

▶ Integrating Rational Unified Process and Six Sigma

by [Clay Nelson](#)

Sr. Technical Services Specialist
IBM Rational
IBM Software Group

and

[Mike Morris](#)

Technology Management Consultant
North Highland, Inc.

It is not possible to walk down the "Management" aisle at a bookstore without getting slapped in the face by a book about Six Sigma. Chances are your organization is either in the middle of a Six Sigma initiative, contemplating one, or at the very least wondering what all the fuss is about. Six Sigma is an approach to improving business processes, improving quality, and generating higher profits for an organization. However, if some well-intentioned manager gets the idea that your organization should use Six Sigma for software development practices, approach this idea with caution. Six Sigma was developed for use in the manufacturing world, and its application to the software industry is subject to much debate and conjecture.



Although there are benefits to using Six Sigma within a software development organization, you should understand its key concepts and limitations in order to determine the scope of its potential usefulness for your organization. We recommend combining Six Sigma with IBM Rational Unified Process® or IBM RUP® product, a proven software engineering process framework that is used industry-wide to improve the development of software products through the use of best practices. This article will help you understand how Six Sigma and RUP complement one another, and how they can be used together to enhance the effectiveness of your software development organization. We will include:

- ▶ [subscribe](#)
- ▶ [contact us](#)
- ▶ [submit an article](#)
- ▶ [rational.com](#)
- ▶ [issue contents](#)
- ▶ [archives](#)
- ▶ [mission statement](#)
- ▶ [editorial staff](#)

- *An overview of key Six Sigma and RUP terms, along with an explanation of some similarities and differences between the approaches.*
- *Lessons from the field, based on the experiences of a large IBM Rational customer who recently used Six Sigma to evaluate improvements structured around RUP. The application of Six Sigma helped the customer measure the quality of their product and process.*
- *A review of Six Sigma tools and techniques that can help a software development organization avoid common project pitfalls.*

What is RUP?

RUP is a software engineering process framework designed to be tailored for multiple types of software development and deployment projects. The best practices advocated by RUP, such as developing iteratively and using UML visual models, have been distilled from thousands of successful projects across the software industry. RUP provides guidance on how to execute these best practices by detailing the roles, artifacts (deliverables), and key activities of software development.

What is Six Sigma?

Assuming that you have been too busy to make it to the local bookstore, let's define what Six Sigma is and why it's useful. At its essence, **Six Sigma takes the best of process reengineering and applies a statistical rigor for improving process and product quality in measurable terms that are meaningful to the end-user, or customer.**

Statistical concept versus methodology

First, it is important to distinguish between Six Sigma as a statistical concept and the Six Sigma methodology.

- Six Sigma is a statistical concept that measures a process in terms of defects.
- Six Sigma is also the name of a methodology consisting of phases, tools, and techniques that help an organization improve its processes in a way that measurably impacts both customers and the bottom line.

The movement to use Six Sigma for *software* is more about using the methodology to achieve "continual process improvement" than it is about achieving a statistical Six Sigma process output. Because we firmly believe that an iterative approach to software development has numerous advantages over the traditional waterfall approach, we do not recommend using the Six Sigma lifecycle as a Software Development Life Cycle

(SDLC). Instead, we suggest integrating Six Sigma with RUP, at either an organizational or project level.

Six Sigma: The statistical concept

"Six Sigma is a statistical concept that measures a process in terms of defects. A defect is defined as any deviation from acceptable customer limits. If you were to think about measuring the performance of a business service or manufacturing process by percentages, the following statistics from George Eckes' *The Six Sigma Revolution*¹ would represent performance at 99.9%:

- One hour of unsafe drinking water every month.
- Two unsafe plane landings per day at O'Hare International Airport in Chicago.
- 16,000 pieces of mail lost by the US Postal Service every hour.
- 500 incorrect surgical operations each week.
- 22,000 checks deducted from the wrong bank accounts each hour.

Obviously, these are not the kind of results we can live with. Six Sigma advocates developing or improving processes that deliver only 3.4 defects per million opportunities,² or 99.99966% performance. However, Six Sigma involves more than just applying what was learned in a statistics class.

