Experiences Applying Methods Supported By CASE Tools

Kathleen Culver-Lozo

Vicki Glezman

AT&T software projects have experimented with different software engineering methods supported by computer-aided software engineering (CASE) tools. This paper examines the challenges of adopting methods and tools by summarizing the experiences of 15 projects. By examining the experiences of these projects, conclusions about how to apply methods and tools can be made.

Introduction

Software development methods are technical "recipes" for building software. Methods provide special notations, tasks, and quality assurance criteria for software analysis, design, development, testing, and maintenance. Because the notations tend to be detailed and graphical, developers usually turn to software tools to help make applying the methods practical. These tools, called computer-aided software engineering (CASE) tools, provide graphical editors to represent software in the appropriate notations and provide consistency checkers to automatically evaluate the quality assurance criteria.

Although adopting methods can be challenging, they offer projects important benefits. First, the software development interval—the time needed to develop the product—can be reduced. Most methods today emphasize closely examining customer needs and the architecture chosen to implement them. These examinations reduce the changes that must be made during the implementation, testing, and maintenance phases of software development, where making a change adds time to the development cycle. Because methods also prescribe how intentions should be expressed and interpreted, the interval is reduced because communicating analysis, design, and implementation intentions is simplified.

Second, methods can also reduce software development costs. If project members can make fewer changes, rework is reduced. Also, methods can promote the reuse not only of software, but also of documentation, designs, analyses, and test scripts. If a project can reuse work instead of developing it, then effort is saved.

Finally, methods can improve the quality of the software produced. The emphasis on examining customer needs increases the chances of a successful product launch. Also, the quality assurance criteria associated with the work products of a method increase the chances that the product will function as designed.

For example, a transmission project developing 300,000 lines of code reported a 50 percent productivity improvement. The same project also reported a quality improvement when requests for changes to the delivered product were reduced 35 percent. The reduction was attributed to the emphasis on accurately gathering and communicating customer needs.

However, the results varied for other projects, often because development organizations stopped applying the methods before completing their first release. This paper examines the challenges of adopting methods by studying the experiences of projects that have used them.

Experiences With Methods and CASE Tools

In recent years, AT&T software projects have experimented with different methods and CASE tools. By examining the experiences of these projects, conclusions about how to successfully apply methods can be made. Table I lists 15 projects that have tried different methods and tools. The table shows a project identifier and the application domain

Table I. Projects Adopting Methods

Project ID	Application domain	Methods Used	Tools Used	Size (S,M,L)	Status (Q,C,I)
A	operations system	-Yourdon's modern structured analysis -Hatley/Pirbhai's structured design	-StP	S	C
B	network management system	-Yourdon's modern structured analysis	-StP	M	Q
C	network management system	-Hatley/Pirbhai's structured analysis	-StP	S	Q
D	operations system	-Hatley/Pirbhai's structured analysis	-StP	S	Q
E	operations system	-Hatley/Pirbhai's structured analysis	-StP	M	Q
F	work management system	-Hatley/Pirbhai's structured design	-StP	S	C
G	operations system	-DeMarco's structured analysis	-Excelerator®	M	Q
H	operations system	-Shlaer/Mellor object oriented analysis and recursive design	-Teamwork	S	I
I	operations system	-Shlaer/Mellor object oriented analysis and recursive design	-Teamwork	L	I
J	operations system	-Ward/Mellor structured analysis	-Teamwork	S	C
K	work management systems	-Shlaer/Mellor object oriented analysis -CRC cards and WirfsBrock object oriented design	-StP -CRC cards tool	M	I
L	work management system	-Hatley/Pirbhai's structured analysis -Page-Jones structured design	-StP	S	C
M	network management system	-Hatley/Pirbhai's structured analysis -Yourdon/Constantine structured design	-StP	L	C
N	network management system	-Hatley/Pirbhai's structured analysis	-StP	L	Q
O	network management system	-Shlaer/Mellor object oriented analysis	-Teamwork	L	I

Excelerator is a registered trademark of Index Technology Corporation

(i.e., the type of system) of the project. The projects studied are applications for managing:

- A data or telecommunications network (i.e., network management system)
- Equipment in the network (i.e., operations systems)
- Work assignments of telecommunications employees (i.e., work management systems).

The projects used methods for analysis, design, or both, and could use structured methods, object-oriented (OO) methods, or a combination of the two. Structured and OO methods are further distinguished by the specific practices recommended by a methods expert. Thus, structured analysis as prescribed by Yourdon[1] is slightly different from structured analysis in Hatley and Pirbhai's prescription.[2]

The projects also had a variety of CASE tools from which to select. The most popular tools used were Software through Pictures® (StP) by Interactive Development Environments, and Teamwork® by Cadre.

