Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1998 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1998

What Developers Believe About OOSD: An Empircal Study

Richard Johnson
Southwest Missouri State University

Bill Hardgrave
University of Arkansas Fayetteville

Follow this and additional works at: http://aisel.aisnet.org/amcis1998

What Developers Believe About OOSD: An Empirical Study

Richard A. Johnson
CIS Department
Southwest Missouri State University

Bill C. Hardgrave
CISQA Department
University of Arkansas, Fayetteville

Introduction

Object technology (OT) is a relatively new branch of information technology (IT) that is based on the object paradigm. A major component of OT is object-oriented systems development (OOSD), a process that many believe to be far superior to conventional systems development (CSD) (Rumbaugh et al. 1991, Coad and Yourdon 1991, Booch 1994, Coleman et al. 1994, Jacobson et al. 1995). One concern with OOSD is the shortage of hard empirical evidence either for or against it.

The purpose of this study is to explore the validity of many beliefs about OOSD using a large random sample of experienced systems developers. Additionally, beliefs held by developers who are experienced in OOSD are compared to those who are not. Through such a study, the most significant and substantial beliefs can be identified, and the effect of experience in OOSD on such beliefs can be determined.

Background

Beliefs About OOSD

The OO literature has provided much in the way of expert opinion concerning beliefs about OOSD (Rumbaugh et al. 1991, Coad and Yourdon 1991, Booch 1994, Coleman et al. 1994, Jacobson et al. 1995, Pancake 1995, Korson and McGregor 1990, Fayad et al. 1996, Fayad et al. 1994, Fichman and Kemerer 1993), the vast majority of which is very positive. A sample of many of the most commonly reported beliefs can be grouped in various categories as follows:

Advantages of Using OOSD
- Easier modeling of real-world systems
- More understandable analysis and design models
- Easier transition from analysis to design to implementation
- Improved communication between developers and users, and among developers
- More flexible development using software components
- Less development time
- Improved forms of modularity of code
- Increased reuse of analysis and design models
- Increased reuse of program code
- More stable designs
- Less maintenance and easier maintenance
- Greater developer productivity
- Improved quality of completed systems

Disadvantages of Uing OOSD
- Poorer run-time performance
- Longer development time
- Difficulty of learning OOSD
- Relative immaturity of OOSD

Factors Facilitating the Use of OOSD
- Management commitment to OOSD
- Availability of OOSD training
- Open-mindedness toward new technologies
- Availability of OO mentors and champions

Factors Inhibiting the Use of OOSD
- Confusion of multiple OOSD methods

- Hybrid OO programming languages
- Ingrained experience with conventional methods

Theoretical Foundation

Beliefs about OOSD can be expected to contribute significantly to developers' satisfaction with and acceptance of OOSD. In turn, the acceptance of OOSD should enhance its actual use. In support of this view is a very well-known general theory of industrial and organizational psychology, Ajzen's (1988) theory of planned behavior (TPB). The TPB suggests that beliefs and perceptions influence one's intention to perform a specific target behavior. The three categories of beliefs and perceptions found in the TPB include the usefulness (U), the social acceptability (SA), and the ease of use (EOU) relative to the target behavior under investigation. The target behavior in this study is the use of OOSD methods.

Methodology

Sample

In order to assess developers' beliefs about OOSD, a random sample of developers was obtained from subscribers to the *Communications of the ACM*. This sample was limited to those who belong to certain relevant special interest groups (SIGs) associated with systems development (e.g., software engineering, programming, MIS, and OIS). In addition, to ensure the respondents included a sizable number of developers with experience in OOSD, part of the sample was drawn from subscribers to *OOPS Messenger*, and from registrants at recent *OOPSLA* conferences. Only individuals who considered themselves to be actively involved in systems development were invited to complete a mail survey.

Survey Instrument Development

A survey instrument was prepared listing various types of beliefs on usefulness, social acceptability, and ease of use of OOSD. These beliefs were drawn from the extensive OO literature and from a prior pilot study of OO developers. Respondents were asked to record the strengths of these beliefs using Likert-type scales ranging from 1 (strongly disagree) to 7 (strongly agree) where 4 is neutral. Respondents were also asked to indicate their level of familiarity with OOSD. The possible responses ranged from 1 (very unfamiliar with OOSD) to 7 (extensive experience with OOSD on actual projects). Several other items were included in the survey to compare the demographics of the sample to the population.

