brian henderson-sellers
swinburne university of technology

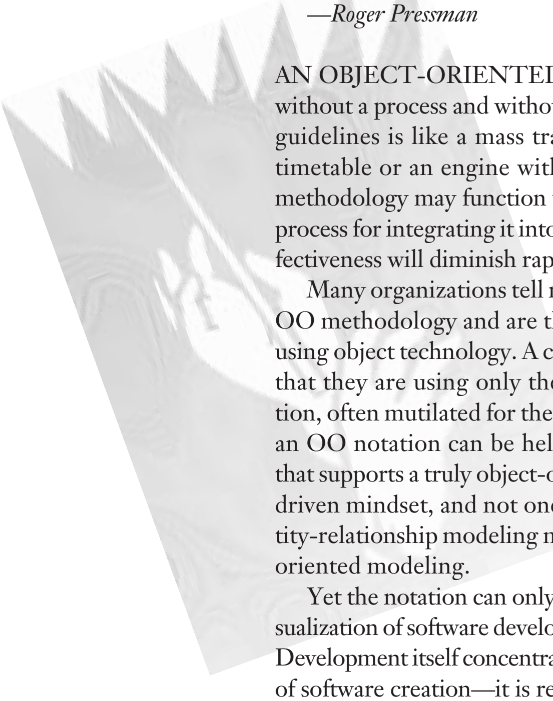
OO Project Management: The Need for Process

How to get people and technology to work together.

Editor:
Roger Pressman
R.S. Pressman & Associates, Inc.
620 E. Slope Dr.
Orange, CT 06477
pressman@rspa.com
http://www.rspa.com

It seems that the software engineering community follows the same pattern whenever a new paradigm is introduced: Excitement, followed by hype, followed by debate, followed by a harsh recognition that the new technology is not the proverbial silver bullet. If the technology has merit, we then pursue more practical and important concerns. Object technologies have followed this pattern. Since most software projects fail because of poor project management, Brian Henderson-Sellers' advice should provide practical and important guidance for OO project managers.

—Roger Pressman



AN OBJECT-ORIENTED METHODOLOGY without a process and without project management guidelines is like a mass transit system without a timetable or an engine without oil. Although the methodology may function well initially, without a process for integrating it into the organization its effectiveness will diminish rapidly.

Many organizations tell me they have chosen an OO methodology and are thus fully committed to using object technology. A closer inspection reveals that they are using only the methodology's notation, often mutilated for their own purposes. Using an OO notation can be helpful if you choose one that supports a truly object-oriented, responsibility-driven mindset, and not one that is really only entity-relationship modeling masquerading as object-oriented modeling.

Yet the notation can only offer support in the visualization of software development process artifacts. Development itself concentrates on the technical side of software creation—it is really the design process, the recipe for building a system from its ingredients. Together, the design process and the notation form the methodology. On the other hand, in the business world such a process-incorporating OO methodology is only a small part of building a system. The methodology says nothing about business issues, people, organizations, or technology. At this higher level, then, we have another process—the *software engineering process* or *life cycle process*—consisting of human resources, technology, and methodology.

Whichever OO methodology you choose, I contend that only pretending to use a development approach that has no process element in it will lead to business failure. For large-scale industry projects, any methodology you adopt must contain a serious description of the design process along with advice on how to best use the tools at your disposal to leverage this process. You can tailor the process to your particular organization, in terms of the normal project management variables of time, money, people, scope, tools and techniques, and quality. You can then implement the process in a new environment of iterative development, incremental delivery, and rapid prototyping. This software engineering process lets you retain your responsibilities to organize, direct, staff, plan, and control, which leads to the specific project management tasks of scheduling, estimating, monitoring, contingency planning, and risk management.

Using a development approach that has no process element will lead to failure.

DUAL FOCUS. Project management in object technology has two main focal points:

- ♦ development and implementation of a resource allocation plan, and
- ♦ development of a software development plan and strategy.

Planning resource allocation. The development and implementation of a resource allocation plan emphasizes project management, especially planning, estimation, and control. Typical supportive techniques include cost estimation, Gantt charts, CPM charts, PERT charts, priority setting, project planning, risk analysis, contingency planning, traceability, and workflow analysis.