Six Sigma: The methodology

Six Sigma is also a business philosophy with a goal of continual process improvement in order to reduce costly defects inside an organization. As we noted above, it offers phases, tools, and techniques that help an organization improve its processes in a way that measurably impacts customers and the bottom line. However, use of the methodology *does not* mean you plan to achieve near perfect execution, or statistical Six Sigma output. It is up to management to determine the acceptable sigma, or level of variance, in a process.

Six Sigma projects follow a lifecycle designed to ensure that an organization does not jump straight to a solution without first understanding underlying problems, processes, root causes, and supporting

Six Sigma Terminology

Critical to Quality (CTQ): a subset of features that are critical to the customer's perception of quality.

Voice of the Customer (VOC): stated or unstated customer needs or requirements. Tools for capturing VOC provide links showing that project outputs will positively impact customer needs.

Black Belt: a certified Six Sigma subject matter expert who mentors or manages Six Sigma projects.

Tollgates: formal checkpoints between phases of the methodology.

RUP Terminology

Lifecycle: the end-to-end phases of a process across time. The RUP lifecycle comprises four phases: Inception, Elaboration, Construction, and Transition.

Development Case: a RUP artifact that is used to tailor the process for a

data.

project or
organization.

The Six Sigma methodology has two major lifecycle variants: DMAIC (Define, Measure, Analyze, Improve, Control) and DFSS (Design for Six Sigma). The more widely recognized DMAIC is used to *improve performance with existing processes*.

- **Define:** Lay out the problems to be solved by the project.
- **Measure:** Identify product requirements and Customer Needs (CTQs).
- **Analyze:** Understand root causes.
- **Improve:** Improve the process.
- **Control:** Maintain performance over time.

The other lifecycle variant, DFSS, is applicable to *the development of new products and processes*. DFSS projects do not have a universally accepted lifecycle; however, DMADV (Define, Measure, Analyze, Design, Verify) is most commonly used for process definition.³

- **Define:** Lay out the problems to be solved by the project.
- **Measure:** Identify product requirements and CTQs.
- **Analyze:** Understand root causes.
- **Design:** Design the new product or process.
- **Verify:** Verify that the new product or process meets customer needs.

Six Sigma and software development

Six Sigma was developed for use in the manufacturing world, and its application to the software industry is subject to much debate and conjecture. It is easy to measure a production process for manufacturing widgets because you repeat the same process for every widget that comes off the assembly line. Conversely, because every software project is different, most experts agree that software processes must be flexible frameworks, not rigorously followed sequences of actions. You would not develop a small Web application the same way you would develop a financial trading network. In his article "Six Sigma, Hardware Si, Software No,"⁴ Robert V. Binder asserts that "...the behavior of a software 'process' is an amorphous blob compared to the constrained, limited, and highly predictable behavior of a die, a stamp, or a numerically controlled milling machine."

Without a repeatable process, it is difficult to measure software development in the same way you measure process in the manufacturing world. Achieving 3.4 defects per million lines of code is not a valid quality goal for a software project. Binder references the defect density per thousand lines of code (KLOC) for the NASA Space Shuttle avionics system

as .01 failures per thousand lines of code.

For the sake of argument, assume that a Six Sigma software standard calls for no more than 3.4 failures per million lines of code (0.0034 failures per KLOC). This would require a software process roughly two orders of magnitude better than current best practices. It is hard to imagine how this could be attained, as the average cost of the shuttle code is reported to be \$1,000 per line.

Software design is as much art as it is science. David Hallowell points out in his article "Six Sigma Evolution for Software Systems"⁵ that companies engaged in Six Sigma initiatives, including IBM, Motorola, and Kodak, realized early on that "[Software] Design was special." Success requires more than the absence of negatives such as defects and delays; it is also about customer satisfaction and conjuring designs that can evolve along with business changes.

We cannot think about applying Six Sigma to software only in terms of the statistical concept. As we noted earlier, the movement to use Six Sigma for software is more about the use of the methodology and the goal of "continual process improvement" than it is about statistics. When Six Sigma advocates read the Standish Reports citing the poor state of software development projects in our industry, they believe that Six Sigma can help. These advocates, some of them technologists, some of them business people, have set out to improve the processes for developing software in their organization. As we will see later, this is exactly the kind of Six Sigma approach that is appropriate for software.