Results

The survey yielded a final sample size of n=150. Using the chi-square test for equal distributions, the sample demographics were carefully compared to population demographics based on six different descriptive variables (age, education, development experience, company size, job function, and use of OO). For each of these variables, there was no significant difference (at the $\alpha = 0.05$ level) in the distributions of the sample and the population. Therefore, the sample appears to be very representative of the population of systems developers.

Table 1 provides results showing (a) the most substantial, statistically significant beliefs (belief strength $\alpha \geq 5.0$, $\alpha = 0.05$) and (b) the statistically insignificant beliefs ($\alpha = 0.05$) for the entire group of 150 developers. Table 2 provides those beliefs that exhibit a significant ($\alpha = 0.05$) difference in mean strength between those experienced and inexperienced in OOSD.

Discussion

The substantial items in the usefulness (U) category confirm many of the advantages of OOSD found in the OO literature. In particular, developers believe that using OOSD results in certain *task-related* benefits including better modeling, improved modularity, and improved quality and maintainability of systems. In addition, developers perceive certain *personal* benefits resulting from using OOSD including increased marketability and job satisfaction.

All the items in the social acceptability (SA) category are considered substantial with the exception of SA8 (i.e., the approval of customers/users of your software). It is likely that developers believe that customers/users of software care little about *how* software is developed (CSD vs. OOSD) as long as it meets their requirements.

The substantial items in the ease of use (EOU) category relate to the personal motivation of the developer, exposure to different methods, and the suitability of OOSD methods to projects. All such factors would be expected to make OOSD easier to use.

Table 1(b) indicates that there are three non-factors relative to the use of OOSD: unrealistic user expectations for OOSD, confusion of multiple OOA/D methods, and the affordability of OOSD training, tools, etc. The first of these three corroborates a finding from Table 1(a) that, in the view of developers, user expectations are not a major concern in the application of OOSD. Also, developers do not appear to be concerned about the abundance of OOA/D methods and OO programming languages. Developers apparently have little concern about the costs of migrating to OOSD (although managers might), especially if they are as thoroughly convinced of its benefits as evidenced by Table 1(a).

Table 2 sheds some light on differences in the beliefs of those with and without experience in OOSD. Beginning with the usefulness question, developers experienced in OOSD have stronger beliefs in the ease of modeling systems with OOSD, in the

improvements in quality, maintenance, and productivity, and in user and developer satisfaction. With respect to social acceptability, the experienced users of OOSD expect higher levels of approval from coworkers, peers, supervisors, and customers than their CSD counterparts. Also of interest are the differences in items relating to the ease of use of OOSD. The experienced users of OOSD are generally more experienced in a wider variety of development methods, are more motivated to use OOSD, and are experiencing greater management commitment to OOSD.

Conclusion

This study has been largely successful in determining which specific beliefs about OOSD are more important to software developers in general, and in comparing the beliefs of those inexperienced with those experienced in OOSD. The belief strengths of 150 experienced developers were averaged and subjected to a t-test for significance. The findings suggest that a large number of the beliefs about OOSD use, in the areas of usefulness, social acceptability, and ease of use, are held quite strongly by all systems developers.

It appears that the most substantial benefits of OOSD exist in the area of developer effectiveness, as opposed to developer efficiency. Management should consider the adoption of OOSD in view of the strength of the beliefs regarding its usefulness. Management commitment to OOSD is viewed as more important by experienced OO developers. While adopting a new process such as OOSD may be viewed as somewhat challenging, actual experience tends to strengthen beliefs in its payoff.

References
References available upon request from first author.

