traditional design methodologies like JSD.

As many as 89 companies out of 104 use the object-oriented paradigm, and the population N is 89 for the survey in the following research questions. If there are no answers to a survey question this is pointed out in the survey results.

Benefits

(Q1) Has the object-oriented paradigm been found useful when developing large-scale and complex information systems?

Theory – Studies: Due to the experiences of Berg et al. (1995) and the assertions by Booch (1994), Coad & Yourdon (1991, pp. 6-9) and Henderson-

Sellers & Edwards (1994, p. 5) the object-oriented paradigm is useful when developing large and complex systems.

Survey: The results are the following: Yes 94%, No 3% and Not Sure 3%.

Discussion and conclusions: The object-oriented paradigm has been useful when developing large-scale and complex information systems by software companies in Finland with five or more employees. This is the task-related belief that best corresponds to the review of previous studies and theory because as many as 94% of the respondents in the survey were of the opinion that the object-oriented paradigm seemed to be useful when developing large-scale information systems. No other question generated such high a percentage.

(Q2) Has object-oriented information system development been more productive and faster than traditional information system development?

Theory – Studies: Improved *productivity* was an experienced benefit of the object-oriented paradigm in the *Survey of Advanced Technology 1996* (Pickering, 1996). According to Henderson-Sellers & Edwards (1990) object-oriented information system development is faster than traditional information system development. In the results of 12 empirical studies reported by Johnson (2002) better productivity was considered a major benefit.

Survey: The results for the question "*the object-oriented paradigm is more productive*" are the following: Yes 68%, No 21% and Not sure 11%.

The results for the question "*the object-oriented paradigm is faster*" are the following: Yes 53%, No 36%, Not sure 10% and No answer 1%.

Discussion and conclusions: The knowledge and experience of the object-oriented paradigm probably affects the productivity. Because a clear majority of the companies in found object-oriented information systems development as more productive than traditional information systems development this is probably the case among software companies in Finland with five or more employees.

(Q3) Has the quality of object-oriented systems been better than the quality of traditional systems?

Theory – Studies: Due to the object-oriented paradigm, the *quality* of the information system can be improved, because programs are made of existing tested components and not developed from scratch every time (Gillach & Deyo, 1993; Sheetz & Tegarden, 1996; Smith & McKeen, 1996; Taylor, 1990, p. 104). Based on the results of 12 empirical studies reported by Johnson (2002) better quality was considered a major benefit. In the question a more general quality

term is used, although one has to remember that there are several different types of quality (Reeves & Bednar, 1994).

Survey: The results are the following: Yes 70%, No 17% and Not sure 13%.

Discussion and conclusions: A clear majority of the Finnish software companies were of the opinion that object-oriented information systems are of better quality than traditional information systems.

(Q4) Has there been a better and more 'natural' communication between information systems developers and end users because of using the object-oriented paradigm?

Theory – Studies: In the empirical study by Johnson (2000) improved communication with users was found a benefit. In addition, Davis & Morgan (1993) and Gillach & Deyo (1993) propose that using object-oriented software development makes it possible for the users and software developers to speak the same language.

Survey: The results are the following: Yes 22%, No 57% and Not sure 21%.

Discussion and conclusions: A majority (57%) of the companies were of the opinion that there is no better communication between information system developers and end users. It seems rather obvious that the findings by Johnson (2000) in the US cannot be compared with the results from the survey. In Finland the companies are generally smaller than in the US.

(Q5) Has maintenance of object-oriented applications been easier than maintenance of traditional functional applications?

Theory – Studies: Many researchers like Agarwal et al. (2000), Booch (1994, pp. 77-78), Caliò et al. (2000), Johnson (2000), Nowicki & Kosiak (1996) and Radin (1996) argue that maintenance of object-oriented information systems is easier than maintenance of traditional functional information systems. However, researchers like Wilde & Huit (1993) propose that maintenance of traditional functional information systems in reality is easier than maintenance of object-oriented information systems. Hatton (1998) and Wilde & Matthews (1993) propose that the complexity of object-oriented information systems is one reason why they are more difficult to maintain than traditional functional information systems.

Survey: The results are the following: Yes 64%, No 10% and Not sure 26%.

Discussion and conclusions: It appears that the theoretical proposition is true for Finnish software companies. The maintenance of object-oriented information systems is easier than the maintenance of traditional information systems according to a clear majority of the Finnish software companies.

(Q6) Has the object-oriented system development process been seen as a uniform 'one model' from problem domain to code and maintenance in the software company?