Some advocates also promote the use of the methodology for managing individual software projects. As is often the case with any technique or approach, people sometimes get a little overzealous and try to apply a good thing to everything. *Using Six Sigma lifecycles (DMAIC or DMADV) as the lifecycle (SDLC) for managing a software development project is where we believe the ground gets shaky.* Key activities in each of the Six Sigma phases indicate that DMAIC and DMADV closely align with the traditional waterfall development lifecycle. Years of experience have validated that a more iterative approach to software development has numerous advantages over a waterfall approach. Iterative development demands a focus on early risk mitigation by producing executable software. System integration is not pushed off into the future, and testing is not delayed until the end of development. *We firmly believe in these advantages of the iterative approach, and that is why we can't recommend a Six Sigma lifecycle for software development.*

However, some aspects of the Six Sigma methodology are extremely valuable and often necessary in the software lifecycle, especially if your organization "speaks Six Sigma." So now let us describe what we believe to be the right way to integrate Six Sigma into the widely accepted RUP methodology.

Integrating RUP and Six Sigma

Six Sigma is a methodology for process improvement, whereas RUP is a process framework for software development. A high-level comparison of RUP and Six Sigma is shown in Table 1.

Table 1: Comparison of RUP and Six Sigma

IBM Rational Unified Process compared to Six Sigma	
Similarities	<p>Focus on the root causes of business problems to solve the right problem. Both methodologies use tools such as fishbone diagrams and Pareto charts to ensure that the solution is addressing the real problem.</p> <p>Improve and use process. RUP and Six Sigma share a common philosophy that improving a process and then adhering to it can improve the product the process delivers. They also provide tools and techniques for modeling business processes.</p> <p>Focus on results as defined by the customer. RUP stresses gaining agreement on requirements by all stakeholders, and defining functional requirements using use cases, from the users' point of view. Six Sigma advocates the use of CTQs to align the solution with the customer's definition of a quality product. Both methodologies advocate gathering requirements from the customer; however, RUP is more specific about managing software requirements, whereas Six Sigma focuses on more general product or process requirements.</p> <p>Design quality into the product and relentlessly pursue overall quality. Both methodologies adhere to this principle.</p> <p>Use metrics to objectively measure performance. Using metrics is core to the philosophy of both frameworks. Six Sigma is a statistical measurement in and of itself.</p>

Differences	<p>RUP focuses on the software development process and Six Sigma on the business process. RUP details specific roles, activities, and artifacts necessary to deliver an executable software product. It is a process framework that can be tailored to individual organizations and further refined for each software project. Six Sigma provides a framework for business process improvement and techniques and tools for launching a new product or process.</p> <p>RUP supports the UML. RUP provides guidance on using the industry standard modeling language: the UML. Six Sigma does not specifically advocate or offer advice on a process notation.</p>
--------------------	---

Integration of the two methodologies can occur at either an organizational level or a project level. In other words, you can use concepts and tools from Six Sigma to help coordinate a project's business and software teams, or you can use it to help improve your organization's software development processes by adopting RUP best practices.

- **Project level integration.** You can achieve Six Sigma and RUP integration by mapping the lifecycles, tools, techniques, and terminology of the two methodologies. This creates a customized version of RUP that integrates in part with Six Sigma for a specific software development project. This approach gives software teams the best practices and guidance of RUP while allowing them to communicate with the business team that "speaks Six Sigma."
- **Organizational level integration.** Six Sigma can be used to improve a RUP implementation inside an organization. It is always advisable to adopt RUP in stages, usually by implementing one best practice at a time. Six Sigma provides a process improvement framework (DMAIC) to manage and measure the success of each stage. For example, if an organization determines that it does not manage requirements well, it will be beneficial to use Six Sigma to guide implementation of the RUP Requirements Discipline to address this problem.

The rest of this article explores these two distinct ways of integrating Six Sigma with RUP in more detail.

Project level integration: Managing a Six Sigma and RUP project

If your organization has adopted Six Sigma, managers will probably expect all employees to talk about project deliverables and metrics using Six Sigma vocabulary. If your organization is also using RUP, you may have struggled with how a single

Mapping RUP and Six Sigma Terminology	
RUP	Six Sigma

project can use elements from both of these methodologies. The best approach is to map both terminology (see sidebar) and content from both plans. For instance, you can add new roles, activities, or even disciplines to RUP to adapt it to the Six Sigma culture. You can also overlay the DMADV (or DMAIC) lifecycle with the RUP lifecycle.

