Standards, compliance, and Rational Unified Process

Integrating RUP and the PMBOK

William Cottrell

May 26, 2004

from The Rational Edge: This article explains the relationship between IBM Rational Unified Process, or RUP, and the PMBOK, the Project Management Body of Knowledge, maintained by the Project Management Institute, or PMI. It describes the relationship of RUP to industry standards, what compliance means, how to leverage standards to improve your tailored use of RUP, and how you integrate those standards with RUP to achieve business value.

Standards -- how some people hate that word. And while I don't share their distaste, I do understand it; nearly every day, I have some reason to discuss the relationship of industry standards to Rational Unified Process®, or RUP®. Some think of standards as their nemesis. I think of standards as an opportunity for companies to better integrate their software development efforts with the business and legal processes defining much of today's corporate landscape. True, software teams may not be involved in most audits, inspections, certifications, or assessments, but more than likely, IT teams are heavily involved with the results in the form of financial, organizational, and/or business process improvements.

What drives us to pursue compliance with industry standards? Consider a competitive merger in which the acquired company may already be compliant with some industry standard and the acquiring company wants to achieve the same level of compliance, company wide. Or consider a company deciding to outsource some part of its business that is not a core competency. More than likely, the external supplier is using standards that may or may not be part of the hiring company's current business processes. Finally, consider a regulatory audit. Any deficiencies will no doubt require changes in business processes. That means software development could be affected. The list of drivers goes on, and all are cases for standards adoption.
Believe it or not, there are industry standards, guidelines, and, yes, processes that can help us all achieve business goals, resolve business process deficiencies, gain a competitive edge, do more with less, or simply provide the equivalent of a thesaurus for better communication.

Over the past two decades, I’ve seen -- or been a part of -- my share of process improvement projects, and I’m now convinced that industry standards and guidance are not about whether your company is compliant with them, but whether you can leverage them to improve your processes. Yes, it is sometimes important to have that banner of compliance hanging outside the office building, but the bottom line is, if your improvements don't add value, the improvements will be superficial.

RUP is designed to be your business process for software development. While RUP embodies many industry best practices included in industry standards, in general, it is not designed to be compliant with industry standards. But RUP can be, and is, implemented -- with some tailoring -- to help a company become compliant with industry standards. In other words, RUP is a vehicle to help achieve business goals, gain a competitive edge, or simply provide a better communication vehicle so industry standards can add value in your workplace.

One such standard or guide is the Project Management Institute’s "Project Management Body of Knowledge®," or PMBOK®. The PMBOK "describes the sum of knowledge within the profession of project management". Its stated purpose is to "identify and describe that subset of the PMBOK that is generally accepted" and to "provide a common lexicon within the profession and practice of talking and writing about project management," but clearly some practitioners think of the PMBOK as their software project management process and procedures.

This article explains the proper relationship between RUP and industry standards, what compliance means, how to leverage standards to improve your tailored use of RUP, and how you integrate them with RUP to achieve business value.

If you are not familiar with RUP or the PMBOK, start by reading about each in the next section. Otherwise, head straight to the section entitled How RUP and the PMBOK relate to each other.

Relating RUP and the PMBOK
To understand how RUP and the PMBOK relate to each other, you must first understand their respective concepts and frameworks.

What is RUP?
RUP is a risk-driven, use-case-based, and architecture-centric, iterative software development process. RUP embodies industry-standard management and technical methods and techniques to provide a software engineering process particularly suited to creating and maintaining component-based software system solutions. RUP communicates roles, activities, and artifacts organized by process workflows that guide project teams through software engineering disciplines under the purview of operational business phases and decision making milestones.

If you are not familiar with RUP or the PMBOK, start by reading about each in the next section. Otherwise, head straight to the section entitled How RUP and the PMBOK relate to each other.
RUP’s foundation consists of three key elements: the role, the activity, and the artifact, as shown in Figure 1. The role performs activities and produces artifacts. Each role is primarily responsible for a set of activities and artifacts. But all roles will contribute to other activities and artifacts. Roles, activities, and artifacts are used repeatedly during the execution of workflows. The workflows form a sequence of tasks unique to each of the nine software disciplines in the RUP iterative development software lifecycle framework (see Figure 2).