Project management focuses on planning, controlling, and measuring the allocation of time, money, people, scope, and tools and techniques—

0740-7459/97/$10.00 © 1997 IEEE

all in the context of achieving quality goals. The project schedule identifies temporal constraints, possibly temporal orderings (particularly of deliverables), units of work (tasks), and milestones by which to measure progress. Furthermore, a good project manager realizes

- the knowns and the unknowns of the project environment,
- what must be done to eliminate the uncertainties and unknowns, and
- what must be done to ensure milestones are technically and politically feasible.

A good project manager must replan as the project unfolds and make midcourse corrections as necessary. A winning OO project management strategy builds management trust and retains customer involvement by delivering results incrementally, if possible on a quarterly basis. Timeboxes can be useful in this context—here delivery is made incrementally to the customer based on the end of a planned duration, not on completed functionality.

Resource allocation planning can be broken down into 13 subtasks:

- Choose your hardware.
- Choose your project team.
- Choose your toolset.
- Decompose large-scale development programs into projects.
- Develop an education and training plan.
- Develop an iteration plan.
- Develop a timebox schedule.
- Identify project roles and responsibilities.
- Manage your subsystems.
- Set up a metrics collection program.
- Specify individual goals.
- Specify quality goals.
- Use dependencies in the System Object Model to generate a first-cut project plan (Gantt chart).

These subtasks require the selection of animate and inanimate objects; planning for people, process, and training; goal setting for quality and personal achievement; and evaluative procedures using metrics.

Developing plans and strategies. This second high-level task concentrates on developing context plans and strategies using a strong, fairly traditional IS planning focus. Typical supportive techniques include database authorization, password protection, physical security, process modeling, and wrappers (a technique for turning non-OO code into large objects).

Large-scale software planning must depend upon business decision making. I recommend a third-generation OO methodology that I and my colleagues developed for OO process support (*The OPEN Process Specification*, Addison-Wesley, 1997; http://www.csse.swin.edu.au/cotar/OPEN/).

OPEN, which stands for object-oriented process, environment, and notation, focuses on those organizational decisions that define the organization's culture and its software development paradigm.

Here are the subtasks, listed alphabetically, for forming a software development plan and strategy:

- Develop a hardware capacity plan (disk storage, RAM, and so on).
- Develop a contingency plan.
- Develop a security plan.
- Establish a change management strategy.
- Establish a data migration strategy.
- Tailor the life-cycle process.

Perhaps the most important of these subtasks is the last one. As exemplified in OPEN, this subtask creates a toolbox of goals, tasks, techniques, life-cycle processes, metrics, and deliverables. These elements are all specially tailored to your organization, but remain within the standardized OPEN methodological framework. Thus, project managers can assemble their own life cycle process to accommodate the unique characteristics of their project and organization.

Planning at this level must take into account the existing culture. For example, Larry Constantine and Lucy Lockwood ("Fitting Practices to the People," *American Programmer*, Vol. 7, No. 12, 1994, pp. 21-27) describe four stereotypical organizational cultures: closed, random, open, and synchronous.

- In a closed culture, management is traditional, stable, and hierarchical. Project management follows an authoritative style.
- At the other extreme is a random culture where independence, creativity, and self-expression reign. A manager in such a culture must be a facilitator, not an authoritarian.
- In an open organizational culture, cooperation, openness, and communication are keywords.
- Finally, in a synchronous culture, we see quiet efficiency, smoothness, and harmonious functioning, managed either in terms of a shared tradition or by a charismatic leader fostering alignment with a common vision and shared sense of mission.

Constantine and Lockwood argue that unless an organizational self-assessment is undertaken, project management—and also change management—will likely fail, since the style of the manager and the culture shared by the individuals could readily conflict, with possibly disastrous results. You must also develop several fairly traditional plans: capacity, contingency, security, change management, and data migration, including the incorporation of legacy systems.

In migrating to object technology, project managers and senior managers must emphasize the selection of an appropriate process. They should *not* opt for object technology by merely observing and then selecting the most popular notation.

Regarding your OO process, make sure the project management support you will need is available in the methodology: in its documentation, in its CASE tool support, and in the skills of the trainers and mentors you might employ to assist you in installing it. Don't shortchange yourself when implementing the most important element of object technology—a process-focused OO methodology. ◆

A good project manager must replan as the project unfolds.

Brian Henderson-Sellers is director of the Centre for Object Technology Applications and Research (COTAR/Victoria) and professor of computer science (object technology) at Swinburne University of Technology in Melbourne. He can be reached at brian@csse.swin.edu.au.