Stakeholder Needs	Voice of the Customer (VOC) and Critical to Quality (CTQs)
Features	Product Requirements
Milestones	Tollgates
Iteration Assessments	Quality Audits

Mapping the lifecycles

Why is it helpful to map Six Sigma to the RUP lifecycle? We'll use an example to illustrate. Let's say your company has received complaints from customers that ordering products from your company is difficult. Management has initiated a Six Sigma process improvement project using the DMAIC lifecycle to investigate and resolve the issue. As you can imagine, any solution would probably include a new or improved software application, but there are many other elements to the solution that probably have nothing at all to do with software. For instance, one reason customers find the ordering process difficult might be that there are long wait times due to low inventory. Since RUP is a process that applies primarily to software development, it would not make sense to guide the entire process improvement initiative using RUP lifecycle phases (Inception, Elaboration, Construction, and Transition).

Instead, it would be more sensible to use one of the Six Sigma lifecycles -- DMAIC or DMADV -- and their tollgates to guide the overall program.⁶ In their article "Integrating Business and Software Development Models,"⁷ Wallin, Ekdahl, and Larsson suggest that both a general business methodology and a software-specific methodology are necessary for project success. Software development lifecycles "...do not ensure that resources are used in the right projects, that the market is available, or that the organization is ready for release. Similarly, business [lifecycles] do not support software development."

As we noted above, an iterative or spiral approach for software development has become a generally accepted best practice, but the DMADV and DMAIC lifecycles follow a waterfall model. Now, a waterfall lifecycle might be effective for other parts of the improvement project, but it would seriously hinder the software development team. In his article "Going Over the Waterfall,"⁸ Philippe Kruchten explains how to "marry" the two approaches: by mapping RUP to the waterfall process.

Figure 1 demonstrates a *conceptual* mapping of the two lifecycles. Many activities in Six Sigma's Define, Measure, and Analyze phases have an impact on the scope of a software project. In our order process example, the Six Sigma team may have realized early on (by interviewing order clerks) that the software currently used for entering orders was cumbersome. Although there may have been many other contributing factors, through analysis they determined that the software was a root cause of the customer complaints. That would have led them to initiate a software project, managed with the iterative lifecycle defined in RUP. As

we will discuss in the next section, many of the artifacts and findings from the Six Sigma portion of the project would function as inputs to the RUP Inception phase.

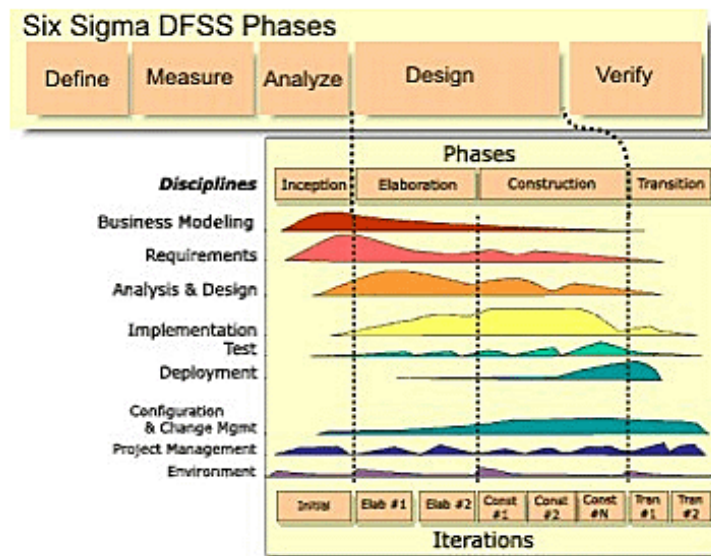


Figure 1: A conceptual mapping of RUP and Six Sigma lifecycles

Note that aligning the Six Sigma and RUP phases involves more than just lining up corresponding boxes on a diagram. The actual mapping of iterations and RUP milestones to Six Sigma tollgates is an activity that should be performed in the Project Management discipline of RUP.