Theory – Studies: The object-oriented paradigm has a uniform paradigm throughout development from analysis to implementation and maintenance (Coad et al., 1995, pp. 481-485; Henderson-Sellers & Edwards, 1990).

Survey: The results are the following: Yes 42%, No 42% and Not sure 16%.

Discussion and conclusions: Because of the results one cannot argue that the 'one model' development process is recognized a lot among Finnish software companies.

(Q7) Has reuse in the object-oriented paradigm been as beneficial as several researchers propose it to be?

Theory – Studies: Reuse often results in less rework in the development process (Basili et al., 1996). However, Mili et al., (1999) propose that the benefits of reuse are not always realised.

Survey: The results are the following: Yes 92%, No 3%, Not sure 3% and No answer 2%.

Discussion and conclusions: Because a vast majority of the Finnish software companies had found reuse beneficial, one can only argue that one of the most promising benefits of the object-oriented paradigm.

(Q8) Has portability of the object-oriented system been a benefit?

Theory – Studies: Theoretically portability is considered a benefit of the object-oriented paradigm (Agarwal et al., 2000). The idea is that an object-oriented program can run on every computer with the assistance of a virtual machine. This is the case if, for example, the programming language Java has been used.

Survey: The results are the following: Yes 55%, No 24%, Not sure 20% and No answer 1%.

Discussion and conclusions: The portability of object-oriented information systems is undoubtedly considered a benefit among Finnish software companies.

(Q9) What other benefits than those already presented have the companies experienced in information systems development?

Survey: The following answers were reported:

- Because the object-oriented paradigm is commonly accepted and used, this makes for better cooperation both in the home organisation and between organisations.
- The developer has integrated the theory in his thinking model.
- Generally, the possibility to develop systems that are more sophisticated compared with earlier solutions.
- Good uniformity, logical, good structure and lack of faults.
- Makes it easier to divide the working tasks (implementation of components) among the system analysts / programmers.
- Systematic approach, easier to distribute.
- Easy maintenance.
- Modularity, two answers.
- Information encapsulation.
- One can concentrate better on the task and not on how the task is implemented.
- Management of large code masses becomes easier.
- The development of new objects by inheritance from existing objects.
- The smaller amount of code due to the object-oriented paradigm.
- The modelling of the problem domain. The presentation of the domain and the analysis is easier with objects than with traditional sequential presentation. One has to remember that it is not feasible to make everything into objects.
- UML Design is more important than code in OOP. Design takes longer time but coding is predictable.

Discussion and conclusions: One can analyse the benefits found in the survey and recognize that most of them are connected to benefits that have already been presented. There are also, however, some more detailed benefits and even some rather 'new' benefits like 'Management of large code masses becomes easier'.

Problems

(Q10) Has the object-oriented paradigm been considered complex?

Theory – Studies: According to the findings from the *Survey of Advanced Technology 1996* (Pickering, 1996), the object-oriented paradigm is considered complex. According to the results of 12 empirical studies reported by Johnson (2002) object-oriented design was considered complex.

Survey: The results are the following: Yes 35%, No 58% and Not sure 7%.

Discussion and conclusions: The truth is probably that for an information systems developer with training in the object-oriented paradigm and for an information systems developer with no “burden” of traditional information systems development, object-oriented systems development is not complex.

(Q11) Have the companies considered the object-oriented paradigm as being immature?

Theory – Studies: The object-oriented paradigm is still considered immature by some researchers. Object-oriented projects are often criticized as promising too much and delivering too little (Bhattacharjee & Gerlach, 1998).

Survey: The results are the following: Yes 28%, No 66% and Not sure 6%.

Discussion and conclusions: A substantial majority of the Finnish software companies were of the opinion that the object-oriented paradigm is not immature.

(Q12) Have the companies found testing object-oriented applications or information systems difficult? What testing problems have the companies experienced?

Theory – Studies: There is often little support for testing object-oriented systems in the object-oriented paradigm and in many object-oriented software development methods (Malan et al., 1995). Kung et al. (1995) present major testing problems.

Survey: The results are the following: Yes 29%, No 62% and Not sure 9%.

Discussion and conclusions: Testing object-oriented information systems is not complicated according to most of the Finnish software companies.

(Q13) Has it been difficult to find experienced object-oriented software developers and system analysts?

Theory – Studies: According to Noack & Schienmann (1999) and Radin (1996) it might be difficult to find experienced object-oriented information systems developers.

Survey: The results are the following: Yes 65%, No 22% and Not sure 13%.

Discussion and conclusions: It was found that it is difficult to find experienced object-oriented software developers and system analysts.

(Q14) Have the companies experienced computer efficiency problems in their object-oriented software development projects?

