



[Home](#) | [Products](#) | [Services & solutions](#) | [Support & downloads](#) | [My account](#)

developerWorks > Rational >

developerWorks



Software program management: Questions and answers from an enterprise workshop

Level: Introductory

[Michael Hanford](#)

Chief Methodologist, SUMMIT Ascendant Methodologies

15 Jun 2005

from The Rational Edge: Between the presentation of a group of practices and their successful application in real software program efforts, there is an understanding gap that must be bridged. Program and project managers have specific needs, constraints, and often ongoing work that must be accommodated when they adopt new practices. This article is based on Mike Hanford's discussions about program management with a group of European senior program and project managers. It also provides a brief introduction to the IBM Rational Program Management Method.

I recently spent a number of days conducting a workshop with a group comprising one enterprise's highly experienced and thoughtful European senior program and project managers as well as members of their supporting software process group. I had been invited to present key concepts of the IBM Rational Program Management Method, and to provide feedback on their approaches, practices, and experiences in mobilizing and managing large initiatives. Below is a summary of what we talked about and learned from one another.

Questions about large initiatives

Many questions that workshop participants raised dealt with issues and situations common to all large initiatives.

How can you tell whether an initiative is a program or a project?

This basic question was also the one participants asked most frequently. My answer? "It depends."

For some initiatives, it may be possible to determine from the outset that they should be structured as programs. For other efforts, the determination of what structure is needed will emerge gradually through investigation and analysis.

However, readers can rest assured that, in the early stages of defining what results you need, as well as boundaries, roles, potential size, and funding needs, the distinction between program and project has little bearing on the work required to get the answers.

Table 1 describes elements that can help you determine whether an initiative should be structured as a program.

Table 1: Elements to help distinguish between projects and programs



Contents:

[Questions about large initiatives](#)

[Questions about the IBM Rational Program Management Method](#)

[Conclusions to apply](#)

[About the author](#)

[Rate this article](#)

Subscriptions:

[dW newsletters](#)

[The Rational Edge](#)

<p>Separation of concerns</p>	<p>You may remember from system design that separation of concerns in an application contributes to reliability and ease of maintenance.</p> <p>In a program effort, individual projects are structured around different specific and well-bounded concerns. These concerns are then transformed, through work effort, into solution components (or needed results), which are then integrated together.</p> <p>Imagine a defense-industry effort to create a new airplane. Designing the airframe would require a distinctly different effort from designing the engines.</p> <p>Each undertaking would require different engineering capabilities and different amounts of time, and the target metrics and parameters would be very different. Therefore, in terms of focus and skills, it would make little sense to assign these two efforts to a single team, under a single manager.</p>
<p>Owning or impacted business segments</p>	<p>For many projects, the solution is required by a single business segment that defines requirements, reviews preliminary designs, and specifies a set of business drivers for component testing / implementation. Typically, the executive responsible for that business segment is the final arbiter of a needed result or solution component.</p> <p>In terms of "ownership" and potential impact, a program typically spans multiple business segments that have competing agendas and different drivers for needed results. These agendas may even be mutually exclusive: Gain for one represents loss for another.</p> <p>Program governance structures have built-in mechanisms to review, refine, and arbitrate competing agendas and needed results, and to reach a consensus consistent with needs defined in the enterprise business strategy.</p>
<p>Finances</p>	<p>As with all business efforts, financial concerns are a major driver for software project and program initiatives. The answers to such questions as:</p> <ul style="list-style-type: none"> • Where will the money come from? • Who will pay what portion of the cost? • Who has the final say about expenditures? <p>will impact the initiative's direction and potential for success.</p> <p>Typically, projects have a single funding source and a single role for decision-making about expenditures. From a process perspective, this makes it simple to get financial decisions made. However, larger efforts may have multiple funding sources, each wanting a say about where and how money is spent. The competition that results can harm the initiative's chances for success.</p> <p>Because of their significant cost, programs are typically an enterprise concern rather than the concern of a single business unit. Funding often comes from multiple business segments and is overseen by the program governance structure.</p>

Even after considering the elements in Table 1, the answer to the original question may not be obvious. During mobilization, when you are getting organized to start work, an analysis of the following three areas may help to resolve the question.

- **Organizational structure.** If the effort has been organized as a single project, are there multiple, competing interests for the project manager's attention? Is the project manager spending most of his / her time arbitrating among these competing interests? Is it difficult for the team to decide on a focus?

Affirmative answers to these questions may indicate a need to structure the initiative as a program.