Finally, Table I indicates whether the project was small (up to 5 people), medium (6-20 people), or large (more than 20 people), and whether it completed at least one release using the methods (C), whether it quit (or scaled back on) using the methods (Q), or whether it is in progress and has not yet completed one release using the methods (I). To benefit from a method, the project has to use it throughout the development process. Many projects do not realize significant benefits until the product's second release. If the project abandons the method before the first release is delivered, much of the investment in methods is lost.

The projects picked for study were part of a database of over 100 project teams that had worked with an internal AT&T organization for information and advice about methods and CASE tools. The study took in projects that required at least two members to apply the methods and that stayed in contact with the internal AT&T organization. Thus, the sample is not random, but provides a cross-section of projects adopting methods at AT&T during the past two years.

Factors Influencing Adopting Methods

Table I shows mixed results for the fifteen projects. While nine projects used—or are using—the methods as planned, six either stopped using them or scaled back on their use. The table alone does not indicate a correlation between project outcomes (i.e., whether they were completed or abandoned) and their sizes, applications domains, methods, or tools used. Thus, other factors clearly affect the extent of a project's success with methods. By interviewing project members, information was collected about the factors that affected the adoption of methods. These factors and strategies for controlling them are described below with anecdotes from the projects.

Appropriate Methods and Tools. Table I outlines the available methods. Project members must make the following decisions:

- Whether to use a method for analysis, design, or both
- Whether to use structured or OO approaches
- Which specific structured or OO approach to use
- Which tool to use.

Carefully choosing a method is important because specific methods provide a different kinds of support for various application domains. For example, Hatley and Pirbhai's structured analysis method strongly supports real-time applications, but is less useful in database-intensive applications. If a method is chosen that does not fit the application being developed, important information will not be collected and disseminated, and project members will spend their time on less important issues.

For example, a large operations system project arranged education to introduce their members to structured methods. They arranged for the vendor to present the database-intensive version when their dynamic applications required the real-time version. To avoid such a problem, a method must be matched to the needs of the project. While an ideal method may not be available, a satisfactory method usually is. The "Getting Started With Methods" section of this paper elaborates how projects can approach selecting a method.

Project members must also be careful in selecting a CASE tool, because different tools are designed for different methods. For example, StP is tuned for structured analysis and design, but currently provides only minimal support for OO analysis and design. The tool's ease-of-use also must be considered because most methods cannot be applied without a CASE tool. If a tool is difficult to use, team acceptance of the method may be jeopardized.

For example, a medium-sized operations system project adopted DeMarco's structured analysis and purchased a PC-based CASE tool. But the interface to the tool was so complex that a clerk had to be hired to input the diagrams. In addition, the PCs used on the project were not networked together, so project members were often working with separate and inconsistent versions of the diagrams. These experiences frustrated project members and impaired their productivity. A more careful evaluation of the tool's ease-of-use could have prevented the project from investing in an inadequate CASE tool.

Adequate Training. After a method and tool have been selected, an investment in training is required. Training time for the various methods ranges between two and four weeks. The staff needs one week to read the appropriate text or texts, and between one to three weeks of class time. Tool training can usually be completed in three days. The cost of classes varies and can be high if organizations must send staff members to off-site classes.

Because of lost staff time and class costs, many organizations are tempted to truncate or avoid the education process. But by skipping training, project members may need more time to learn the method on their own, and then may apply it incorrectly.

For example, the managers for a network management system had their original staff members trained on structured methods. But as new staff were added, they were given time to read the texts, but they did not attend classes. The managers found that the new staff spent several extra weeks trying to apply the methods that the initial staff did not need. The new staff also debated about how to apply the methods, and in the end applied them incorrectly. Had management trained the new staff, weeks of confusion and frustration could have been avoided.

Enthusiastic Management. While some managers may be excited about a new method, others may not. Some may feel they already know how to manage a project, and that a method cannot help them further. Others may resent a change in the scope and responsibilities of themselves or their staffs. If managers are not enthusiastic about a new method, they will not schedule the time needed to learn about and apply it. They will continue to judge their employees' performance on criteria relevant to the old way of doing business instead of on criteria based on the new method.

For example. some members of a small operations system were eager to use structured analysis to help understand their customers' needs. Management reluctantly agreed, but did not allow the entire affected staff to take the necessary training classes. The managers also laid out the project milestones based on old analysis procedures, instead of in terms of structured analysis. And they continued to evaluate employee's performance on criteria relevant to the old way of analyzing a system. The result was that the staff was confused about how to use structured analysis and fit it into the project milestones. Advocates of the methods were also subjected to more scrutiny from managers, and spent much time trying to "sell" their managers on using them. Eventually, structured analysis was abandoned because it was blamed for schedule slips.

