IBM®

Adoption through execution:Project-level mentoring to improve software capability

Level: Introductory

Kurt Bittner, Communities of Practice Architect, IBM
Saif Islam, Rational Services Manager, IBM

15 Jun 2004

From The Rational Edge: This article explains how IBM Rational services help organizations make lasting improvements in their process and software capabilities, using a proven mentoring approach.

▸ **Show developerWorks content related to my search: saif islam**

Increasingly, organizations are realizing that technical goals converge with business goals. To become more productive, they must first improve their software development capability. Greater productivity may mean producing solutions faster by reducing development cycles, or producing higher-quality solutions at lower cost. Either way, the underlying goal is to produce solutions faster, cheaper and better.

Better software capability can help you achieve this by improving predictability and repeatability to achieve greater project success. It also supports quality improvement by helping to reduce defects, change requests and re-work.

But improving software capability is a complex challenge. Software development is an inherently creative endeavor that seeks solutions to problems that have never been solved. Development projects can be unpredictable: Business needs evolve at a rapid pace, often in parallel with development efforts.

The trickiest challenge of all is improving a company's process for developing software -- a key component of software capability. Teams must continue working productively while process changes are being implemented, which is rather like trying to change an airplane engine while the craft is in flight.

That is why the IBM Rational solution for process improvement includes not only a proven, effective development process and automated tools, but also services that help organizations integrate process changes without disrupting their business. Process improvement requires organizational and behavioral changes -- changes in the way people communicate and collaborate as they do their work. Bringing about such changes requires a proven strategy, careful planning, flexibility and creativity in executing plans, and insight into issues surrounding organizational change. This article explains how IBM Rational services help organizations make lasting improvements in their process and software capabilities, using a proven mentoring approach.

Why is it so difficult to improve software results?

Most organizational process improvement initiatives fail, whether in the software development organization or other parts of the business, because real behavioral change is hard. As a result, many people believe that improving software capability by adopting a better process is an impossible goal. Years of experience have shown us that the goal is achievable, but the reasons for failure are many and complex. For example:

- Organizations often fail to take into account the cultural aspects of change.
- Organizations often try to make large changes all at once rather than making a few changes, measuring the results, and then undertaking further change based on those results.
- People often regard organizational change activities as separate from their day-to-day work (even though it is their day-to-day work that needs to change).

Software organizations are notoriously poor at investing in process and automation improvements. A common misconception is that *process is optional until proven otherwise*. Another is that *process automation tools are* really *optional until proven otherwise*. Correcting these misconceptions requires presenting concrete evidence that process and tool improvements lead to greater project success. We have found that engaging with clients and actually helping them to execute process changes is the only way for IBM Rational mentors to prove that both process and supporting automated tools are essential.

Change is disruptive

One reason organizations resist process change is that they fear it will slow them down and make them miss project deadlines. These fears are justified to some extent: Initially, all changes trigger a drop in productivity, as represented in Figure 1, area A. By the end of the T1 period the organization has bounced back to the productivity level before the change, but not until the T2 period does the organization "earn back" the productivity it lost because of the change. The end of period T2 represents the break-even point, after which the organization begins realizing benefits from the changes.

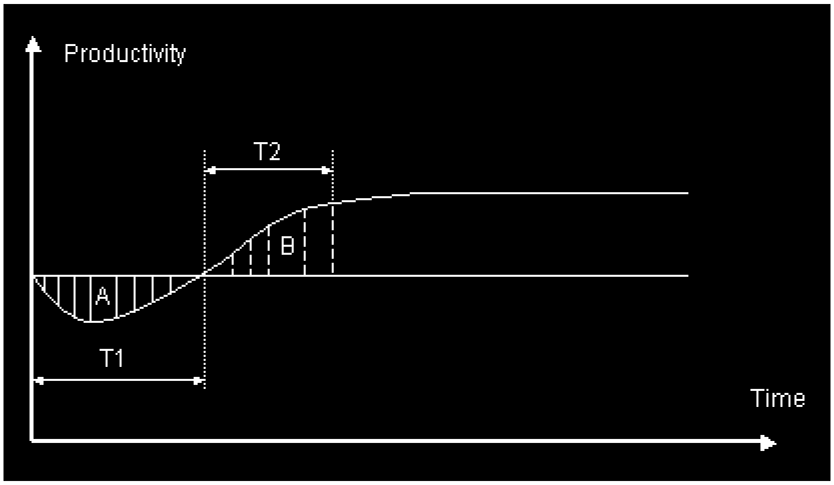

Figure 1: Typical effect of organizational change on productivity

Change requires experienced leadership

If change introduction is poorly managed, an organization can easily find itself worse off, or no better off, than it was before. If the organization takes too long reaching the end of the T1 period, people typically lose patience with the change and may give up on it, which results in a net loss of capability.