Theory – Studies: There are often problems with computer efficiency because some object-oriented designing takes up a lot of computer processing time (Booch, 1994, pp. 288-289).

Survey: The results are the following: Yes 49%, No 47% and Not sure 4%.

Discussion and conclusions: Probably the truth is in the direction that if not developed correctly it seems that object-oriented information systems often have computer efficiency problems. The computer efficiency issue is probably also associated with the type of information system being developed.

(Q15) Have the starting costs been high when launching a completely new object-oriented information system development project, due to a lack of artefacts to reuse?

Theory – Studies: The starting costs are often huge when one begins a new object-oriented information system project because there is nothing to reuse and everything has to be developed from scratch (Booch, 1994, pp. 288-289).

Survey: The results are the following: Yes 46%, No 37% and Not sure 17%.

Discussion and conclusions: It seems that starting costs are high when launching a completely new object-oriented information system project, due to a lack of artefacts to reuse.

(Q16) Has the company had problems finding components to reuse?

Theory – Studies: Finding the components to reuse is a serious problem in many object-oriented projects. The usability of the components has to be good too, and for example, banks nowadays are defining usable standard business components. Nevertheless, it might still be difficult to find good components to reuse. (Radin, 1996)

Survey: The results are the following: Yes 46%, No 39% and Not sure 15%.

Discussion and conclusions: Probably the reality is that some Finnish software companies have had problems in finding components and some others have had no problems finding components to reuse. This has probably to do with the management of components in the software company in question. Another issue that has to be taken into consideration is the length of time a company has been involved in object-oriented information systems development. The longer the time the smaller the problems in finding components for reuse probably are.

(Q17) Has there been a problem with reuse in the sense that software developers do not want to reuse a component, because they claim that it does not work, or it is too troublesome to learn how the component works?

Theory – Studies: According to Nokso-Koivisto (1995) and Radding (1999) system developers often avoid reusing existing modules, because they claim that the modules ‘do not work anyway’ or ‘it is not worth the effort to figure out what the module (component) does and how it works’.

Survey: The answers on the question "*No reuse because components do not work*" are the following: Yes 28%, No 61% and Not sure 11%.

The answers on the question "*No reuse because troublesome to learn how a component works*" are the following: Yes 28%, No 61% and Not sure 11%.

Discussion and conclusions: It seems that Finnish software companies do not have a problem with software developers not wanting to reuse a component because they feel that it does not work.

(Q18) What other problems or obstacles of the object-oriented paradigm, other than those presented, have the Finnish software companies experienced?

Theory – Studies: Are some of the problems or obstacles the same as those presented by, for example, Taylor (1990, pp. 108-113) or Pancake (1995)?

Survey: This question was divided into two parts (A and B). Not all companies gave an answer. When more than one company gave the same answer this is noted after the answer.

A. Have you experienced a lack of support for any concepts in the object-oriented world? The results were the following:

Tools:

- Lack of good automatic testing tool that is integrated with the design and would make it possible to systematically test the application and generate test cases.
- Lack of ready object-oriented and cheap tools that small companies can use.
- UML is not suitable for the communication between software developers and end users.

Documentation:

- The documentation of classes and solutions in IDE is often not sufficiently connected to reality.
- No systematic approach for documentation.

Programming and design:

- The move from functional/data oriented programming to the object-oriented paradigm has been hard.
- Good class hierarchy: usually 3-4 iterations before one can get it right.
- Hard to build components in some circumstances.

General:

- The lack of understanding and skill to compare software systems and architectures among the managers.
- Lack of clear guidelines.
- Lack of good and skilful developers.

B. What other problems or obstacles of the object-oriented paradigm, other than those presented, have you experienced? The results were the following:

Programming and design:

- Hard to present and design the sequence of the messages.
- Problems with the usage of objects that are only needed in certain circumstances.
- The development and evaluation of components takes a lot of time (and costs).

Tools:

- The object-oriented tools have shortages. Testing could be improved.
- The prices of the tools are usually too high for our small company.
- High prices on system tools and object-oriented tools. The object-oriented world has a rather good theoretical background, but there is a lack of CASE tools.

General:

- Customers do not always like it because of its complexity.
- First it was difficult to come into the 'object world thinking'.
- Resistance against change to the object-oriented paradigm.
- The object-oriented paradigm is not a "miracle medicine"; with the object-oriented paradigm one can develop just as bad or as good solutions as without it. All depends on the working skills of the developer.

Discussion and conclusions: Many problems connected with the object-oriented paradigm were presented. Most of the problems were, however, variations of the problems with the object-oriented paradigm that are presented earlier in this study. No generalisations can be made.