Lifecycle mapping adds complexity to the software project, so recommend it only if you have a management directive to use DMAIC or DMADV. For a software development project, the RUP milestones are adequate business decision points for funding and go/no-go decisions. Undertake the lifecycle integration *only if it is absolutely necessary for successful coordination of the project.*

Mapping the content

Why do you need to map the content of Six Sigma and RUP? Because it's essential to coordinate business or client team members with software development team members. These two groups usually speak different languages to begin with; using two different methodologies only compounds the problem. When you map the content of the two methodologies, you can achieve agreement on deliverables, process activities, roles, and techniques. Without this mapping, coordination is extremely difficult.

The RUP disciplines contain details concerning roles, activities, artifacts, guidelines, and tool mentors -- in other words, the content of RUP. The vertical axis in the RUP diagram (Figure 1) shows the RUP disciplines. Mapping content from Six Sigma and RUP establishes consistency in the use of terms, roles, and activities between project team members and project sponsors. It also promotes agreement on the inputs and outputs from the various stakeholders, thereby reducing confusion about who does

what when, and to what extent.

What Six Sigma tools and terms should you incorporate into RUP? What role does the Black Belt play in the software development process? Figure 2 shows one way to modify elements in the RUP requirements discipline to align with Six Sigma terms. Instead of sitting down with a business stakeholder to talk about "Stakeholder Needs" (the RUP term), the analyst gathers CTQs. Mapping terminology is a powerful way to help business team members and clients understand how Six Sigma can fit into your software development process.

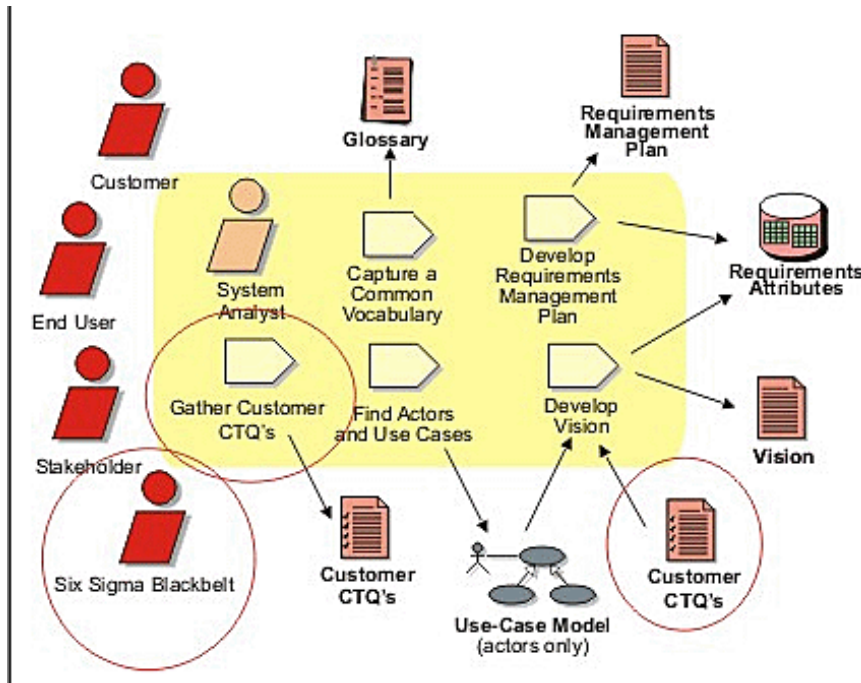


Figure 2. A conceptual mapping of RUP and Six Sigma content. (Six Sigma added or modified content is circled.)

The RUP process framework uses an OMG specification for defining what work will be done and by whom. It is intended to be customized for each organization and then modified further for each project within that organization; different organizations modify or replace standard RUP content to different degrees. In a project that combines RUP with Six Sigma, decisions about what and how much to customize depend partially on which methodology is most dominant in the organizational culture. If the business has wholeheartedly adopted Six Sigma, it may be in the best interest of all stakeholders to modify RUP content. This might be done simply with a Development Case, or it might involve more sophisticated solutions, such as developing a custom RUP/Six Sigma Web site using RUP Plug-ins and IBM Rational Rose XDE Developer® with IBM Rational Process Workbench.® Customizing RUP content usually means adding roles, activities, artifacts, guidelines, and tool mentors specific to the organization's or project's needs.

