Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1999 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1999

Adopting Object-Oriented Development: One Company's Experience

Bill Hardgrave
University of Arkansas

Adopting Object-Oriented Development: One Company's Experience

Bill C. Hardgrave, University of Arkansas, whardgra@comp.uark.edu

Abstract

In 1997, AA Company, made their initial move to OO development. Hoping to achieve success in a client-server environment, AAC initiated a phased pilot project using Java as the language. To date, these projects could be considered failures. What happened? This paper chronicles the events of OO adoption by AAC and discusses some things, in retrospect, that should have been done differently.

Introduction

AA Company (AAC) is a $1.5 billion publicly held service organization (largest in its industry) with about 12,000 employees nationwide (note: company is real; name is changed to hide true identity). IS development is conducted internally with company employees (some contract labor is used, but sparingly). The majority of the approximately 150 developers are located in one location. AAC made the 1997 Computerworld=s Premier 100 Users of Internet Technology, and was listed in the top 15 of the 1998 Computerworld=s Best Places to Work in IS. AAC=s development environment is very traditional, characterized by IBM mainframes, dumb terminals and PC workstations, DB2 relational database, CICS, and COBOL.

About four years ago (circa 1994), AAC began a move to client-server computing. In many ways, this marked the beginning of their move to object-oriented development. Dissatisfied with traditional development efforts in a 3-tier environment, AAC began to explore alternatives. By 1997, the company had tried several different ways to produce systems in a client-server environment. In mid-1997, AAC turned to OO as a possible solution and initiated a 4-phase pilot project.

After almost 18 months, the OO pilot project is on hold. It is not clear whether AAC will continue their OO efforts. This paper provides a chronology of events in AAC=s attempt to adopt OO. An assessment of AAC=s adoption process is provided.

The "Move" to OO

The move to OO began in August 1997 and was put on hold in December 1998. The following provides the chain of events during this time period.

August 1997: By August 1997, the internet was playing a major role in AAC=s business. Unsuccessful in previous attempts at a true 3-tier architecture, AAC decided Java would be the way to go for 3-tier. They sent a few developers to Java training and, upon their return, began a 4-phase internet project using this new approach (hereinafter Internet Project).

October 1997: It took about two months for AAC to realize that the limited Java training received by the small group of developers was not sufficient to develop the first phase of the 4-phase project. Thus, AAC brought in a consultant from a major computer company to begin work on phase 1 of the project. Basically, the consultant did the front-end (GUI) design and wrote the Java programs; the AAC developers did all the "back-end" work connecting the GUIs via CICS/COBOL to the DB2 database. No OO training was provided; no formal OO methodology was followed; no OO design tools were purchased. IBM=s Visual Age for Java was the programming tool used.

May 1998: The Internet Project, started in October 1997, was put into production in May 1998. The system used Java applets to communicate to servlets on a web server which, in turn, communicated with an IBM mainframe (accessing a DB2 database). Internal (i.e., on-site) users of the system were pleased. Remote users, however, were not so happy. Loading the applet was taking an average of 2.5 minutes - obviously unacceptable. AAC decided not to extend the contract with the current computer company beyond phase 1 of the project. No useful documentation, in the form of OO design documents, was provided by the consultant.

Several members of management felt that OO was not given a fair chance to succeed. No clear goals for OO development were specified before beginning the project, no plans were made concerning the adoption of OO into the AAC environment, and good OO design practices were not followed by either the consultant or AAC developers. Thus, within a few weeks, AAC brought in a second consulting firm with the purposes of: (1) providing a good OO design; (2) producing and providing good OO design documents; (3) providing a proof of concept of OO for AAC (i.e., can OO work in AAC=s environment?); and (4) creating an architectural prototype to demonstrate the efficiency of a 3-tier architecture. The plan was to have the new consulting firm work on phase 2 of the project while AAC developers reworked phase 1 to improve performance.

Management was hopeful that phase 2 would provide the necessary information to make the decision whether or not to continue with OO as a development approach. AAC developers would be actively involved with the consultants on phase 2. If successful, the knowledge gained on phase 2 could then be used to finish phases 3 and 4 using only AAC developers.

A new development tool, Visual Café, was chosen to replace Visual Age for Java.

Four AAC personnel were assigned to phase 2: two developers (one of the developers was involved with phase 1), one technical services person, one quality assurance person, and a project manager. Also at this time, an OO 'advisor' was brought in to work with AAC on the project. The advisor was asked to: (1) prepare an OO adoption strategy for AAC; (2) train and mentor the new development team; and (3) oversee the work performed by the consultants. The OO adoption strategy was prepared and endorsed by management by the end of May.

June 1998: The consulting company was on-site for the first week; all other work was performed off-site. The project deadline was set for August. OO design documents were to be delivered incrementally as they became available.

July 1998: The OO Advisor began training the new development team during the first week of July. The first training session consisted of OO concepts. Subsequent training and mentoring were designed to mirror the deliverables (i.e., "just-in-time" training). For example, the first deliverable was a use case model. Thus, the development team was taught about use cases. The learning was then applied to the documents received from the consultants. At this point, the OO Advisor became a mentor to the team in analyzing and modifying the documents. This type of training/mentoring process occurred for each of the deliverables (e.g., class diagrams, state diagrams, sequence diagrams, etc.) during the month of July. This turned out to be a great method for the development team to learn OO development.