- **Boundaries.** Does the initiative seem to have multiple "masters"? Is there a need to conduct meeting after meeting to resolve issues, allocate funds and resources, and debate content and boundary issues?

Affirmative answers to these questions may indicate the need for a permanent steering committee with well-defined authority; this is a program mechanism.

- **Resources.** Will the initiative require a large staff -- more than, say, forty people -- over its lifecycle? Will it require people with a broad mix of capabilities and skills rather than just a few analysts and programmers? Will individual staff members need to join (or leave) the initiative in almost every month of its lifecycle?

Again, affirmative answers to these questions may indicate that the effort should be structured as a program rather than as a project.

What are the program manager's role and authority vis-a-vis the executive sponsor's role and authority?

During our discussion of these questions, experienced program managers in the group offered multiple models for their role as well as the overall program structure. Here is a role definition for a program manager:

The program manager is responsible for the planning and execution of program activities, and for the delivery of work products and results in conformance with the goals and outcomes defined in the overall business strategy and program justification.

The program manager provides leadership as well as day-to-day management and control of all program resources and funding.

The program manager works closely with the executive sponsor(s), who defines needed outcomes, sets boundaries or constraints, and exercises oversight for the program. The two roles collaborate to ensure clarity and understanding in these areas.

As this definition makes clear, the program manager directs the program work after interpreting the executive sponsor(s)'s definitions of outcomes and

boundaries. It is the program manager's job to divide the overall effort into achievable interim work products with well-defined results, shape an effective organizational structure, and make decisions all along the way.

In our initial discussion, some of the European managers described their program manager role as someone whose primary responsibility is to *coordinate* and *communicate* rather than actually lead or direct the program. Coordinating and communicating are part of the program manager role, however, it entails far more.

To ensure consistency (and finality) in decisions and direction, as well as rapid and clear decision-making, *one person* must ultimately be in charge of the day-to-day effort. The alternative is to engage in lengthy negotiations, devote significant time to assembling and explaining what is needed, go through multiple decision-making iterations and discussions on the same issue, and suffer the effects of a lack of coherent vision across multiple (and sometimes integrated) decision points. This one person is the program manager.

Of course, the program manager's decision-making authority can (and should) be subject to *appropriate* constraints. These should be jointly set by the program manager and executive sponsor(s).

If program staffing includes a program manager and several project managers, what is the role and authority of each?

In presenting the respective organizational structures of their current programs and large projects, workshop participants described a variety of models for the project manager role. We examined this role's responsibilities along three organizational dimensions:

- **Downward.** How does the role relate to the staff of an individual project?
- **Upward.** How does the role relate to those who provide direction and leadership?
- **Sideways.** How does the role relate to other project managers both within and outside the program?

Along the *downward* dimension, the project manager's role is much the same as it would be within a standalone project. Within the boundaries of a component project, the project manager:

- Plans, organizes, directs, and controls the work effort for the individual component project.
- Manages for on-time delivery of specific work products and components.
- Manages work within the project plan framework.
- Manages technical staff.

However, being part of a larger program *does* create differences for project managers along the *upward* dimension. Individual project managers must work within a larger framework, or program plan, which defines dependencies and interactions among the various individual projects.

In addition, they must accept direction from the program manager and escalate issues that cross individual project boundaries up to that manager. In other words, for the duration of the program, the primary reporting relationship for individual project managers is to the program manager.

Individual project managers also have an *indirect* reporting relationship to the executive sponsor(s) because the program manager they report to, in turn reports to the executive sponsor(s). Should the need arise for changes in direction or other adjustments that impact individual projects, the executive sponsor(s), steering committee(s), and program manager would make the decisions and then communicate them to the project manager(s).

From a *sideways* perspective, all of the individual project managers are professional colleagues and peers, working within the context of the program. This group should be able to decide easily on minor adjustments and changes, and -- if necessary -- communicate them to the program manager.

For more important decisions, the program manager (or one or more individual project managers) typically schedules a group discussion with relevant project managers (including the program manager) to seek consensus about a course of action. In some circumstances, the program manager may elect (or be required) to escalate important issues or concerns to the executive sponsor(s) for action or decision-making.

Proposed changes that involve projects or organizations *outside* the program are a bit different. In such cases, the program manager and (ultimately) the executive sponsor(s) must represent the program and make decisions on its behalf. Project managers should escalate significant issues to these people but can settle minor items through interaction with other project managers. At the outset of program mobilization, the program manager should assemble all the individual project managers and work out a simple set of guidelines for interactions outside program boundaries.

What is a program management office, and what functions does it perform?

