

# "Managing Object Oriented Projects"

## Panel Session

Laura Hill, Kenny Rubin, John Daniels, Charles Berman, James Coplien, Doug Johnson

**Abstract:** Projects using Object Oriented technology typically fall into one of two categories: either they are run with traditional project management techniques, or they are run by "technology experts" who don't use *any* project management techniques. This panel will attempt to give out some hints as to how to be more successful managing OO projects. Questions to be addressed include:

- How is Object Oriented Project Management different? Is a "paradigm shift" required for project management?
- What aspects of Object Oriented project management should be stressed? What are some tips, tricks or gotchas experienced by the panel members?
- Does a project manager need to be more technical to lead an Object Oriented project? Should he/she have had previous experience with Object Oriented development?
- What are some alternate organizational structures which would better support Object Oriented projects?
- What is the nature of an Object Oriented life cycle and how is it different?
- Is there a difference in techniques between different languages or between different domains

Audience members with relevant experience will be invited to sit on the panel in two "guest" seats.

### Kenny Rubin, ParcPlace Systems

[krubin@parcplace.com](mailto:krubin@parcplace.com)

When we develop software using object technology, we must pay attention to the classic areas of project management: planning, organizing, staffing, directing and controlling. Our goals in each area remain unchanged when we use object technology. We are still concerned with planning and scheduling the project activities, selecting the appropriate staff, directing the staff towards a successful outcome, and controlling the project according to the schedule and handling anomalies as they occur. As such, a project that is lead by an "object technology expert," without basic project management skills, is a prescription for failure.

Although basic managerial goals remain unchanged when we use object technology, the specific ways in which we achieve the goals are changed. Object technology affects the software development processes, resources, and products. As such, a project lead by an experienced project manager, who does not have an understanding of object technology, is likely to fail due to ignorance surrounding required technical and organizational change. The best managers of object projects have good fundamental project management skills, and appreciate the influence of object technology on software development processes, resources, and products.

### Biography

Kenny Rubin is Manager of Methodology Development at ParcPlace Systems, Inc., where he leads the creation of the Object Behavior Analysis and Design (OBA/D) method and tools products. Mr. Rubin has over ten years experience working on and managing software development project using object-oriented technology. He is also the co-author, along with Adele Goldberg, of the book *Succeeding with Objects: Decision Frameworks for Project Management*, and the author of over 30 publications on the topics of managing object-

oriented projects, object-oriented analysis and design, artificial intelligence and human-computer interaction. Over the past five years, Mr. Rubin has presented tutorials, workshops and papers at over 100 conferences and companies world wide.

Mr. Rubin is a member of the Editorial Board of Object Magazine and a faculty member of the Stanford University, Western Institute in Computer Science (WICS). He received his BS. in Computer Science from the Georgia Institute of Technology and his MS. in Computer Science from Stanford University.

### **John Daniels, Object Designers Limited**

*jdaniels@cix.compulink.com*

I don't think managing a project that uses object technology is really much different to managing any other project. Much of the problem lies with managers believing it should be different. Here are some conclusions I've drawn from my exposure to many OT projects:

- Every project manager must understand the technology being used on their project.
- Evolutionary delivery is the best way of reducing project risks.
- Each cycle in an evolutionary development must be time-boxed and must deliver some executable code.
- The future is very likely to look like the past.
- It's never too soon to begin work on the technical/software architecture.
- One person must have overall technical control.
- Very often, the high-level design isn't written down.

### **Biography**

John Daniels has applied object technology in a range of industrial and commercial applications since 1985. He prepared and presented what was probably the first

commercial training course on OO design in the UK in 1986 and has given lectures and tutorials at most of the major European object technology conferences. Since 1990 John has spent a large proportion of his time advising project managers on the effective use of object technology and in helping to rescue troubled projects. He is the author, with Steve Cook, of a book entitled "Designing Object Systems", published by Prentice-Hall in September 1994.