Inexperienced managers often believe that the first step in process adoption is to define/customize the process and configure the tools (e.g., customize templates, set up a process Web site, etc.). Once that is done, they believe, the adoption effort will become a simple matter of educating staff about the new process and instructing or encouraging them to use it. In our experience, this approach is ineffective. Although process definition is necessary, it is never sufficient to effect change.

As Figure 2 depicts, you need something to connect the process and tools to business results; in other words, *people have to apply the process before it can produce results*. Merely training people on how to use the process is not sufficient; that will give them knowledge but likely will not change their daily behavior. And failure to effect changes in daily behavior is the root cause of process improvement effort failures.

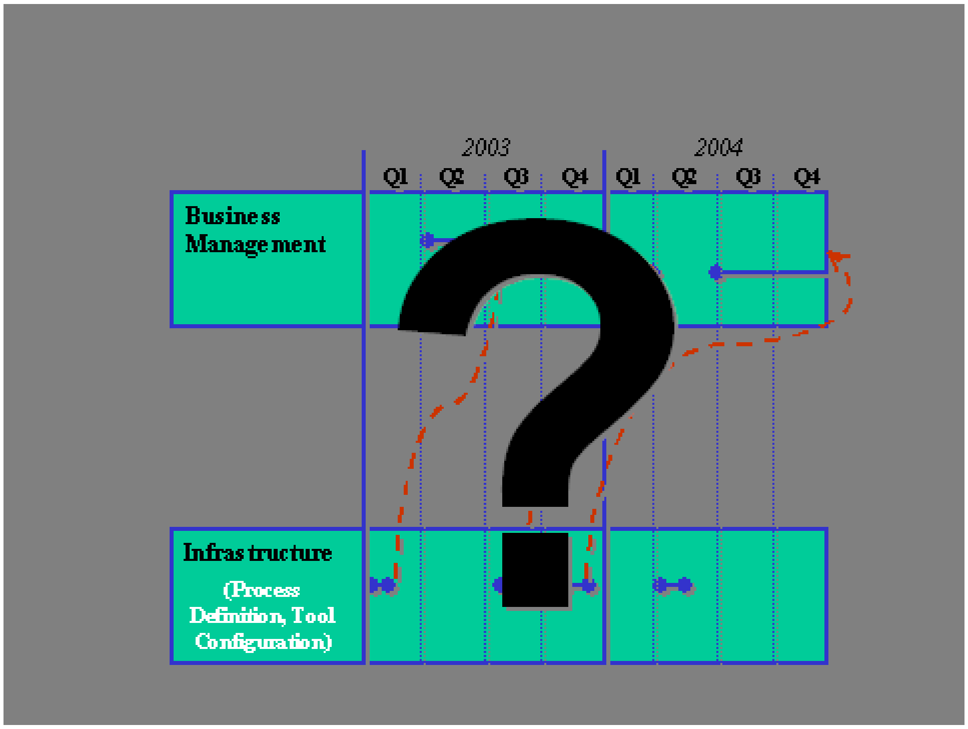

Figure 2: The missing link between process and tool definition and business results

When organizations make the mistake of focusing too heavily on process definition:

- The process improvement effort becomes isolated from the people who must implement the process. Instead, it is under tight control by a group of "specialists" who won't actually use the new process and most likely don't have enough experience to understand what problems must be addressed. For changes to be effective, the people who will need to change their behavior -- in this case, the actual teams developing software -- must be involved in the entire effort.

- It steers people away from best practices for organizational change. To make changes successfully, you must involve the people who will be asked to change their behavior. No one wants new rules forced upon them; involving people in the change will build support for it. This is even more important if people are currently experiencing the effects of a poor process.