**Figure 1: Key elements of IBM Rational Unified Process**

![Figure 1: Key elements of IBM Rational Unified Process](image)

The RUP framework is two dimensional, with axes indicating *time* and *content*. The time dimension is organized by phases, iterations, and milestones. The content dimension consists of software disciplines containing the workflows, roles, activities, and artifacts as they apply to that discipline.

You implement the RUP framework via a complementary toolset, the capabilities of which generally map to the types of activities and artifacts required (Figure 3).

**Figure 2: RUP framework**

![Figure 2: RUP framework](image)
Figure 3: The RUP framework is implemented via a complementary toolset

As shown in Figure 3, RUP consists of five distinct parts:

1. **The RUP process framework.** This is the knowledge base of industry-proven best practices that forms the heart of RUP.

2. **The process delivery tools.** These are the tools that deliver the valuable process content to the practitioner when needed, in the form and quantity they need.

3. **The Rational Process Workbench.** This consists of RUP Organizer and RUP Modeler. RUP Organizer allows you to create simple plug-ins that complement, without altering, RUP's underlying structure. RUP Modeler allows you to create structural plug-ins for RUP that change RUP's underlying meta-model.

4. **The Configuration tool.** Otherwise known as RUP Builder, helps RUP users configure a base RUP configuration with the plugins created in RUP Organizer and RUP Modeler.

What is the PMBOK?

The PMI PMBOK is a basic reference for those interested in or already working in the project management profession. It contains a subset of the body of knowledge maintained by project management practitioners and academic institutions. That subset includes the generally understood best practices that are widely used for project management in a wide variety of industries. Like RUP, the PMBOK describes key elements and processes, and it defines a project management framework. But it does not provide a prescription for using them in the context of software development. Rather, it is a general guideline for project management in any industry. The PMI expects experts in their specific industries to apply these guidelines to their respective business processes.

The key elements include roles, project management processes, and artifacts. Just as in RUP, the role performs the process activities to produce artifacts.

The PM role, project management processes, and artifacts are grouped in the project management discipline as knowledge areas. The PM processes describe best practice details for each knowledge area. The PMBOK framework consists of process groups, knowledge areas, and project management (PM) processes\(^2\) (Figure 4). The knowledge areas group the PM processes by project management content. That is, we can categorize the content of the PM processes into one of nine knowledge areas. The process groups (initiating, planning, executing, controlling, and closing, etc.) organize the more detailed PM processes over time.
Figure 4: The PMBOK PM process matrix
If we illustrate the framework in a diagram (see Figure 5) showing level of activity for each process group based on the time and content dimensions, we see a relative weight of activity over the project lifecycle that somewhat resembles that of the RUP framework diagram (see Figure 2 above).

Figure 5: PMBOK framework
Relating RUP and the PMBOK
Now that we understand the RUP and PMBOK concepts and frameworks, we can compare them to help us understand how they relate. We will compare their utility to determine how one framework fits with the other. Here is the comparison:

- The PMBOK describes guidelines based on industry best practices. RUP helps software development teams implement software industry best practices. While RUP is tool-independent, when you use it in conjunction with IBM’s software development tools, you can significantly improve productivity, completeness, reusability, and more.
- The PMBOK describes a generic project lifecycle. RUP prescribes a generic software development lifecycle within a project lifecycle.
- The PMBOK describes guidelines for any size project. RUP can be tailored to implement any size software project.

In making the above comparisons, my point is that the PMBOK describes project management best practices and RUP prescribes -- helps us implement -- software development best practices, some of which are related to project management. This is a key differentiation that helps answer the question I am often asked: "Does the PMBOK fit into RUP, or does RUP fit into the PMBOK"?