This experience suggests that a new method should not be adopted until it has the full support of all managers involved. The managers must commit to developing project plans consistent with the methods, changing performance evaluation criteria, and securing training for all staff involved. If these steps are not taken, management is not committed enough to the methods to make their use successful.

Enthusiastic Staff. Not only may managers be reluctant to adopt a new method, but staff members may share their reluctance. One or two staff members may feel they know how to do their job, or may not like the new responsibilities implied by a new method. When this happens, the staff members may choose not to cooperate with the adoption of a method. They may debate with managers and other staff on the importance and benefits of the method, and may therefore apply it poorly. To prevent this, managers must clearly state their expectations for adopting the method. If the new performance criteria and expectations are made clear to staff members, they are more likely to comply.

But if most staff members are reluctant to adopt a new method, managers have a more difficult challenge. The staff's reaction may be the result of a strong bias against a method, or of previous failures using methods. To overcome this bias, an advocate from the staff should be encouraged to apply parts of the method to his or her job. A staff advocate is often more effective than a manager in changing staff members' attitudes.

For example, the managers of a work management system were interested in using structured design. Their staff, however, felt that structured design had little value, and constrained their creativity. The managers found someone on the staff who was curious about structured design and was willing to try it. The managers gave that person the chance to learn the method and expanded his responsibilities on the project so he could apply it. When the staff member began expressing satisfaction with the method, other project members became more open to trying structured design.

Staff reluctance can also be the result of poor management-staff relations. Staff may feel that managers are forcing a method on them without understanding the real problems they face when developing software. Both managers and staff must improve relations before a new method is adopted.

Realistic Expectations. Projects often adopt a new method, expecting that with some training and a CASE tool, software development time or cost will be instantly cut in half. These expectations are cultivated by the media, CASE tool vendors, and seminar providers. However, experience has shown that the benefits of methods do not accrue until later in the project life cycle. Managers often become disappointed when the benefits they expected are not realized in the short term. As time pressures mount and budgets tighten, these managers often choose not to continue the necessary investment to make the methods and tools successful.

For example, a project chartered to develop a network management system decided to experiment with structured analysis on one of its teams. The team invested in CASE hardware, software, and training. However, it soon became apparent that although the quality of the requirements produced by the team would improve significantly, the productivity of the team would not greatly change. Because management expected large productivity gains from the team, structured analysis was abandoned. This example illustrates that although the entire project would benefit in the long-term from the improvements in requirements provided by structured analysis, short-term considerations took priority. Consequently, the investment in the CASE tools and training was lost because management expectations could not be met.

Managing expectations is difficult because methods offer much promise to improve productivity, costs, and quality. However, before making an investment in methods, the objectives and success criteria for the methods should be defined. Ideally, these objectives would be reviewed with someone knowledgeable about the methods, familiar with the project domain, and sympathetic to the project's interests. Relying on CASE vendors to confirm the feasibility of these objectives is risky because vendors' first interest is in selling tools and training.

Emphasis on Methods and Process. When a method is introduced, its most visible feature usually is the CASE tool that supports it. The tool provides a window on the notations used by the method and the rules for consistency. Its visibility is often the reason many project members spend more energy selecting and learning the tool than the method. They expect that once the tool is mastered, they will enjoy the benefits promised by the CASE vendor. Unfortunately, mastering a CASE tool is only a small part of the transition a project needs to make. Project members must accept new ways to think about their work. New deliverables must be produced. Hand-offs and other responsibilities from one group to another must change. These aspects of adopting methods are not addressed by CASE tools. However, many project members still focus on managing the transition to the new tool alone, not the transition to the new software development process the methods imply. When this happens, the methods are poorly applied and do not provide the expected benefits. Project members often then conclude that CASE is mostly "hype."

For example, a software project for testing an operations systems was interested in CASE tools, and invested heavily in hardware and tool training. These investments came after careful study of CASE tools and hardware platforms. But similar attention was not paid to selecting a method, changing the project plan, and redefining people's responsibilities. Today, this project has a lab filled with equipment, but the project members are using the old methods.

Emphasizing methods and process instead of tools is challenging because methods are abstract while tools are concrete. Also, some organizations that value technical accomplishments hold the bias that CASE tools are technical and methods and process are not. If project managers want to increase the chances for a successful method introduction, they should try to manage the process, not the tools. The tools would be managed through the process. Selecting a staff member to be a "process engineer" whose responsibility is to focus on the process is one strategy for achieving this process focus.

Getting Started With Methods

Though adopting methods supported by CASE tools can be challenging, the risks can be managed. This section describes two critical actions successful projects have taken to ensure successful adoption of a new method. These actions involve creating two plans that seek to manage the selection of a method and to incorporate the method into the project.

Methods Selection Plan. Because a project has many methods and tools from which to choose, a strategy for selecting a method and tool must be decided upon and clearly stated. A basic decision—one that must be made early—is whether to use structured or object-oriented methods. Both approaches offer gains in productivity and quality through revolutionary development process change.