Organizational level integration: Process improvement with Six Sigma

A completely different way to view the intersection of RUP and Six Sigma is at an organizational improvement level. RUP implementation is best done in incremental steps, usually by adopting a couple of best practices and disciplines at a time. For instance, if you know that your organization needs to improve its requirements practices, implementing the Requirements and Change Management disciplines would contribute to improving the organization's performance.

RUP itself offers advice for implementation of the process within an organization. It also has points of "self reflection" called *iteration assessments* at the end of each iteration and phase as you execute a RUP project. Iteration assessments are a way of checking not just the product under development, but also how well the process is performing for the project. Most organizations face challenges executing and managing process improvements across the organization as a whole, not just on individual projects.

Improvement initiatives need the support of management and should be tied to business strategy. This is one of Six Sigma's strong suits. Many executives operate by the mantra "In God We Trust; all others bring data." Six Sigma stresses the importance of process improvements based on data analysis. It provides quantifiable ammunition to validate software process improvement efforts and helps ensure continued executive support.

Voice of the Customer (VOC)

Before you embark on an improvement effort, it is important that you understand how well a process is currently performing. There are many ways to measure the effectiveness of a software development process; however, only a subset of these measures will impact the process in ways that are important to the end user, or customer.

Using a Six Sigma technique known as Voice of the Customer (VOC), the process improvement team can identify which measures drive customer satisfaction. Improvements in these measures will impact the process and improve both the customer's

Lessons from the field

A large IBM Rational Software client recently used Six Sigma to target process improvement structured around the disciplines within RUP. The team built an organizational-level Development Case for the RUP workflows using Six Sigma's DMADV approach.

First, this client used Six Sigma to assess its customers' view of quality, with the following results:

- Customers expected quick execution, to match a dynamic and fast growing industry.
- The pace of the business translated into a need for more software features and quicker releases.
- In addition to reducing defects, the team focused on:
 - Reducing cycle time per iteration and release. Increasing the number of customer requests and new features included in each release.

perception of product quality and the *actual* quality of products and process.

Six Sigma tools and techniques help a software development organization avoid these pitfalls:

- **Implementing a process that is not easily measurable or that lacks the correct metrics for defining improvement.** Six Sigma forces a team to think in measurable terms.
- **Implementing a measurable process without an infrastructure in place for ongoing process measurement, management, and improvement.** When a Six Sigma DMAIC lifecycle is complete, the team can begin another DMAIC project to further improve the process.
- **Selecting metrics for a process because they are easy to measure or reflect positively on the organization but do not have meaning for the end user or customer.** With VOC, you will know what is important to customers.
- **Reinventing the wheel.** The benchmarking against industry best practices that you do in the Measurement phase does not preclude you from adopting a process rather than building your own. Configuring RUP to meet your organization's needs is a better, more cost-effective solution than building a process from scratch.
- **Relying on manual data collection instead of using automated tools.** The expenditure of time and effort necessary to gather and manage metrics manually is often not worthwhile. Using a tool such as IBM Rational ProjectConsole, a project team can collect metrics and display them on a project or organizational Web site.

Aligning their RUP implementation to these measures enabled the organization to make significant improvements in areas important to its customers, and included the following benefits:

- Cycle time per iteration was reduced from twelve weeks to eight on the pilot projects.
- Over the course of six months, average cycle time per iteration decreased to six weeks.
- The number of implemented features increased by 30 to 50 percent per iteration across business units.
- All of this was accomplished without negatively impacting defect measures.

The client has now begun to implement IBM Rational® ProjectConsole to track defect and cycle trends across projects.

Six Sigma allowed the team to increase its customers' *perception* of product quality as well as the *actual* quality of its product and process.

Extending the RUP Environment discipline with Six Sigma

The RUP Environment discipline focuses on the activities necessary to configure RUP for a specific organization or project. These activities

provide the software development environment -- which includes both processes and tools -- that will support the development team in achieving business objectives as well as IT objectives.

By extending the RUP Environment discipline with the Six Sigma tools and techniques we have described in this article, you can establish a reusable improvement framework for your organization. This framework will be capable of supporting ongoing improvement initiatives and garnering the management support imperative to success.