This question sums up close to a dozen questions that workshop participants asked about different aspects of the program management office (PMO). For many, the whole concept of a PMO was a new one that they wanted to understand. For others the concept was familiar, but their organization's business

executives and managers had approached it with resistance and skepticism.

According to the IBM Rational Program Management Method, a PMO is:

...a function within the program that provides management, administrative, and process support, together with "expert" resources to the program. In effect, the program management office is the "staff" function for the program. The PMO will collect and manage information and work products; develop status and tracking data; manage the budget, facilities, and infrastructure; and provide "expert" resources, such as architects, modelers, and testers, to all of the projects within the program.

A major role for the PMO is to continuously assess whether the ongoing work effort is on track and the interim deliverables and results are consistent with the overall program goals. The PMO also validates that the program results are contributing to the success of some component of the enterprise business strategy.

The PMO is also charged with achieving economies of scale within a large and complex effort. Suppose that each of five program projects requires an architect's services. Upon closer examination, it becomes clear that one project requires an architect half time, another requires one two-thirds time, and so on. So instead of supplying each project with one full-time-equivalent (FTE) architect, the PMO can identify and retain one or two architects and allocate portions of their time among all the projects, thereby achieving significant resource savings. Such economies of scale can extend to many areas: identifying / acquiring resources, managing the technical infrastructure, data collection and status reporting, for example. Rather than having each component project retain and use different individuals to perform these functions, the PMO can centralize the required tasks and assign resources to the projects only as needed.

If business executives and sponsors don't understand the need for a PMO -- and view it as overhead -- how can we convince them of its value?

In many enterprises, justifying a request means quantifying the costs savings, return-on-investment (ROI), and speed and / or productivity gains you will realize by instituting your request. Unfortunately, organizations rarely give as much credence to justifications, such as risk reduction, quality improvement, or improvements in process and execution.

Enterprises charter and approve large programs when they are pressed to make significant changes, achieve specific business results, and / or to react to competitive threats in the marketplace. The importance that senior business executives place on these goals is typically reflected in the program's funding level. If the program budget is \$200 million in US dollars, for example, we know there is a serious need, and the executives expect serious results. In such instances, doesn't it make sense to do everything that is reasonably possible to reduce the program's risk of failure and improve its chances for success? No one would answer this question with a *no*.

However, risk reduction is not free. Improving the likelihood for success is not free. In nineteenth-century America, many saloons had a free lunch counter stocked with sliced meats, breads, hard-boiled eggs, and other good things to eat. Of course, saloon proprietors simply incorporated the expense of this "free" lunch into the price of the beverages they sold. That's what led people to coin the expression, "There ain't no such thing as a free lunch"! The same is true for the important governance and monitoring functions of a PMO, which minimize risk and maximize the programs chances for success. For programs involving major resource investments, we can -- and should -- follow the dictates of ordinary business common sense and institute a well-organized, well-funded PMO to manage services and staffing levels and ongoing assessment and monitoring functions.

To the business executives and sponsors who complain that a PMO represents an extravagance or unnecessary overhead, we should ask, "How lucky do you feel today? Lucky enough to gamble more than \$200 million"? They need to understand that a well-defined, effective PMO is a significant mechanism for reducing program risk and improving the likelihood of success. In a future article, we will explore the role of the PMO in greater detail.

Questions about the IBM Rational Program Management Method

During the course of our workshop discussions on how to structure a program, I presented the IBM Rational Program Management Method in outline form. This method deals with the early portion of the work lifecycle, which includes mobilization, start-up of the PMO, establishment of the technical and physical infrastructure, planning, and implementation of management and control practices (see Figure 1).

Program Management – Five (5) Modules +

Mobilization:
Defines the governance, organizations, boundaries, & financial management

Program Office:
Establishes the administration and support function

Infrastructure:
Provides the resources, facilities, tools & technical environment

Planning:
Defines the process and produces individual project plans and a Program plan

