



Previous screen

# 4-01-25 Implementation Planning for Software Methods and CASE

Rebecca Winant

## Payoff

This article explains the elements that make up a plan to successfully implement formal software methods and CASE in any organization. Techniques to make the entire planning process easier are included.

## Introduction

Planning is critical to the smooth launch of and effective progress during a software development project. Planning sets the project's direction, expectations, and objectives. It makes changes and improvements easier to implement and to track. However, it is also time-consuming and requires much effort on the part of the developers and users. For this reason, adequate and comprehensive planning is often not done. Simplifying the planning process can help organizations get off to a more confident start using a standardized software methodology and Computer-Aided Software Engineering tools.

This article presents a basic framework for establishing an implementation plan. The framework discussed in this article does not provide an outline of a specific plan but examines what must happen so that an effective plan can be developed and how to support implementation of the plan. This approach has been adapted by many different organizations, most of which were just beginning to use CASE and formal methods or were reintroducing CASE tools and development methods after a previous false start. However, any organization should be able to adopt it.

## Gaining a Perspective on Change

Productivity improvements using advanced technology usually imply a change in the way people are currently doing their jobs. Although organizations anticipate the positive results, unexpected side-effects can be negative, possibly resulting in the failure of a formal methods and Computer-Aided Software Engineering implementation program.

Organizations that intend to start using a standardized development methodology and purchase CASE tools should first ask questions about their current systems development process: What systems are currently available? What systems are desirable or necessary and why? Each of these questions could and should have several answers, and it is important to obtain all of them. This means asking questions of both management and staff, and ideally, asking questions of people at the corporate level as well as at the departmental and project level.

Prescribed software methods and CASE should be consistent with overall business plans. The information collected from the questions, when sorted out, can provide a valuable perspective on desirable and likely changes. This is essential information for planning and monitoring those changes.

The approach presented in this article references three implementations of software methods and CASE by a national broadcasting company, a software group within a hardware vendor, and a telecommunications company. These organizations' experiences provide insight into how these plans can be developed and used. In each organization, the focus was to provide appropriate support for the individual work environments on the



Previous screen

basis of needs, technology, and culture. Aspects of those experiences are related to give a sense of the variation in organizations and their approaches and plans.

Before collecting the essential information to begin the implementation plan, the organization must establish goals for the project and provide mechanisms for effective communication during the project.

## Setting Clear Project Goals

Implementation plans are starting points for change. The negative effects of change can be avoided when clear goals are defined. Therefore, the first task an organization must do is state clearly the desired outcomes. What is the purpose of having a standardized methodology? On what basis is a particular Computer-Aided Software Engineering tool or method selected? What is expected of them? What evidence is necessary to verify that the desired benefits have been obtained? One company stated that the mere acquisition of a Computer-Aided Software Engineering tool allowed it to check off a key customer requirement. However, that customer listed a means, not an end. It is essential to determine the objective the customer hoped to achieve using the CASE tool.

CASE tools do not guarantee anything; direction must be set by the organization using the tools and techniques. The main functions of implementation planning are to set realistic expectations and to build a plan that is realistic and avoids disruption while achieving specific goals.

Knowing the status of the project relative to the goal at any point in time is important. Otherwise, a lot of activity could be interpreted as wasting time. Therefore, goals should include sufficient detail to know exactly when they have been reached. It might be seeing a numerical measure of dollars, market share, or time; observing diagrams and documents; experiencing a behavioral shift in department dynamics or in project team attitude; or some combination of these. This is the baseline against which progress will be assessed. Setting clear goals is the first step in the whole process.

## Maintaining Effective Communication

A success factor for any methods and Computer-Aided Software Engineering implementation program is effective communication. Recognizing this makes all the other steps easier, and probably faster and less expensive.

The process of implementing methods and CASE can be divided into three basic phases: assessment, planning, and monitoring. Information gathering and information distribution must be ongoing to support the activities within these phases. The phases and activities are shown in [Exhibit 1](#). The purpose for these ongoing tasks is to ensure effective communication. Change is facilitated by involving people who will be affected by it. This includes both management and staff in the departments that are involved with systems development. The kind and type of information people are given about the change should be tailored to their role in the organization.