### **Charles Berman, Fidelity Investments**

*ceberman@fmrco.com*

There has been a strong motivation to adopt Object Oriented techniques in the Financial Services sector where the high rate of change, and constant need for customization plays to its strengths. However, it isn't enough to adopt Object Oriented technologies; other changes are required. The process of Object-Oriented software design and development has paralleled trends in other disciplines such as manufacturing, where individuals are empowered by small, interdisciplinary teams with high bandwidth, informal channels of communication, and spiral methodologies that incorporate step-wise refinement at each of the design, build and testing stages. As project managers we are responsible for fostering this environment as well as performing the more conventional tasks of defining milestones, evaluating results and adjusting the course.

### **Biography**

Charles Berman is Principal Technical Consultant for Fidelity Investments and has been involved in the development of object-oriented applications for Financial Services and Network Management since 1986. Operating as project manager or systems architect, he has directed full life cycle development efforts in both C++ and Smalltalk. At Fidelity Investments, where Mr. Berman is working on CORBA-based systems for international securities trading and portfolio management, his principal challenges are enterprise object modeling, fault resistance and distributed deployment.

## Jim Coplien, A T&T

*cope@research.att.com*

A manager can run an object-oriented project according to the same general project management rules as apply for any paradigm. If there are any differences for OO projects, they relate to doing risk management in a field of emerging tools and methods that lack track records available for other methods (e.g., as for database design). It does not require a management paradigm shift.

Why the popular belief that OO project management is so different? The Smalltalk legacy of OO has popularized iterative development, and iterative development has come to be equated with OO. Look, C programmers on fast machines have been doing iterative development for a long time; I've worked in an "interactive batch" FORTRAN shop where people could turn around changes almost as quickly as we can on a work station. That was 22 years ago. Such techniques were discouraged in the formative years of software methods, when industrial processes were the only available model for software. Now iteration is fashionable, and perhaps more common than it was 15 years ago. But the same faster machines that help people iterate in C++ and Smalltalk are helping the C people do the same thing, so iteration is pretty ubiquitous—even in projects whose formal process descriptions don't admit it.

Prior OO management experience is nonetheless a plus: familiarity with the tools and methods can help a project from making naive choices.

### Biography

Jim is a member of the Software Production Research Department in AT&T Bell Laboratories. He is currently studying organization communication patterns to help guide process evolution. Among the long-term goals of this research are to develop a generative pattern language. Such a language would help new organizations build development processes from first principles, tuned to needs and constraints of individual businesses, technologies, and organizations. Other research interests include software architecture patterns and large-scale C++

development. He is author of "C++ Programming Styles and Idioms," and co-editor (with Doug Schmidt) of "Pattern Languages of Program Design." He writes a patterns column for the C++ Report.

## Doug Johnson, Rothwell Interntional

*doug@rw.com*

At the core, management of a object oriented project is no different than management of any other project. You still need to know what you are doing (analysis), how you are going to do it (design), you need to do it (code), and you need to make sure you have done it (test). What OO technology does change is the cost equations. Coding is cheaper, so you tend to do more prototypes; good design is rewarded, so you tend to spend more time in design; complexity is better managed, so you can do more with less. A successful project manager must understand both project management and the cost issues. It is clearly a help, but not essential that they have OO development experience.

One of the more interesting differences in OO projects derives from the fact that system structure always reflects organizational structure. Since good OO systems are structured differently than conventional systems, you need a different organizational structure. Many conventional systems are structured around major function. This is a disaster for an OO project where function is spread across many classes. A OO project should be organized with teams owning groups of related classes (subsystems) and individuals within teams owning classes. Interfaces among subsystems should be carefully defined as part of the design process and closely maintained and tested during the coding process.

### Biography

Douglas Johnson is a Senior Consultant with RothWell International specializing in OO project management and systems architecture. With 15 years experience in OO technology and large project management, he has held project manager or lead architect positions for projects as large as 100 developers and \$44,000,000 budgets.