The approaches use similar model construction techniques and notations, and they cover the same portion of the lifecycle: analysis, architecture, design, and coding. However, they have a different emphasis: functional versus object.

The object-oriented approach may have certain advantages, but it also has a larger learning cost, and is supported by less mature tools. Because the two approaches are similar, learning one will aid in learning the other, and projects may want to use structured methods as a stepping stone to object-oriented methods.

Given the importance and the challenge of selecting an appropriate method, a methods selection plan should be prepared that defines:

- The staff members responsible for selecting the method and CASE tool. These people must be empowered to make the selection to prevent other managers or staff from rejecting it. They must also be prepared to explain the expected benefits.
- The time frame for the decision.
- The candidate methods to be considered. These methods should be limited to those with a suitable reference text, have detailed training classes, have a supporting CASE tool, and be endorsed by the best current practices of the organization.

Methods-Specific Project Plan. Once a decision has been made about methods and tools, a project plan is usually prepared before work begins. The project plan typically includes sections that define the tasks, project milestones, roles and responsibilities of team members, and standards to be used. The project plan for an organization adopting a new method should reflect its effect on the deliverables, milestones, and other relevant factors. In particular, a Methods-Specific Project Plan would include:

- A specification of the methods chosen.
- The tool chosen to support the method.
- The parts of the application implemented with the methods. To minimize risk, a project can apply the methods to one or two non-critical features.
- The staff members assigned to these features, with their roles and responsibilities defined. These individuals should be enthusiastic about applying the methods.
- An outline of the specific training required, the training schedule, and the names of the staff and management involved. No one should be allowed to start work until their training is complete.
- The method-specific deliverables. These will vary according to the method chosen. For example, Shlaer and Mellor's OO analysis[3-4] includes an information model for each domain, a state model for each object and relationship, and a process model for each state. This combination of models comprises the analysis. The project plan might include a deliverable matrix whose rows list each domain, and whose columns represent each model. The intersection of each row and column represents a deliverable who due date appears at the intersection. A similar matrix could be used for deliverable inspections. A rule of thumb for the development lifecycle when using specification methods supported by CASE is 40 percent for analysis and architecture, 35 percent for design, 10 percent for coding, and 15 percent for testing.
- The method-specific conventions and standards for the project. This will minimize confusion and eliminate unnecessary discussion. Once adopted, compliance with the standards or conventions must be enforced. An example of a convention in structured methods would be "context diagrams contain only one process."

Preparation of a Methods-Specific Project Plan by management demonstrates its commitment to methods use. The project plan is the key to process definition in an organization. Focusing on it insures proper emphasis on methods and training, which increases the probability of success.

Summary

Methods supported by CASE tools can reduce development intervals and product costs, and can improve the quality of software. But before these benefits are realized, the methods must be systematically applied to the project. Those projects that prematurely discontinued methods use failed to achieve the expected benefits. Root cause analysis of these projects revealed several factors affecting the adoption of methods and tools. These factors could be managed through careful method selection, enthusiastic management and staff, reasonable expectations, and a method-specific project plan.

References

1. Edward Yourdon, *Modern Structured Analysis*, Yourdon Press, Englewood Cliffs, New Jersey, 1989.
2. Derek J. Hatley and Imtiaz A. Pirbhai, *Strategies for Real-Time System Specification*, Dorset House, New York, 1988.
3. Sally Shlaer and Stephen J. Mellor, *Object-Oriented Systems Analysis: Modeling the World in Data*, Yourdon Press, Englewood Cliffs, New Jersey, 1988.
4. Sally Shlaer and Stephen J. Mellor, *Object Lifecycles: Modeling the World in States*, Yourdon Press, Englewood Cliffs, New Jersey, 1992.

__Kathleen E. Culver-Lozo__ is a member of technical staff in the Software Process Architecture and Engineering Department at AT&T Bell Laboratories' Red Hill facility in Middletown, New Jersey. She is responsible for defining software development process for various AT&T projects. She joined AT&T in 1987, and has a B.A. in economics from the University of Michigan, Ann Arbor; an M.B.A. in information systems from Boston

College, Massachusetts; and an M.S. in psychology, also from the University of Michigan.

Vicki Glezman *is a member of technical staff in the Software Process Architecture and Engineering Department at AT&T Bell Laboratories' Red Hill facility, Middletown, New Jersey, where her current activities include CASE startup assistance and Best Current Practices presentations, specifically for the Specification Techniques Supported By CASE Tools BCP. She joined AT&T in 1986 with a B.S.E.E. from Pratt Institute, Brooklyn, New York, and an M.S.E.E. from Syracuse University, New York.*

(Manuscript received July 17, 1992)