## Establishing an Implementation Plan

In organizations in which there is weak or nonexistent communication about the goals and relevant facts of a methods and CASE program, failure of that program is common. For example, for one organization, a big kick-off meeting for the whole program provided



Previous screen

evidence that a failure was likely to come. The presentation of CASE products, software methodology concepts, and new development procedures was met with apathy because the attendees had no idea what was going on. Management and staff were at the meeting because they were told to be there, but they had no basic information about how the new software and new procedures related to company goals, project goals, or their jobs specifically. Senior management was not given an overview of the program, the tools, or the expected benefits. The program was designed with the assumption that management had already bought in because the bills were paid and the program was included in the schedule. This example of noncommunication is extreme.

Conversely, when information flows, the results are usually positive—even if the goals are not fully met. In another organization introducing CASE, management made several adjustments to its original plan based on feedback from two pilot projects. A purchased life cycle methodology was tailored to the corporate culture and the in-house tools in use, and a version was then written to focus people on the important objectives for a project as well as the key tasks and deliverables. Although the specific method initially brought into this organization was replaced with another two years later, the organization had fundamentally changed the way it developed systems and the way it interacted with its customers. The expense of training, consulting, and everybody's time on the initial method was not wasted because the organization improved its development process and became more sophisticated about identifying appropriate methods for its development needs. Information exchange keeps effective communication going, makes change understandable, and motivates people to be interested in becoming part of the process.

## **Information Gathering Techniques**

### **Assessment**

The first phase of information gathering is usually called needs assessment, but it is more appropriately called assessment, because it should encompass more than simply determining needs. It should also focus on what the organization currently has and what it wants to have, as well as what it needs and why. These questions should focus on five areas in which methods and Computer-Aided Software Engineering impact is felt. The five areas each play a role in how an implementation plan is shaped.

### **Technical Skills.**

This is an assessment of people's technical skills in terms of formal or informal methods, type of development, and development experience. This assessment provides information for selecting a method and for determining education, consulting, or hiring requirements as part of the planning process.

### **Methodology.**

This includes information about the formal or informal methods, tool use (CASE or ad hoc), and life cycle methodology. This information can be used in selecting a method. A totally unstructured shop may not want to leap to object-oriented methods immediately, because it requires a certain level of discipline and sophistication to implement. Wants and needs relative to the phase or phases of the life cycle should be considered. An organization



Previous screen

may not want to introduce change on everything from analysis through testing unless development is currently performed systematically and formally.

### **Hardware and Software Environment.**

This should list current, desired, and needed technology. Along with the previous categories, this information should be considered in selecting a Computer-Aided Software Engineering tool. For example, CASE code generator should probably run in the development environment, but many organizations have analysis tools running on PCs while they develop for mainframe or workstations.

### **Organizational Relationships.**

This refers to the ease of communication between the development organization and its customers and any other department that provides specifications or will use the delivered system. Another factor to consider is the ease of access to these people. If there is a need to include them, it might narrow the choice of solutions for the implementation plan.

### **Management Support.**

This should identify whether management support for development methods and CASE exists and, if so, where it exists and how it exists. For example, the budget commitment to implement methods and CASE must come from management. The CASE tools and any required training, consulting, or hiring constitute another expenditure. Other expenditures may include the purchase of related software and new hardware. If management support is not extensive, the scope of an effort to implement CASE tools and a software methodology is likely to be small and runs the risk of not being cost-effective or successful.

### **Using an Assessment Matrix**

An assessment matrix, as shown in [Exhibit 2](#), can be used to gather information on all of these five areas. When filling in a matrix, the interrelatedness of the cells should be considered, and conflicts across the rows and columns should be identified. The implications of change should be addressed, and desired results should be weighted by importance or criticality. An overly ambitious vision of the future that implies a radical change is not realistic and is generally never realized. One company created a plan without filtering or setting priorities for the matrix information, which included an extreme change from a partly structured Common Business Oriented Language and mainframe shop to an object-oriented C++ and UNIX workstation environment. Although such an environment could reasonably be expected to produce savings through the reuse of code and more effective computing on distributed workstations, for a first project that involved such a major change without careful analysis of priorities and interrelationships, the reality was that this did not even come close to happening. Determining priorities contributes to a reasonable plan that can succeed in attaining the stated goals.