Controls:
Sets up management / control practices and reporting

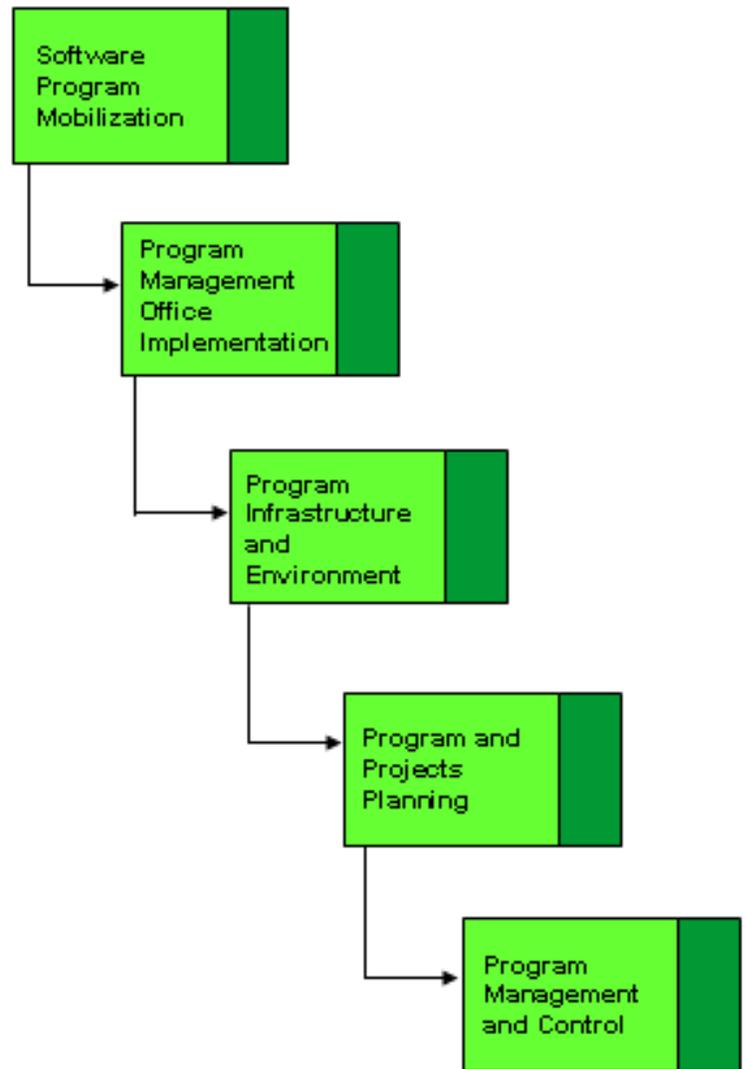


Figure 1: IBM Rational Program Management Method

During the early portion of a program lifecycle, success depends on the ability to gain traction -- that is, to move steadily forward in understanding the work and needed results, and in assembling an effective team structure and resources. The practices in the IBM Rational Program Management Method provide guidance in all of these areas.

What's the best way to introduce the IBM Rational Program Management Method in a large IT organization?

The members in the group representing the software methods and practices function asked this question because getting a large organization to take up and sustain new practices until they are well-integrated into the work ethic is indeed a difficult undertaking. Really, the implementation issue is only secondarily about using a new method. Primarily, it is about getting the organization to see value in the change so that it is willing to invest the time and resources necessary for the change.

The state of an organization has much to do with its receptivity to change. Organizations that are successful at what they do and have initiatives that measure up to success parameters are likely to see little value in introducing a new methodology and will find abstract arguments about the future and what "may" happen" unconvincing. However, as some of the large initiatives in this specific European enterprise had not been entirely successful, and as the organization is entering a period in which it will increase its number of large-scale initiatives, many of the project and program managers, as well as members of the methods and practices group that support them, did understand the need for change.

Another key to a successful introduction is identifying a specific business or IT segment that will be receptive to new practices. If an initiative is in an execution state, it is very difficult -- and high risk -- to add the burden of process change on top of other challenges. In contrast, a group that is considering a large initiative (or group of initiatives) and developing an approach to provide business justification for them might be a good starting place.

Any process change should begin with the question, "Historically, how well have our initiatives performed, and what can we do to address past problems or deficiencies?" You might compile a list of initiatives that are either under discussion or identified in the enterprise business strategy but not yet started. Then, as you can conduct informal discussions to identify key decision-makers, discuss past challenges, and obtain a full view of these near-term initiatives, your aim would be to identify a small target group that would benefit from the type and timing of your initiative. The group should include important decision-makers and conditions favorable for demonstrating the value of new practices, which would include risk reduction, better management oversight / insight, more effective resource use, and the ability to address specific challenges.

Do we have to use all of the method, or are some pieces more important than others?

I have a great deal of empathy for managers adopting a new method or process, especially one as robust as the IBM Rational Program Management Method. In summary, here is my reply to the group of managers when they posed the question above:

It is less about the method than about the situation in which you find yourselves, and the specific needs that you have -- as an organization -- to change and improve. However, whatever approach you take, it must fit and succeed, within the context of what is (and has been) successful within your own business and IT culture in general.