- Managers are assuming that change is easy and takes little time. In reality, behavioral change happens slowly and gradually; it results from training that reinforces the change, providing mentoring, and supporting goals and rewards. The *real* process an organization follows is embedded in its daily habits. Publishing a process Web site will have little effect on these habits; managers must actively influence and lead change.

- Managers may overlook the important catalytic role that leaders and mentors play in making change successful. People will accept change if they feel confident of the outcome; good mentors not only transfer skills but also instill confidence.

Executing the process -- not defining it -- is the real challenge. Project teams must know where to start, how to proceed, and how to apply the process and tools effectively in real projects. We have observed many implementations of IBM® Rational Unified Process,® or RUP,® an extensive framework with a content-rich knowledgebase. This richness is RUP's great strength, but it is also a liability. Without experienced leaders who can tailor the process to solve a particular team's problems and address its objectives, the team might feel overwhelmed. It is important that people see measurable results early on in a process implementation, and only experienced managers can ensure that this will happen by choosing the right starting point.

Training alone is not enough

Inexperienced managers often rely on formal training classes to teach employees how to use a new process, tools, and technology. Training is always helpful but typically not sufficient. For most people, acquiring the ability to apply a new methodology, tool, or technology requires five stages of learning:

1. Basic knowledge
2. Understanding key concepts
3. Application of key concepts
4. Ability to make judgments based on concepts
5. Ability to adapt concepts to new situations (improvisation)

Most learners get only basic knowledge from a training class. They have to apply this knowledge to real problems in order to become truly capable. That is hard to do if you're fresh out of a basic training class unless you are working with a skilled mentor. Mentors can help learners quickly advance to the second and third learning stages and then support them in stages 4 and 5, as the learners gain more experience. It takes time and experience to hone the good judgment necessary for these stages, and there are no shortcuts. As Fred Brooks noted in *The Mythical Man Month*, "Good judgment comes from experience, most of which comes from bad judgment."

Adopting process through execution and mentoring

To achieve sustainable improvements, organizations must adopt process as they work on real projects; we refer to this as "adoption through execution." This approach has two major advantages:

- As we have noted, people will accept change if they can see that it leads to a positive outcome. These outcomes are apparent only within the context of an actual project. Managers might see more reliable schedules and improved delivery capability. Other project team members might feel they are acquiring valuable career experience or improving their quality of life. The sooner you can show people positive results, the sooner you will get strong support for the change. Until then, you will be running on "good will," which will disappear if people don't feel compensated for the disruption and extra effort the change entails. Unfortunately, organizations most in need of change typically have the lowest reserves of good will. In such cases, managers must demonstrate results throughout the change effort in order to maintain momentum.

- The only process that really matters is the one people actually use. We have noted that real change occurs only when people's daily habits change, and that requires doing and practicing over time. Change efforts based on "telling and teaching" or coercion almost never succeed. People change when they are motivated to apply the necessary effort.

The mentor's role

Successful organizational change typically requires an outside catalyst, and mentors can fill that role. To be effective, the mentors (typically software development organizations require more than one) must function as a high-performance team that collaborates to provide consistent messages and direction to the project team. Inconsistencies can erode learners' confidence in the guidance they receive and ultimately undermine the change effort.

To be effective, mentors must work with the organization for an extended period of time (four to six months if the desired change is at a program level, or two to three years for an enterprise-level adoption involving multiple project teams). They must cultivate trust by getting to know team members, the problem domain, project objectives, and the organization. Unless they are deeply engaged, it is difficult for mentors to add value.

A lead mentor should coordinate the team. This person must have depth of knowledge and experience across all nine RUP disciplines, as well as experience with organizational change. Other mentoring team members should have breadth and depth in at least four or five disciplines.

Mentors help people focus on *doing* without having to figuring out what to do. IBM Rational mentors also pass on engineering best practices. They build confidence in the outcome of a change because they have been successful on other projects. This confidence builds resilience within the team so that it can recover from the inevitable setbacks that will occur during the change effort. Of course, the mentors' ultimate goal is to build enough expertise and confidence within the client organization and project team for them to support the new way of working without external help.