### **A Sample Assessment Matrix**



Previous screen

	Have ----	Want ----	Need ----
Technical Skills	what: Good S/W code skills, Fair testing/not enough time Experienced with systems	what: Better analysis and documentation why: Testing and user communication	what: Legacy for audits, distributed design skills why: Flexible designs
Methodology	what: None	what: A set of techniques and guidelines on project use why: Support skills and communication	what: Common and simple language for documents why: Standard use
Hardware Software Environment	what: IBM mainframe/COBOL	what: LAN and PCs with client/server architecture why: Flexibility	what: Distributed design why: Flexible, efficient
Organizational Relationships	what: Little access to system users and business division due to geography	what: Better means of communication, audit information why: Lower S/W cost	what: Meet audit requirements, improve relationship with business unit why: Job requirement
Management Support	what: Adequate resources who: Department mgmt	what: Resources for training and tools who: Department mgmt	what: Same as want who:

In the sample matrix in Exhibit 2, the Want column has been used to reflect goals that are 12 months away and the Need column reflects the business reason or motivation for that goal. Another piece of information in this type of matrix might include penalties for the needs if certain changes do not take place. In other situations, the Want entries might reflect longer-term goals of three to five years, while the Need entries are used to reflect desired change for the short term. In this type of matrix, the Have, Need, and Want columns represent a linear direction to move toward. In the example in Exhibit 2, the Want and Need entries are different aspects of desired changes to the Have column. The content and use of the matrix depends on the responsibilities and interests of the person using it. There are no hard and fast rules other than to portray the organization's particular situation as accurately and objectively as possible.

### Information Gathering for the Assessment Matrix

Information should be collected from various sources to get the most accurate perspective. With most organizations, the easiest way to accomplish this is by using a survey—either using a written questionnaire or conducting thorough interviews, or some combination of the two, depending on the culture and time available. In general, management answers are easier to obtain through interviews. A separate set of questions for managers and staff can be used to address each group's specific issues. An identical set



Previous screen

of questions for everyone provides a pulse of the organization and gathers everyone's opinions. It is not necessary to cater to individual responses. However, weighing all of the attitudes and opinions is part of creating a plan that will work.

It is human nature to slant self-assessment questions toward the positive, and many questions geared to what is wanted or needed will possibly be either inflated or understated. The volume of feedback should balance those out. An opportunity to create improvement can be missed if the assessment matrix does not include real problems or needs in the current development process. Management often views the company as having a higher level of software development effort than is actually present. Having an independent agent perform the survey is one way to combat this blindness or subjectivity.

It is equally important to get information from external sources, such as journals, conferences, industry associations, or books about formal software development methods and Computer-Aided Software Engineering. These can help when comparing various methods and CASE tools and learning about their track record. Companies in the same industry may be run differently. Companies in different markets may have similar management or system development practices and their experiences can provide guidance in how to set priorities and avoid potentially damaging efforts.

### **Survey Styles.**

Multiple choice questions are easiest and provide a more consistent measure. To do these well, the multiple choice must offer a range and should map to the assessment matrix. The range allows extremes to surface. For example, the following survey question might be used for assessing current organizational relationships:

Q: The typical team you work with on a project:

- Communicates very well.
- Communicates adequately.
- Lacks effective communication.

If all of the responses to this question weigh in at 75% for answer 3, 15% for 2 and 10% for 1, it is reasonable to assume that communication at the project level is a problem. Coupled with questions that test access to information, staff-management communication, and interdepartmental communication, it should either point to specific isolated problems or broader problems. This wide representation makes it easier to develop an appropriate solution. If interdepartmental communication is an issue, the plan should call for a third-party facilitator to be included as part of a development team. Some project-specific problems can also be solved with team-assigned roles and responsibilities. Many communication problems are alleviated with better ways of distributing information.

An example of a survey question that focuses on project management capabilities is:

Q: The objectives for the projects you work on are:

- Clear to everyone.
- Known only by project leaders.
- Not clear to anyone.





Previous screen

If responses to this question indicate that people feel as though they are mostly in the dark, tactics for projects can be developed to make that situation visible so it can be resolved. For example, a team can distribute its best guess at the project objectives to customers for their corrections and input. Project conventions might specify a communal data base or reference of open project issues, assumptions, and unallocated facts. One item in the set might be the project objectives.