Successful integrated process improvement: Lessons learned

In our experience with clients, we have seen that process improvement teams who combine RUP and Six Sigma can increase their likelihood of success by applying some important lessons:

- **Train your core team in fundamental tools and techniques.** Successful teams ensure that their core process improvement people are fluent in:
 - RUP philosophy, terminology, artifacts, activities, and so forth.
 - Six Sigma process improvement techniques, including DMAIC and DFSS.
 - Basic principles of organizational change management.
- **Tailor methodology training.** Successful teams accept that some business people do not want or need to understand the finer points of each methodology. They avoid overloading software engineers with these details. To avoid confusion, they make sure that everyone understands the "big picture" of the processes, but provide detailed training only on the specific workflows or specialties that will affect each person.

Conclusion

As RUP becomes the de facto software development process standard and Six Sigma gains in popularity for general business process improvement, it is increasingly probable that software development organizations will be challenged to reconcile the two methodologies. Six Sigma and RUP share common goals: Focus on quality as defined by the customer; design quality into the product; and relentlessly pursue overall quality. Although integrating these two methodologies requires more than simply aligning goals, integration is indeed possible. It can be done on either a project or an organizational level, so it is critical to determine the purpose of using both methodologies, along with the scope and extent of their use.

Software development is a team sport. Consequently, it is important that all team members are "reading from the same music" and following a consistent methodology. For those looking to leverage the best of both

worlds, Six Sigma encourages a culture that champions quality results and provides some useful techniques for software development organizations. RUP provides software development teams with guidance on activities, roles, and artifacts that make industry best practices practical. Together, these two methodologies can drive predictable, measurable, and excellent software development results.

References

Books

G. Brue, *Six Sigma for Managers*. McGraw-Hill, 2002.

George Eckes, *The Six Sigma Revolution*. Wiley, 2001.

Philippe Kruchten and Per Kroll, *The Rational Unified Process Made Easy*. Addison Wesley, 2003.

Articles

Robert V. Binder, "Six Sigma: Hardware Si, Software No!" Whitepaper posted at <http://www.rbsc.com/pages/sixsig.html>

Christian Buckley and Carl Ashcroft, "Perfection by the Numbers: Key Learnings from Six Sigma." Posted on Rational Developer Domain (2003): <http://www.ibm.com/developerworks/rational>

David Hallowell, "Six Sigma Evolution for Software Systems." Six Sigma Advantage (2003): <http://www.6sigma.com>

Philippe Kruchten, "From Waterfall to Iterative Development: A Challenging Transition for Project Managers." *The Rational Edge*, December 2000. http://www.therationaledge.com/content/dec_00/m_iterative.html

Philippe Kruchten, "Going over the Waterfall with the RUP." *The Rational Edge*, September 2001. http://www.therationaledge.com/content/sep_01/t_waterfall_pk.html

Six Sigma Advantage, "The Software Industry Meets Six Sigma." Whitepaper posted at: <http://www.6sigma.com>

Christina Wallin, Fredrik Ekdahl, and Stig Larsson, "Integrating Business and Software Development Models." *IEEE Software*, November/December 2002.

Other

"RUP Plug-in Kit" from IBM Rational. See Rational Developer Domain at <http://www.ibm.com/developerworks/rational>

IBM Rational Unified Process. See <http://www.ibm.com/software/awdtools/rup>

Acknowledgments

Thanks to the many colleagues, friends, relatives, and clients who have reviewed and given feedback on the content of this article.

Notes

¹ George Eckes, *The Six Sigma Revolution*. Wiley, 2001.

² G. Brue, *Six Sigma for Managers*. McGraw-Hill, 2002.

³ To learn more about Six Sigma, read "Perfection by the Numbers: Key Learnings from Six Sigma," by Christian Buckley and Carl Ashcroft, available on Rational Developer Domain: <http://www.ibm.com/developerworks/rational>

⁴ See <http://www.rbsc.com/pages/sixsig.html>

⁵ See <http://www.6sigma.com>

⁶ At these tollgates (milestones), investment, progress, and go/no-go decisions are made.

⁷ Christina Wallin, Fredrik Ekdahl, and Stig Larsson, "Integrating Business and Software Development Models." *IEEE Software* (November/December 2002).

⁸ Published in *The Rational Edge*, September 2001:
http://www.therationaledge.com/content/sep_01/t_waterfall_pk.html



For more information on the products or services discussed in this article, please click [here](#) and follow the instructions provided.
Thank you!