

# The critical need for real on-the-job training

**W**HEN DR. RICHARD WIENER called me yesterday and invited me to write a guest article in this special issue of JOOP, he made the common editorial request for something a little controversial. I am therefore taking this opportunity to write something somewhat iconoclastic (although unfortunately also somewhat questionable from a marketing standpoint).

But first, let me give you some background that should help justify some of my observations and recommendations. Since 1986, I have made my living largely from providing classroom training in object technology. I have provided both public and in-house training through my own company as well as through four other training companies. In these courses, I have trained managers and software engineers in my own method using my own materials as well as several other well-known methods using training materials developed by other companies. In turn, I have been trained by other trainers (e.g., by sitting in on as many tutorials at conferences as I could). It is therefore with some trepidation that I write that there is a fundamental limit on the value of classroom training and conference tutorials.

### THE PROBLEM

Such classroom training can only provide initial information; it cannot significantly develop skills in the short time allocated, no matter how many exercises are included. Classroom training also cannot provide the transformation of information into knowledge that comes via practice and experience. And it is practice, experience, and the development of skills that are most important when it comes to overcoming the substantial learning curve associated with mastering any new paradigm. Do not misinterpret me; organized classroom training is critical because it lays the necessary foundation upon which skills are later developed. But, it is inadequate by itself. And any training organization that promises to transition students into object-oriented software engineers solely via classroom training is, at the very least, significantly overstating the benefits of their courses.

Unfortunately, there are still a great many managers who believe

that a week of object-oriented analysis and design followed by a week of C++ training for their developers will do the job. They are mistaken. On average, it takes several months for software engineers (and managers) to master the new mindset and develop the skills necessary for using this mindset effectively. It therefore takes several months for the associated quality and productivity advantages of object technology to appear. Therefore, it is the manager's responsibility to ensure that this transition occurs as smoothly and as rapidly as is practical. And this requires significant, formal, on-the-job training. Learning by osmosis while working on one's first project is both inefficient and results in software being developed and delivered that is of poorer quality than necessary. Yet, many developers are buried so deep in the trenches under such inadequate schedules that they do not feel they have time to keep up with the rapidly evolving technology. Thus, many developers with *X* years of experience often really only have a single year of experience repeated *X* times. Such developers often feel, with some justification, that if management really wanted them to learn, they would provide the necessary training and resources such as books, journals, and conference proceedings. Unfortunately, managers often feel, also with some justification, that, as professionals, software engineers should know what they are doing and keep up with the technology on their own.

### THE SOLUTION

The solution is to recognize that developers (and managers) progress most rapidly under an organized apprenticeship program in which masters train journeymen and apprentices. I recommend approximately 1 master for every 5 journeymen and 15 apprentices. Instead of using the most experienced experts solely as chief analysts or designers, they also should be assigned significant mentoring duties. Experts will achieve their greatest positive impact on the project if they are also used as force multipliers who improve the skills of their assigned journeymen and apprentices.

Finding apprentices who wish to learn object technology is easy. The problem is finding the necessary number of journeymen and masters who have achieved that status through years of

experience using object technology on real projects. Whereas a single master may be assigned part-time to support multiple projects (if adequate journeymen exist), journeymen must be assigned full-time, often in the role of technical manager. After all, technical managers cannot lead their staffs into new and better ways of doing business if they do not understand the technology themselves. One may either scatter the few such experts one has among all of one's object-oriented projects or hire experienced people (which is difficult because so many organizations are transitioning to object technology that the supply of experts does not meet the demand).

But what do you do if you do not have and cannot hire enough masters and journeymen to make an effective apprenticeship program? You can get help from the many training and consulting companies identified in this issue of JOOP. If possible, have your initial classroom training be project specific, and use the project as the basis of the exercises. During object-oriented analysis and design training, analyze and design parts of the project. During language training, code those parts of the project that were analyzed and designed in the previous course. In this way, the exercises gain relevancy, both management and the students see the practicality of what is being learned, and a bigger training budget can be justified because real work is also being done under the supervision of the outside masters.