Because these surveys ask for current status, questions that ask for a response on a scale should be included. They can be tallied as a score. For example:

Q: What rating would you give to the current proficiency with structured methods?

(nonexistent) 1 2 3 4 5 (excellent)
-------------------------------------

In addition, specific recommendations for what is working well and what could be improved and why should be solicited. The option to provide positive feedback is not always obvious, but it should be included to gain a comprehensive view of the organization. Therefore, the option should be made specifically clear.

## Organizing the Information

A matrix organizes information and is used to determine if organizational needs are consistent with goals and wants. [Exhibit 3](#) illustrates such a matrix and shows where the conflicts, unknowns, and synergy exist in the lists of wants and needs. Areas that are unknown require further investigation or redefinition. At this point, priorities should be set for wants and needs as well.

## A Matrix Correlating Assessment Information

Next, solutions should be identified that address the high-priority items and the wants and needs that have been selected. Together with an assessment of current skills and practices, this helps in deciding how much of each element should be introduced in an initial program. Should new analysis and design methods be tackled together? Does a metrics program have to be part of an initial CASE implementation? What support is needed? A solutions matrix assesses support for organizational needs. [Exhibit 4](#) is a support matrix that shows the CASE and software methods support for all the stated needs. For both of these types of matrices, the elements suggested for the implementation plan can be added in a version with potential solutions to determine the best fit.

## Solutions Matrix

### Refining Expectations

Often the goals set initially are overly optimistic. After a check on the status quo and a look at the reports on CASE and methods use(both pros and cons) the goals should be reviewed. Are the established goals attainable and within control? A common problem when CASE or methods do not deliver everything as promised is that developers and their



Previous screen

customers become discouraged and the apparent or suspected culprit—methods or Computer-Aided Software Engineering or both—is blamed and eliminated. Often a lot of progress has been made and reasonable models are built, but if the original goals were too ambitious and not tempered by assessing the realities of need and constraints, positive change might be overlooked.

## **Elements of an Implementation Plan**

### **Vehicles for Information Gathering.**

Surveys are a useful feedback mechanism. Surveys should be simple and usually no more than two pages long, so participants will not feel burdened. These surveys should be blind and individual responses should be considered confidential.

### **Vehicles of Information Distribution.**

Ongoing communication about the pilot projects, methods guidelines, tips and models, and the Computer-Aided Software Engineering tool is important to keep motivation high and to keep the learning curve for everyone on an upward path. Information can be distributed through bulletin board notices (if such openness does not pose a security concern), memos, newsletters, monthly focus group meetings, and internal conferences or forums. The people who lead these sessions should be able to contribute success stories, lessons learned, and recommended guidelines.

Disseminating some survey information is another possibility. For some questions and for certain situations, this can be a positive step. For example, in one organization, a partially written and partially oral survey was conducted to see if individual opinions were in sync with the overall tone. The survey helped determine that, in fact, only a minority totally disapproved of the project. The majority requested coaching to help but by and large retained a positive attitude toward the program. Publishing the overall statistics and feedback helped quell the discontent.

### **Tools.**

This element of the plan should include what software and hardware has been selected for the pilot projects and when it will be acquired. Any support that comes with the new tools should also be identified.

### **Education.**

Methods and product training are the obvious components of this element of the plan, but sessions to inform testing, management, and requirements personnel about their expected roles and responsibilities and to familiarize them with new documents and the new process should also be included. Education should support system life cycle activities, enhance current skills, and provide a deeper understanding of method and tool features. An effective and comprehensive education program can prevent problems, such as reinventing the methods, counterproductive management actions, and misdirected technical directions.

### **Consulting.**

Consulting can be viewed as another mechanism for training. Occasional spot checks by an experienced CASE and methods user can help keep a project on track. The role of this type of consultant is that of a coach. The intent is not for the consultant to build models or do the work for team members, but to critique the team's work, providing reassurance when they





Previous screen

have an effective model and suggestions for change or correction when they are diverting from a reasonable model.

### **Schedule.**

As part of the plan, a schedule should be established. It should allocate budget and resources for all the previous elements of the plan.

## **Establishing Policy to Support the Program**

An organization that is serious about attaining the benefits of using formal software methods and CASE tools should institute and enforce a policy that reinforces their use. This means that those who resist using the methods are penalized and those who comply are rewarded. To do otherwise presents an inconsistent message from management. Either the methods are important or they are not. Whatever the stated goal is, the policy should reinforce it.