August 1998-September 1998: During August and September, the consulting company continued working on phase 2 of the project. As design documents were created and/or modified, they would be sent to AAC for evaluation. AAC would, in turn, approve or request modifications.

AAC developers, in addition to working with the consultants on phase 2, reworked phase 1 to improve performance.

The original deadline for phase 2 for the end of August was not met. Tension between the consulting company and AAC grew as the project became stuck in a design loop (design -> modifications).

October 1998: After failing to meet original objectives and falling almost two months behind schedule, the consultants were fired. Although the consulting company agreed to finish the project "for free" (i.e., no additional charges beyond the original estimate), the finished project was far from acceptable.

November 1998: The development team attempted to continue with phase 2 on their own. Unfortunately, good OO design practices were quickly jettisoned in favor of their traditional methods of immediately writing (and re-writing) code until it worked. Thus, the project reverted back to a Java programming project. The project was continuing, but not based on the design. Code was written and modified as needed. The consulting company was no longer involved; the OO Advisor was rarely involved.

December 1998: The best and most experienced OO developer of the development team (he was involved in phases 1 and 2) leaves the company.

Current: The project is currently on hold. Phase 2 was completed in December. Original specifications were met, but performance is still an issue. The resulting Java code has little resemblance to the original design. The second developer is back to her original job; the project manager is not actively involved; the quality assurance person is no longer involved; and top management is not pushing it. It is unclear whether or not the company has given up on OO.

Lessons Learned

To this point, OO development efforts at AAC would have to be considered failures. Two phases of a 4-phase project are "complete", but not satisfactorily. What went wrong? In retrospect, there are several things AAC could have done differently, as discussed in the following sections.

<u>Perceptions of OO:</u> AAC adopted OO hoping to solve their client-server development problems without properly studying the problem and making a determination of the worthiness of OO. They viewed OO as the "silver bullet." Also, AAC equated the use of Java as the adoption of OO. Furthermore, they viewed Java as just another language to be learned by any developer. Overall, they perceived OO to simply be a programming language that would cure all of their development ailments.

The Pilot Project: The pilot project chosen by AAC to evaluate OO was appropriate in many ways, but inappropriate in one major way. The pilot project followed popular guidelines (e.g., Cockburn, 1998; Henderson-Sellers and Edwards, 1994): the 4-phases were interrelated, the system was important but not critical, and four people plus a mentor staffed the project (for phase 2). Unfortunately, the project was calendar driven, which, according to Booch (1996) is a "recipe for disaster."

Planning: Well after the pilot was started (after phase 1, but before phase 2), AAC developed an adoption plan. Unfortunately, many mistakes which were difficult to recover from had been made by this time. The lack of a plan is consistent with their improper perception of OO (discussed earlier). A proper plan detailing the steps, activities, and measures should be developed prior to the development effort (Booch, 1996).

Long-Term vs Short-Term: OO is not a short-term solution. AAC was looking for immediate solutions to their problems. Taylor (1992) and Booch (1996) suggest OO will cost you more for the first few projects (compared to traditional techniques); it will take about three projects to start realizing benefits.

Organizational Commitment: Although several members of management wanted to experiment with OO, and at times top management supported the initiative, the support was neither strong nor sustained. No clear champion emerged. For OO, and most any other innovation, a champion (i.e., executive sponsor) is needed (Cockburn, 1998).

Training: AAC provided only minimal training for phase 1 (Java training for one of the developers). Beginning with phase 2, the 4-person development team received "just-in-time" training, as recommend by Cockburn (1998). This strategy seemed to work very well for the group. However, AAC did not provide adequate time for training and subsequent learning. All training was provided in a one-month period of time. Sufficient time was not allowed for the subsequent learning process. Normally, it takes about 6 to 9 months for a developer to fully embrace OO and become productive (Booch, 1996; Jones, 1995).

Measuring Success: AAC wanted OO to solve the problems they were having in a client-server environment. However, the objectives of the pilot project were never explicitly stated by management and no clear indicators of success were provided. Management viewed the pilot project as a failure due to performance issues and lack of productivity. Performance was an initial concern, so it could be assumed that it was an implied measure. Developers often stated they did not know how they or the project were to be evaluated.

Conclusion

Many companies are currently making the move to OO. It is estimated that 80% of organizations will be using OO by 2001 (Computerworld, 1994). However, many of these companies are struggling to adopt and integrate OO. This paper has chronicled one company's efforts to adopt OO. Several mistakes were made by AAC early in the adoption process. Currently, the OO pilot project is on hold and it is not clear whether AAC will continue its efforts to adopt OO.

References

Booch, G., Object Solutions: Managing the Object -oriented Project, Reading, MA: Addison-Wesley, 1996.

Cockburn, A., Surviving Object-Oriented Projects: A Manager=s Guide, Reading, MA: Addison-Wesley, 1998.

"Corporate IS Considers Object Plans," *Computerworld*, 28 (4), Jan 24, 1994, 61, 64.

Henderson-Sellers, B., and Edwards, J., BOOKTWO of Object-Oriented Knowledge: The Working Object, Sydney, Australia: Prentice-Hall, 1994.

Jones, N., "The OO Skills Crisis," Research note: SPA-700-1250, published by the Gartner Group, November 29, 1995.

Taylor, D.A., Object-Oriented Information Systems: Planning and Implementation, New York, NY: John Wiley & Sons, Inc., 1992.