For IBM Rational projects, a by-product of the mentoring effort is that the project team progressively adopts RUP techniques, as needed, to meet real objectives. In other words, they configure the process in an iterative way, *on demand*. The result is an organization-specific process based on RUP, and a proven tool usage model that fits the organization's needs and environment. Mentors help the project team select appropriate process elements based on the team's strengths and weaknesses. As Figure 3 shows, this is the best way to achieve successful business and technical results.

In the next section, we will present basic guidelines for a successful process change effort.

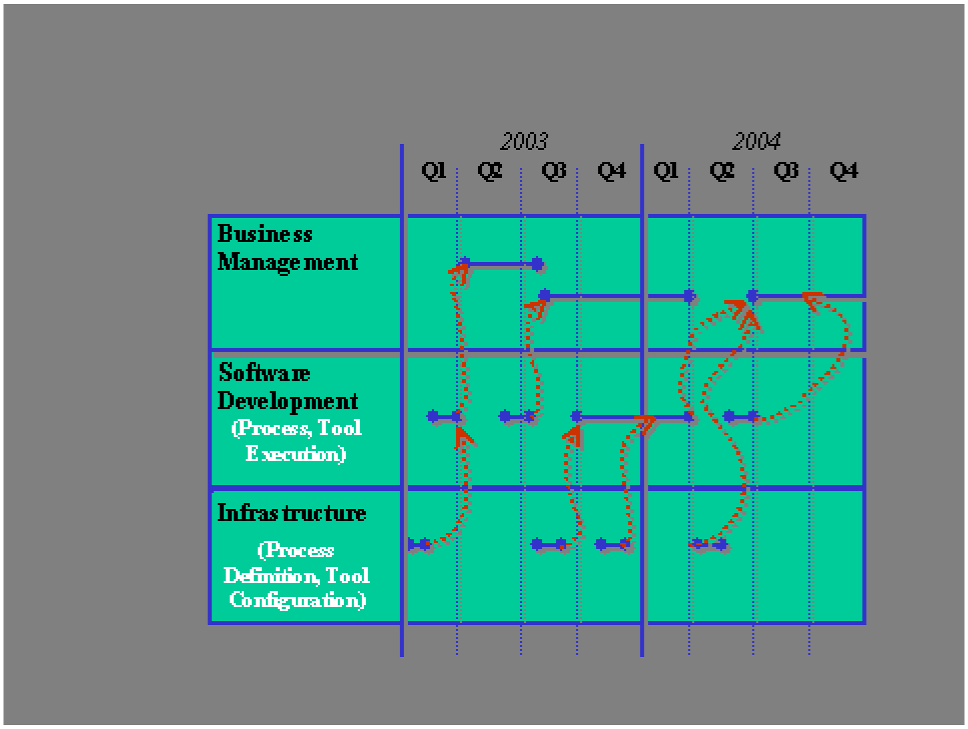
2003
2004
Q1
Q2
Q3
Q4
Q1
Q2
Q3
Q4
Business Management
Software Development (Process, Tool Execution)
Infrastructure (Process Definition, Tool Configuration)

Figure 3: Applying new process and tools incrementally, within real projects, is the best way to achieve successful technical and business results.

Guidelines for successful process change

As we have noted, changing behaviors and daily habits of people working on teams drives successful organizational change. Many process change efforts fail because managers assume that if people know what to do, they will do it. This is rarely the case. Instead, the organization must lead people through the change.

To explain how to do this effectively, we will begin with three basic steps.

Step one: Identify the business results you want.

To ensure that you choose the right process and tool improvements for your organization, you must first define the business results you want to achieve. These might include reducing costs, improving quality, or shortening time-to-market. For example:

- An insurance company might want the ability to develop applications more quickly so that it can provide new services to clients ahead of its competitors. In this instance, instituting a process with numerous, time-consuming cross-checks to ensure high-quality (i.e., low-defect) systems would not be a good business strategy[1]; the company must build applications rapidly in order to stay within its windows of opportunity.

- A medical device manufacturer, in contrast, does need to build software applications with high reliability. System failures might cause patient deaths and have disastrous consequences for the company. In this instance, extremely rigorous process and quality control is mandatory to achieve the desired business result.