## **Monitoring the Implementation**

Monitoring the implemented vehicles of information exchange listed in the plan allows the planners to fine-tune their program to the ongoing needs of the organization. Even negative feedback about the introduction of Computer-Aided Software Engineering and methods can be used in a positive fashion.

Even the most successful organizations usually start out with the appearance of interoffice problems: The introduction of methods may have been forced on them from the finance and auditing department, and the managers may be afraid that the change is going to make their job of delivering a system on schedule more difficult. In addition, methods mean building models, which may have to be built by hand if a CASE tool is not being purchased.

The underlying technique used to address these types of problems is information distribution. The planners should determine what specifically is objectionable about a suggested program and ask for alternative suggestions. Sometimes minor adjustments make a big difference to the participants. Some of the most successful elements in such programs are suggested by people involved in the program.

During and at the end of the project, credit should be given for steps completed toward the goal. If steps appear not to be leading to the desired outcome, adjustments should be made to the program. Certainly, at the end of a project (i.e., after the system is installed), the overall effect of the first use of CASE and formal software methods should be analyzed.

## **Three Programs and Their Effectiveness**

The following synopses of the software methods for Computer-Aided Software Engineering implementation planning at three companies illustrate, respectively, a success with strong management support, a success because of an active grass roots movement, and a failure due to unclear management direction.

### **The Broadcasting Company—Strong Management Support**

A broadcasting company was asked to change its development practices by company auditors, and senior systems management executed that order. One specific goal was to



Previous screen

produce documents that would permit inspection, so that conformance to requirements could be checked. A life cycle methodology was brought in to govern the steps and a structured method was to be used in each phase to produce documents.

The program included methods training and consulting, no CASE tools, and no changes to development hardware or software. Initially, the consulting was used to monitor and support the modeling activities in two pilot projects and eventually up to eight other projects. Both written surveys and interviews were conducted to assess ongoing needs. Feedback was given to the participants in memos from management and through tune-up sessions tailored to individual participant needs. This method eliminated potentially large obstacles in implementing a successful program.

The keys to making this project successful were working separately with management in their planning and estimating activities for projects and letting them know the status of their staff relative to progress with the methods. Once-a-week sessions were held with voluntary participation to discuss specific problems. Because no technical staff was allowed to attend management sessions and no management was allowed in technical sessions, the discussions were more open and therefore more useful.

### **The Software Group—Active Grass Roots Efforts**

The software group's objective was to improve its software quality. This was largely a well-organized grass roots movement, because there was not much in the way of budgeting support except the purchase of a CASE tool and not more than 10 days of consulting during the course of a year. Management did permit the group to spend sufficient time on their methods and CASE activities.

The group defined its quality goals in terms of reliability and collected statistics for those measures. The group used an informal, internal training program for methods and Computer-Aided Software Engineering and for technical and managerial staff. In addition, budget permission was obtained for some basic consulting to fill in a few gaps in experience with architecture.

Two projects were involved. Participation was on a volunteer basis, and the two teams were highly motivated. The consulting or coaching effort ultimately needed to provide little direction, because these groups often presented their own analysis. Consulting was often simply confirming their observations and decisions.

### **The Telecommunications Company—Unclear Management Direction**

The telecommunications company set up a program involving the entire management team. The program consisted of CASE and methods training and consulting. One missing element was the involvement of the technical staff in either the planning or selection process. Although they were somewhat motivated to try new tools and techniques, the goal of the organization was never explained to them.

Some positive change was observed at an individual level, however. Three participants turned into the local experts, because they set personal goals in the absence of organizational direction. Some management feedback was negative and was typical of the overall communication style. Some members of the project management team still had not fully bought into the program's applicability to their specific work. Ultimately, the initial plan for a company program failed. The latest word was that the organization is starting to reintroduce methods again.