The answer is “neither.” Why? Because the PMBOK by its own definition is designed to be applied to your existing business processes. Therefore, we can implement RUP as our software
development business process, tailor it to our company, a line of business, a department, a program, or some other organizational unit, then apply PMBOK best practices. So, how does all this fit together with respect to RUP and PMBOK utility? Let's look at a few pictures to try to explain.

**Figure 6: The PMBOK framework is implemented during each iteration within a RUP project**

Figure 6 illustrates how the PMBOK framework is implemented during each iteration within a RUP project. A RUP project uses PMBOK best practices in every iteration, in all four RUP phases (Inception, Elaboration, Construction, and Transition) as part of the project management discipline. That means we need to tailor RUP to the PMBOK key elements. While the PMBOK is a framework and guidelines, it implies some roles, activities, and artifacts; so, we will consider the PMBOK as an existing process and incorporate its best practices into RUP.

**Tailoring RUP to PMBOK best practices**

At the risk of sounding too much like a commercial, tailoring RUP is now easier than ever with the use of the Rational Process Workbench. Once we know how and why we are going to tailor RUP, we can build plug-ins to configure into RUP. Unfortunately, we still have to apply some brain power to the accomplish the what and how part of task. It would be wonderful if all we had to tailor was the RUP Project Management discipline. Unfortunately, project management activities are embedded in more than just this discipline.

The first step is to find the things you need to tailor. Next step, figure out how to capture and communicate those changes.
Find what needs to be tailored

Below, I outline the steps to find what you need to tailor. Teams should document the results of these steps in whatever manner they wish, but I will offer an example of a result in a simple here:

1. Decide what configuration of RUP to start tailoring. RUP comes in three sizes: small, medium, and large (called Classic RUP). Determining how to choose one of these three is beyond the scope of this article. But for example purposes, I will use the RUP Classic configuration (the full RUP). There will be more roles, activities, and artifacts to review for tailoring. So choose wisely.

2. Ensure that you are familiar with the details of PMBOK framework elements and the RUP key elements. I would not start this activity without having thoroughly read both the PMBOK and the RUP content (at least the role-based material) so that you retain some of the content from each.

3. This step will take you a long way in your understanding of RUP and PMBOK content. To tailor RUP with the PMI best practices we will need to build a map of PMBOK roles, PM processes, and outputs to RUP roles, activities and artifacts. But the mapping is not itself the tailoring we require. It is only the first step. While I have seen a number of mappings of RUP and the PMBOK, I would suggest that anyone looking to tailor RUP with the PMBOK best practices might want to do their own mapping. First, it is not that difficult, because there are not so many roles, activities, and artifacts in the PMBOK. Second, the mapping effort will give you a deep appreciation and insight into the content of each framework. Third, there is enough subjectivity in a mapping that you would probably tailor any existing mapping anyway, just to create consistency with your environment. For these reasons, I recommend you do this mapping yourself and be confident that it fits your organizations project management paradigm and that nothing has been overlooked. There are many ways you could build a map. I suggest you start with each RUP role diagram (there are about 30 in the full RUP configuration), each of which lists all the activities that that role must complete.

1. For each RUP activity in that role (there are about 150 activities for all roles combined in the full RUP configuration), map it to a PMBOK Process Group (Initiating, Planning, Executing, Controlling, Closing). You will not have to spend much time on each. The title alone will give you a hint if there is anything project management related or not.

2. Repeat this step for each role and collect all the RUP activities for PM processes.

3. Then compare the content of each PMBOK PM process for that process group in the PMBOK to the RUP activities associated with that group.

4. From that comparison, determine if you need to adjust any RUP input artifacts with any PM process inputs, RUP steps with PM process tools and techniques, or RUP resulting artifacts (there are more than a hundred for all roles combined in the full RUP configuration) with PM process outputs. (Remember, this part of the process is very subjective, which is why you should do your own mapping.)