As these two examples illustrate, it would be counterproductive to identify and prioritize targets for process change before you understand what business results you want to achieve. As you define desired business results, it is important to confirm them with a full spectrum of stakeholders within your organization. In addition, work with these stakeholders to define a set of objective measures that you can use to evaluate the success of your change effort.

Step two: Determine an appropriate pace for adopting change.

Organizations have different tolerances for change. Those that have been through successful change initiatives are likely to embrace new ones, whereas those with a mixed or poor record of success are often resistant to change. In either case, work within the organization's threshold for change; if you introduce too many changes too quickly you run the risk of destabilizing the organization and actually making things worse for a period of time.

Project managers, guided by experienced mentors, must simultaneously convey a sense of urgency about making necessary changes and control the pace of change to prevent the team from feeling overwhelmed. Some level of discomfort is acceptable (and even desirable), as long as the team recognizes that it is progressing toward achieving the desired business result.

Demonstrating the value of the new process and tools as early as possible in a *real* project is imperative for managers. It is the only way to gain credibility and enlist the trust essential for making further improvements. In many RUP projects, you can see measurable results as early as the first iteration in the Elaboration phase.

If your pilot projects are successful, you can begin rolling out the process and tool improvements at the enterprise level. Organizations should not try to make changes on this scale until they build a track record of project success.

Step three: Use an iterative approach to make continual improvements.

Process and tool improvements are only a means to an end; you should introduce them selectively and incrementally to achieve desired results. As you do this, assess the needs of the project team to identify where you should make changes to achieve near-term objectives. To be effective, start with the basics. For example, if a project team does not have a basic *release engineering* capability[2] there would be little value in introducing parallel development activities (including component outsourcing); you would probably not be able to integrate the results effectively. Similarly, if the team lacks the ability to understand and document requirements and project risks, there would be little value in introducing asset-based development or architectural reuse, since the team would not understand the requirements that should drive these activities. Many projects fail because managers ask teams to run before they can walk.

Manage the expectations of all parties carefully. Temper your team's tendency to want to do everything immediately; it is critical to maintain a sense of proportion and a reasonable pace. Drive process and tool improvements fast enough to achieve results as quickly as possible, but not so fast that the team gets confused, disheartened, or stalled because too much change occurs at once.

The most effective method for introducing change is to tie improvement efforts directly to the work being performed. Use the project and iteration planning the team currently does to drive the introduction of new tools and techniques -- just ahead of when you actually need to apply them. Through a series of focused workshops followed by hands-on mentoring, you can jump-start changes and facilitate skills transfer. Here is an effective way to structure these workshops:

- Early in the project, hold a workshop to create the overall project plan, the initial iteration plan, and the requirements management and change management plans. Following the pacing guidelines we described above, define your iteration plans to introduce just enough process to meet the iteration's objectives. That way, the project team can learn the process by applying it.

- At the transition point between iterations, hold another workshop to review iteration results, plan the next iteration, and identify issues and action plans. Include mid-project adjustments to the process of the project. You can adjust to the pace of process adoption.

Also hold follow-on mentoring workshops to help team members develop the skills they need to execute the iteration plan. For example, during the Inception phase, project teams will need to understand the problem and create a vision for the solution; a workshop led by an experienced facilitator/mentor is the most effective way to make progress on this front.

During the iterations, mentors should work with the team to guide them in RUP activities identified in the iteration plans, which result in the creation of RUP artifacts. Mentors can help team members learn how to play their roles with increasing proficiency as the project progresses. This approach is called *experiential learning* (or learning by doing). If you think of software development as a kind of team sport, then the mentor is a team coach who helps players achieve better results.

More about mentoring

Now that we have seen the basic steps involved in successful change, we can discuss a few particulars about mentors. For example, who should be a mentor? Initially, unless you have process experts on staff, you will need to hire mentors from outside your organization. Over time, these external mentors will transfer knowledge to resident mentors in your organization, so that they can support the process and tools on their own. Typically these are senior technologists, experienced practitioners who can lead or own various RUP disciplines. When they are ready, these professionals will divide their time between applying the process in real projects (60 to 70 percent) and mentoring other projects (30 to 40 percent). This will ensure that they have current knowledge in their respective disciplines. Some organizational change is also required. Mentoring will have to be part of these people's formal job responsibilities so that they can give it the proper amount of attention; it should not be a *spare time* task that they feel obliged to push off in favor of more critical activities.