Previous screen

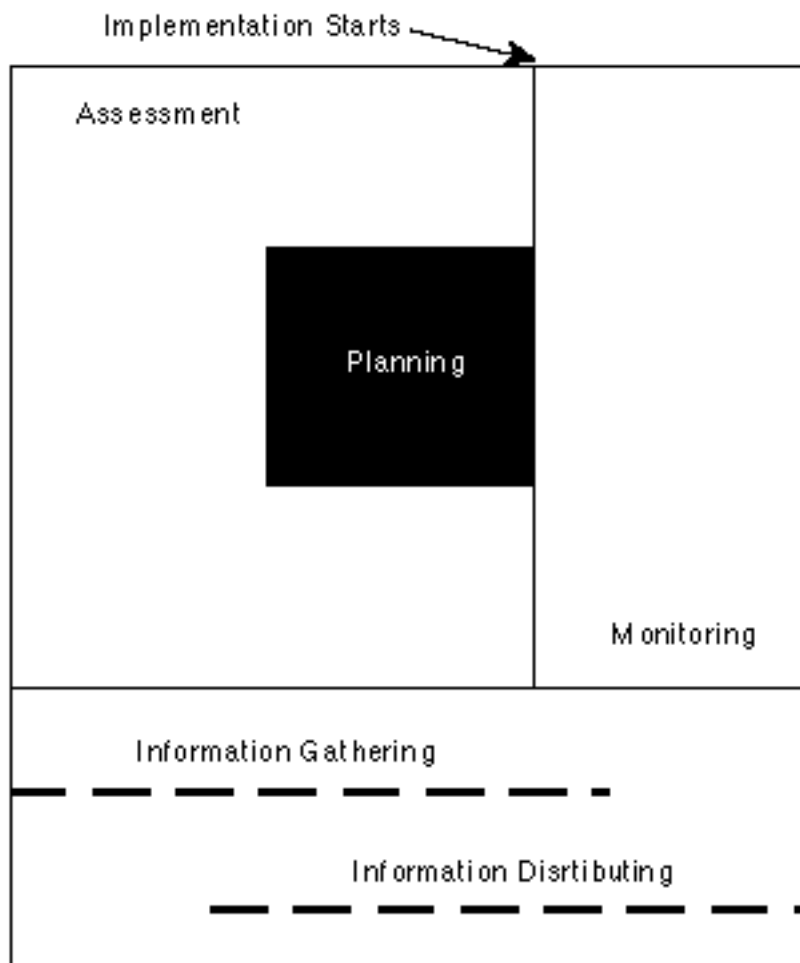
## **Conclusion**

This article outlines the elements that make up a plan to successfully implement software methods and Computer-Aided Software Engineering in any organization. An implementation plan can map a direction. It also is intended to promote some careful consideration of the direction an organization hopes to follow. Once an initial plan and program begin, the elements can be carried forward to continually support change. The plan also provides the confidence that the path the organization is taking is a reasonable and useful one.

## **Author Biographies**

Rebecca Winant

Rebecca Winant is the founder of Esprit Systems Consulting, Inc., in West Chester PA.



### Activities

- Define goals
- Organize assessment information
- Examine possible solutions
- Evaluate solutions fit with schedule, budget, and resources
- Build plan for selected solution
- Compare information about progress with plan
- Adjust plan and/or actions
- Solicit information through surveys, interviews, and brainstorming sessions
- Distribute information through education, coaching sessions, user groups and written communication

What we need     What we want	Standard development guidelines				
	Better communications with customer management				
	Better estimating guidelines				
	Methods/CASE experience				Documentation
Increased productivity	?	✓	?	X not immediate	?
Documentation	✓	✓	-	✓	✓
Better definition of customer needs	X/✓ either way	✓	?	✓	?
Faster development and change	X/✓ either way	?	-	?	X time?

How do the wants mesh with the needs?

—	neutral
?	don't know
✓	positive (compatible)
X	negative (conflict)

Solutions  What we need	Product X CASE tool					
	Tools for testing					
	Tools for metrics				Simulation or prototyping	
				Analysis method		
				Design method		
Skill in method and CASE	✓	✓	?	-	✓	✓
Better communication with customer mgmt	?/✓	-	?	✓	✓	-
Better estimating guidelines	-	?	✓	?/✓	?/✓	?/✓
Documentation	✓	-	-	X	✓	✓
Standard development guidelines	-	-	-	-	-	-

How do the wants mesh with the needs?

- |   |                       |
|---|-----------------------|
| - | neutral               |
| ? | don't know            |
| ✓ | positive (compatible) |
| X | negative (conflict)   |