Published, or "commercial," methods offer generalized guidance for best practices that the authors have distilled and documented through years of working with a number of organizations and initiatives. These practices offer an organization high value, low risk, and an excellent chance of being successful. However, they do not *guarantee* success.

It is necessary for a thoughtful group of managers, such as the group with whom I met, to review and assess the new method's content and compare the guidance it offers against their own experience, their enterprise's imperatives, and the organization's values. This will help them understand the methodology and decide what portions of it to adopt, in what order, and how quickly. The IT organization's internal software process group should function as the agent for leading and coordinating this review and analysis.

During my work with the European enterprise, we had a number of discussions to identify the best starting point for a process introduction. The senior program and project managers, together with members of the internal software process group, rather easily reached consensus: Use the set of readiness checklists the method provides. The IBM Rational Program Management Method is divided into a number of activity groups, or related work activities (see Figure 1).

Each activity group has a corresponding checklist to help you validate readiness to start these activities (see Figure 2).

Checkpoints: Entrance Criteria for the Software Program Mobilization Discipline

Topics

- [Inputs](#)
- [Entrance Criteria and Review Areas in which special attention should be paid](#)
 - [Benefits - Identification & Management](#)
 - [Consensus](#)
 - [Finance](#)
 - [Organization - Projects](#)
 - [Planning - Goals & Milestones](#)
 - [Governance](#)
 - [Scope - Boundaries, Inclusions, Exclusions](#)
 - [Success](#)
- [Value Analysis Matrix for Entrance Criteria](#)
- [Score Ranges and Actions to Take](#)

Figure 2: IBM Rational Program Management Typical Readiness Checklist

The program manager and mobilization team members answer the checklist questions as "yes, completely present," "yes, some present," or "no, few or none present." A value analysis matrix is provided to score the answers and advise actions for certain score ranges.

Workshop attendees saw the checklists as a way to better understand the current situation, or state, before moving on to the next major phase of work. They also saw sharing the checklists with business executives and team members as a way to help them understand the components and / or pre-requisites they needed to begin or continue work on a major initiative.

In their particular enterprise, the checklists fit well with the prevailing business culture, general management practices, and state-of-the-art for program management. In other enterprises, other strategies for getting started might offer more value.

Conclusions to apply

What conclusions can we draw from this experience, and carry away and put into action? Below are brief summaries of these conclusions.

Programs and the enterprise

Enterprises and managers do not typically assemble a work group to determine that the organization should adopt program management practices. Rather, many organizations find that they need to undertake increasingly complex and important initiatives for which tried-and-true project management practices are inadequate -- and may even produce disastrous results. This leads them to investigate new practices and techniques specifically aimed at large, complex initiatives.

Best practices

In the search for better practices, many enterprises and managers assume that good standard methodologies either do not exist or that these methodologies will not accommodate their enterprise's unique needs, so they will have to construct their own.

In fact, a few excellent methodologies do exist, including the IBM Rational Program Management Method. Its practices are based on the experiences of real people doing real work on very large program efforts.

In general, commercial methodologies offer a set of solutions that are proven, integrated, and based on actual experience. Adopting such a methodology saves a tremendous amount of effort and offers a powerful mechanism for reducing risk when an organization undertakes unfamiliar types of work. However, no methodology will be a perfect fit for an organization right out of the box, so it is important to invest the time to assess the enterprise's current "state" and to adapt the methodology you choose to suit your particular needs.

Internal software process group

Another requirement for a methodology's success is a focus within the IT organization that proposes practices and helps users refine them to accommodate real experiences. This internal software process group must also obtain consensus for new practices it introduces. There is an old axiom that says: "What is owned by all is owned by no one." To be successful, process changes must have active (and visible) ownership by the entire organization and a focus upon capability improvement. This can come about only through the activism of the internal process group, which must form solid working relationships with the leaders and managers of major initiatives. Then, people within the organization can pull together to sample, use, and continually refine new practices to meet their changing needs.

For more information about the IBM Rational Program Management Method, please either visit <http://www-306.ibm.com/software/rational/howtobuy/index.html> or call 1-800-728-1212.

About the author



Michael F. Hanford is the chief methodologist for the IBM SUMMIT Ascendant methodologies and a member of the IBM Rational commercial methods content team. He has also worked as a methodology author, a manager for large consulting engagements, and a leader of enterprise process assessment and transformation efforts for IBM Rational and PriceWaterhouseCoopers Consulting (PWCC). Prior to joining PWCC, he was director of software engineering practices for Fidelity Investments Systems Company.