Although an excellent start, such classroom training must be followed by ongoing consulting if local masters are not available. For example, back in the early 1980s I was the software methodologist for a very large (1.3 MSLOC) object-based application and ran the associated OOD help desk. Although I had read Grady Booch's first book, *SOFTWARE ENGINEERING WITH ADA*, and had gone through the same externally provided 4-week classroom training program as the developers, I did not have any more real experience in object technology than anyone else on the project. It was as if I were in the Red Queen's race in *ALICE IN WONDERLAND*—I had to run as fast as I could to stay just a little ahead of the developers who daily came to my desk seeking help with their designs. I only succeeded because my management had arranged for me to obtain a certain number of hours of phone consultations per month from the training company and had also arranged for them to send one or two consultants to our project once a month for the first few months. I recommend the same for any project that cannot obtain access to enough local masters and journeymen for adequate real on-the-job training.

Another technique that can be quite effective for ongoing training is to have weekly training sessions in object technology. The developers can bring their lunches to a conference room, and the masters and journeymen can make a short (e.g., 45 minute) presentation on some topic in object technology. Although it takes significant time to organize and prepare for such brown-bag lunch seminars, they are an excellent and relatively inexpensive way of supplementing one-on-one mentoring and keeping the organization up-to-date on the latest developments.

Managers should also understand that this article applies to them as much as it does to their developers. Object technology has many impacts on management, especially in the areas of estimation, metrics, and development cycles and their impact on

## Real object-oriented shops

*continued from page 51*

skills.) Then, although everyone will have a working familiarity with all the object-oriented trades, key specialists will take primary responsibility for each key area.

If you're currently a programmer, think about the role(s) that will become *your* specialties once object orientation is up and running in your shop. How will you acquire the necessary expertise?

No longer can the planning of training be the mindless exercise of "we'll just send everybody to everything." (Jerry Weinberg once called this the "sheep dip" approach to training.) Personnel who are all things to all projects are object-oriented dilettanti who are likely masters of nothing. But shops who continue to educate their people as though they were a homogeneous horde doing "software stuff" are probably just toying with object orientation anyway.

And by the way—managers take heed!—it takes more than a one-week class for anyone to become proficient in anything, especially C++. But that's the subject of another article. . .

---

*Meiler Page-Jones is President and Senior Consulting Methodologist at Wayland Systems Inc. in Bellevue, WA, and co-author of their courses on "The Synthesis Method for Object-Oriented Development" and "Methods for Client/Server Systems Development." He is the author of PRACTICAL GUIDE TO STRUCTURED SYSTEMS DESIGN and PRACTICAL PROJECT MANAGEMENT. Currently, he carries out training and consulting at many major corporations. He can be reached by email at 76334.1247@compuserve.com.*

---

scheduling, staffing, and formal reviews. Managing an object-oriented project or team requires the development of new skills and involves its own learning curve. Experienced managers (and they are even more scarce than experienced developers) should be assigned mentoring responsibilities over managers new to object technology. Consultants can also be used to support managers as well as developers.

## **CONCLUSION**

Classroom training must be supplemented with formal on-the-job training based on mentoring provided by masters and journeymen who have achieved that status through real experience on real projects. Although adequate classroom training is critical, it only lays a foundation of information. The real training takes place on the job, when necessary skills are developed and information is transformed into knowledge via practice and experience.

---

*For the past decade, Donald G. Firesmith has been providing consulting and training in object technology. He developed and maintains ADM, a fourth-generation object-oriented system and software development method, and OOSDL, a second-generation object-oriented specification and design language. He is a regular columnist for the REPORT ON OBJECT ANALYSIS AND DESIGN (ROAD), and the author of several books on object technology. He can be reached at Advanced Software Technology Specialists, 2910 Webster, Fort Wayne, IN 46807, 219.745.7928; email: 73664.3515@compuserve.com*

---