BENEFITS OF INTEGRATING PMBOK® AND RUP

Project managers use the Project Management Body of Knowledge (PMBOK) as a guideline for the processes and tools necessary to manage projects. The PMBOK is a general discipline that can be used by a project manager in any industry. Within different industries, there are often methodologies used for the creation of the relevant products or services. In the field of software development, these methodologies are referred to as *software engineering methodologies.* One of the more popular software engineering methodologies used today is the Rational Unified Process (RUP). Both the PMBOK and RUP are targeted at executing a project that hits budget and schedule targets while delivering a product that satisfies the customer. Although the PMBOK and RUP have some similarities and overlap in process, they emphasize different aspects of the project and, as expected, the details of each methodology are quite different. At the highest level of abstraction, however, the essential difference could be captured as follows:

PMBOK is focused on ***planning and controlling management activities with an eye to project control,*** hence the emphasis on planning and change control.

RUP is focused on ***planning and executing the work activities with an eye to effective product creation,*** hence the emphasis on an iterative, phased approach.

Combining PMBOK with software engineering methodologies such as RUP can bring many benefits to the IT project manager. Here are some:

- **A defined approach or "track" for creating the project work schedule and deliverables.**

  PMBOK provides a high-level, Initiation-Planning-Execution-Control-Closing outline for projects. This generic approach supports the key activities of most IT projects: "define, design, build, deploy." RUP provides a specific approach with defined phases, iterations, deliverables, tools, and processes for creating the deliverables. RUP provides the details lacking in PMBOK that are necessary for IT projects, Still, RUP is a software-process *framework* in that each project is required to tailor, customize, and *implement* the RUP for its own unique circumstance. A RUP project manager is expected, therefore, to not only understand the framework detail, but also be able to instantiate from it a single, complete plan for the project at hand.

- **The relationship between the Work Breakdown Structure (WBS), which defines the product being developed and its sub-components, and the RUP deliverables required to support development of the product.**

  The benefit to the project manager using RUP is that the work activities and deliverables are already described; they need only be aligned with the project's WBS deliverables and then scheduled. In this way, more of the team's effort is focused on ***what*** they are trying to accomplish rather than on ***how*** they are going to do their jobs.

- **Defined roles for project team members, including their responsibilities, interrelationships, and deliverables.**

  PMBOK describes the roles, responsibilities, and deliverables of the project manager. RUP describes the same items for the software development team including analysts, designers, coders, testers, configuration management staff, etc. In a project combining PMBOK and RUP, the entire team is aware of their individual responsibilities and how the output of their work efforts contributes to the project, the product, and to the work efforts of other team members.

- **A common language between team roles that is integrated into the software engineering life cycle.**

  RUP provides a common language and model set which is used during the entire project life cycle. It is both Use Case driven and architecture-centric. Use Cases define requirements, system/user interactions, and design models utilizing the Unified Modeling Language (UML). UML diagrams are used to convert the text-based Use Cases into system design models. A key design artifact is one or more models of software architecture, which later becomes the basis for coding, testing, deployment and delivery of the final software product. These same artifacts are subsequently used throughout the project. As the project progresses, these artifacts evolve with greater degrees of detail and granularity. As they evolve, they become more "technical," however, the basic syntax for communicating between the team members stays the same. One benefit of this is that it becomes easier for technical team members to present and discuss their designs with less technical members of the team, such as analysts, testers, and subject matter experts.

- **Defined, repeatable processes that are useful for tracking historical information.**

  A further benefit to the use of software engineering processes for project managers is that they define structured, repeatable work processes. By tracking costs against these defined activities, it becomes easier for organizations to build historical databases of experience for future project estimations.

In our next article we will begin discussing the integration of RUP with specific PMBOK knowledge areas.

Craig D. Wilson
Information Technology Management Consultant
craigdwilson@matincor.com
www.matincor.com

Hadar Ziv, PhD
Professor of Information and Computer Science
University of California, Irvine
ziv@ics.uci.edu