Table 1(A). Most Substantial Beliefs on OOSD (4=Neutral)

Beliefs about OOSD compared to CSD	Mean Strength	Std. Deviation	t-Statistic
U1. An easier, more natural modeling process.	5.50	1.42	12.96
U3. An easier transition from OOA through OOP.	5.10	1.41	9.53
U4. More understandable analysis and design models.	5.19	1.37	10.63
U5. More effective code reuse.	5.28	1.52	10.28
U7. Improved modularity of systems.	5.63	1.38	14.45
U8. Improved quality of systems.	5.36	1.33	12.50
U9. Improved maintainability of systems.	5.47	1.33	13.57
U11. Improved communication with developers.	5.02	1.50	8.33
U14. More flexible/adaptable development.	5.23	1.34	11.24
U15. Greater stability of designs over time.	5.35	1.33	12.45
U19. An improvement in your marketability.	5.49	1.40	12.97
U20. An increase in your job satisfaction.	5.41	1.48	11.61
SA1. The approval of coworkers in your organization.	5.47	1.33	13.52
SA2. The approval of peers outside your organization.	5.51	1.15	16.11
SA3. The approval of authors (on computing) that you read.	5.89	1.03	22.50
SA4. The approval of your supervisor/manager.	5.25	1.49	10.28
SA5. The approval of consultants /trainers you encounter.	5.66	1.12	18.12
SA6. The approval of vendors you encounter.	5.33	1.14	14.26
SA7. The approval of your potential future employers.	5.76	1.07	20.10
EOU1. Your interest in trying new technologies.	5.98	1.05	23.05
EOU2. Your exposure to different development methods.	5.46	1.19	15.02
EOU4. Your openness to try OOSD methods.	6.11	1.02	25.37
EOU9. Your knowledge and skill in OO programming.	5.13	1.51	9.13
EOU10. Your personal drive to use OOSD.	5.44	1.48	11.91
EOU20. Suitability of OOSD methods to your projects.	5.29	1.58	10.04

Table 1 (B). Statistically Insignificant Beliefs (4=Neutral)

Beliefs about OOSD compared to CSD	Mean Strength	Std. Deviation	t-Statistic
EOU11. The confusion of too many OOSD methods.	3.91	1.64	-0.70
EOU19. Affordability of OOSD training, tools, etc.	3.98	1.73	-0.14
EOU21. Unrealistic user expectations for OOSD.	3.88	1.62	-0.91

Table 2. Beliefs with Significant Differences Between OOSD-experienced (Exp) and OOSD-inexperienced Developers (Inexp) (0=Neutral)

Beliefs about OOSD compared to CSD	Difference of Means (EXP-INEXP)	Std. Deviation	t-Statistic
U1. An easier, more natural modeling process.	0.83	0.25	3.33
U3. An easier transition from OOA through OOP.	0.81	0.23	3.49
U4. More understandable analysis and design models.	0.87	0.23	3.72
U8. Improved quality of systems	0.57	0.23	2.48
U9. Improved maintainability of systems.	0.58	0.23	2.57
U13. Improved productivity of your work.	0.53	0.26	2.08
U14. More flexible/adaptable development.	0.68	0.24	2.86
U17. Greater user satisfaction with systems.	0.83	0.26	3.18
U20. An increase in your job satisfaction.	0.76	0.27	2.84
SA1. The approval of coworkers in your organization.	0.53	0.22	2.36
SA2. The approval of peers outside your organization.	0.50	0.19	2.58
SA4. The approval of your supervisor/manager.	0.58	0.25	2.27
SA8. The approval of customers /users of your software.	0.35	0.18	1.96
EOU2. Your exposure to many different development methods.	0.56	0.20	2.76
EOU4. Your openness to try OOSD methods.	0.58	0.18	3.20
EOU5. Your difficulty in learning OOA/D methods.	0.82	0.27	3.04
EOU6. Your difficulty in learning OO programming.	0.80	0.27	2.93
EOU7. Your experience with OOSD methods.	1.34	0.25	5.32
EOU8. Your knowledge and skill in OOSD.	1.52	0.25	6.00
EOU9. Your knowledge and skill in OO programming.	1.23	0.26	4.71
EOU10. Your personal drive to use OOSD.	1.07	0.25	4.27
EOU14. Management commitment to OOSD.	0.63	0.29	2.20
EOU20. Suitability of OOSD methods to your projects.	1.09	0.27	4.03
EOU21. Unrealistic user expectations for OOSD.	0.58	0.27	2.17
EOU27. Ability to demonstrate the benefits of OOSD.	0.87	0.29	2.97