The organization should recognize and reward mentoring as a key leadership skill.

Growing an organizational process
As a project progresses, mentors should document the process the team applied during each iteration. This enables the organization to collect experiences across projects; it also enables teams not only to apply process changes in subsequent iterations of current projects, but also to use them as the basis for new projects. This results in a RUP-based process that is grounded in practical experience and can grow to support the organization's specific needs. Instead of being a theoretical exercise, process definition emerges from the experience of solving problems and delivering results.

Summary

The mentoring approach we have described does not focus on defining a process before the project starts. Initially, the project team spends virtually no time writing the development case, customizing the template, and creating a RUP Web site.

Although this may seem strange, it is the best approach. Demonstrating the change's value as soon as possible via technical and business results helps overcome people's resistance to the change and build support for it. The only effective way to achieve these results is through an actual project; if people perceive the change effort as separate from their "real work," it will *never* produce results. Spending a lot of up-front time defining/configuring the process is also counterproductive.

With the guidance and leadership of an effective mentor, and with the support of management to measure and reward positive results, teams can improve their process while getting real work done. This mentoring approach emphasizes that process improvement and project work are not mutually exclusive. IBM Rational has applied it successfully in many engagements across a wide variety of industries and project sizes.

Lessons learned through these engagements include the following:

- Software capability improvement initiatives must be led by experienced practitioners who work with project teams in hands-on mentoring roles.
- Organizations can achieve sustainable technical and business results only by introducing new process and tools in real projects.
- Demonstrable results are essential to making lasting improvements. Unless people see results they will not commit to a change.
- To assess the success of a process change, managers must look at the *real* process that people are following; and a process is only as good as the people applying it.
- Tools are only as good as the process they automate. And they are useful only in the hands of skilled people. Just as putting power tools in the hands of inexperienced users can hurt them, putting software automation tools in inexperienced hands will not only keep a project from achieving the desired results, but may also cause harm.
- Managers should conduct regular status reviews to manage the expectations of key stakeholders. These people should be kept informed about project progress and any obstacles the team has encountered. Often, their support and sponsorship are essential to building support for a change among project team members; their input is also essential as organizational priorities shift.

Remember: Demonstrating the value of a solution is the only way to build credibility; this, in turn, lays the groundwork for a culture of continuous improvement.

References

For more information about IBM Rational Unified Process, or RUP, see http://www-306.ibm.com/software/awdtools/rup/.

Although many books on the market introduce RUP concepts, the one that specifically addresses the adoption process for RUP is: Stefan Bergström and Lotta Råberg, *Adopting the Rational Unified Process: Success with the RUP*. Addison-Wesley, 2003.

Notes

[1] Applications need to meet a threshold quality objective but typically need not be "perfect"; they must, however, be "good enough".

[2] The ability to manage versions of software, to do reliable daily builds, and to control change.

About the authors

Worldwide communities of practice architect Kurt Bittner joined Rational Software ten years ago and has worked in the software industry for more than twenty-one years. Improving the effectiveness of software development organizations, especially through software engineering management and process change, has been his principal focus. A member of the original IBM Rational Unified Process, or RUP, development team, he later managed the RUP development organization as well as other product development organizations inside IBM. In addition to co-authoring *Use Case Modeling* (Addison-Wesley, 2002), he has also contributed to several other books and authored numerous articles.

Saif Islam, a services manager/advisory consultant in Rational Brand Services, specializes in enterprise adoption of IBM Rational Unified Process, or RUP, and process automation tools. Experienced at managing large programs and projects for both IBM Rational and its clients, he helps teams achieve their technical and business goals by applying IBM Rational's process, tools, and services.

His more than twelve years of experience in software development encompasses an array of disciplines, including project management, system analysis, design, and architecture. In addition to teaching a class on object-oriented architecture and design (OOAD) for the graduate program in software at the University of St. Thomas in Minneapolis, he also gives seminars on technical topics at the University's Graduate School of Business.

Share this....

Digg this story del.icio.us Slashdot it!