5. If you find that any changes are required, write them down.

6. Repeat these steps until you covered all activities for all roles; that should include all artifacts, too.
For the purpose of this article, I will provide an initial mapping of all the PM Process Groups to the RUP Project Manager role and associated activities. This mapping is shown in Figure 7.

Figure 7: RUP - PMBOK mapping example

Now, let's examine the first PMBOK Process Group: Initiating. Initiating, in red, is mapped to three RUP activities: Developing Business Case, Initiate Project, Initiate Iteration (also shown in red). Each PM process is mapped to the RUP activities. I repeat that for each role. Table 1 shows Initiating mappings to RUP activities; for the purposes of this article, I will only trace the Initiating mappings to RUP.

Table 1: PMBOK Initiating PM process group mapping to RUP activities

<table>
<thead>
<tr>
<th>PMBOK process group</th>
<th>RUP roles affected</th>
<th>RUP activities</th>
<th>Changes to ask</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiating</td>
<td>Project manager</td>
<td>Developing business case, Initiate project, Initiate Iteration</td>
<td>Include company project selection methods using our portfolio analysis process by modifying those steps under Developing the Business Case.</td>
</tr>
<tr>
<td></td>
<td>Management reviewer</td>
<td>Project approval review</td>
<td>Add Rational Unified Process work order artifact to the resulting artifacts list of the management reviewer and as input to the initiate project activity.</td>
</tr>
<tr>
<td></td>
<td>Business process analyst</td>
<td>Identify business goals</td>
<td>Add our company project selection criteria as input.</td>
</tr>
</tbody>
</table>
Note that in Table 1, I describe some changes to be made where, in my opinion, RUP does not satisfy the intent of the PMBOK guidelines; note also that I show only the exceptions. If I were trying to show my management or an auditor that I am complying with the intent of the PMI Standard, I would probably have to include all the mappings of RUP key elements to PMBOK elements. Once I finish this approach for all PMBOK PM Process Groups in each RUP activity for each RUP role, I can decide how I will make those changes.

**Capture and communicate the changes**

The simplest way to capture and communicate the RUP tailoring is to build a RUP development case. The development case template is available in RUP to capture the process after you have tailored it, and you can place this development case on the project Website for reference. In it you will have links to the tailored process. Everyone can review the tailored material can be reviewed in a team meeting prior to the start of a RUP iteration.

I could take this process a step further by illustrating how to build a plug-in using RUP Process Workbench (RPW) customization tools. As noted at the beginning of this article, these tools are RUP Organizer, which lets you make common customizations that don't affect the underlying RUP meta-model, and RUP Modeler, which you use to make structural plug-ins. But the details of that process could easily double the size of this article. So I will end this discussion here.

**Try this yourself**

I've seen or participated in my share of process improvement projects over the last couple of decades. In doing so, I have witnessed no more than limited success when the industry standard is used as the process for the company, line of business, department, program, or some other organizational unit. I've also experienced attempts to map a company's process to standards and guidelines, and those performing the mapping were certain that the effort would show the way to compliance, maturity, or certification. While mapping a standard to a business provides helpful insight into the standard and the process, it is only part of the effort needed to integrate them. Case in point: RUP and the PMI PMBOK.

By following the techniques outlined in this article, I believe you can tailor RUP to include the PMBOK best practices. Specifically, at IBM Rational, we have found that we can adjust RUP input artifacts with PMBOK process inputs, RUP steps with PMBOK process tools and techniques, and RUP resulting artifacts with PMBOK process outputs.

Once we have found what needs to be tailored, we have a few options to make those changes available to the project team. We can build a development case document, use RUP Organizer, and/or use RUP Modeler.

**References**

Notes


2 Op. cit. Reproduced here by permission of the PMI.

3 Editor's Note: An alternative method for mapping RUP and the PMBOK is detailed in Serge Charbonneau's article, Software project management: A mapping between RUP and the PMBOK," also published in this issue of The Rational Edge.

© Copyright IBM Corporation 2004
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)