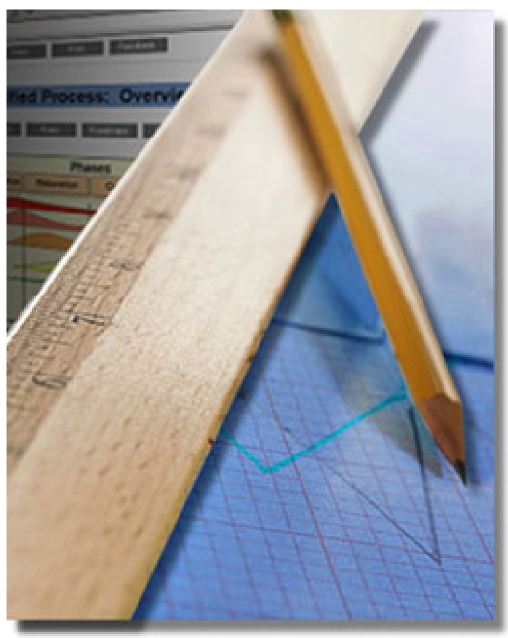


Practical Measurement in the Rational Unified Process

by **Doug Ishigaki**
Sr. Technical Marketing Engineer
Rational Software

and **Cheryl Jones**
Project Manager
Practical Software and Systems Measurement

> It is inherently difficult to manage what cannot be measured objectively.
> - *Walker Royce*[1]



Over the past twenty years, measurement has evolved from something that was a "check-in-the-box" process into a management best practice. Numerous case studies of successful projects and organizations have shown the value of measurement. It is now a key management discipline, as evidenced by its inclusion in various process standards, models, and guidelines such as the ISO/IEC 15939 standard, Software Measurement Process, and the Software Engineering Institute's Capability Maturity Model® Integration℠ (CMMI℠).

Although it is now widely accepted that measurement is a key best practice in software development, implementing a successful measurement program is still a challenge. Many projects and organizations still try either to build their measurement programs around the "ten best measures" or to measure so much that the program becomes a maintenance burden and fails.

There is much more to implementing a successful measurement program than just the measures themselves. The Practical Software and Systems Measurement (PSM) Initiative and the Rational Unified Process,® or RUP,® are two existing process frameworks that explain the measurement process. The PSM initiative team included government, industry, and academic representatives to bring together and *promote the best practices of software and system measurement. RUP contains a process framework of best practices for iterative, incremental, software and systems development. This article will show how PSM and RUP can work together to provide a coherent measurement process. We will introduce measurement concepts in the context of PSM and RUP and provide suggestions about which measures to apply in each RUP phase.*

Why Measure?

Measuring can be an effective management tool because it provides the information required for decision makers to accurately monitor key issues related to progress and quality, monitor performance against a plan, and ask the right questions. In other words, measurements allow managers to make the right decisions based on objective information.

Measurement helps managers do the following:

- *Communicate effectively*. Measurement supports communication among stakeholders throughout all levels of the organization. Objective measurement also reduces ambiguity. It is an effective way to communicate status between supplier and acquirer organizations.
- *Identify and correct problems early*. Measurement facilitates pro-active management. Identifying and managing potential problems early on is less costly than discovering problems later, when they are more difficult to manage. Effective managers do not wait for problems to arise; they anticipate problems.
- *Make informed trade-offs*. Decisions in one area often impact other areas. Measurement helps assess the impact objectively so that managers can make informed trade-off decisions to best meet project objectives.
- *Track specific project objectives*. Measurement helps managers track specific objectives such as: "Is the project on schedule?" or "Is the quality improving?" or "Is the system ready to be delivered?" By tracking actual measures against a plan, managers can visually see progress toward project and organizational objectives.
- *Manage risk*. Risk management is a widely accepted best practice. Known risks are

Standards, Methodologies, and Guidebooks

As illustrated in Figure S1, Practical Software and Systems Measurement (PSM), a product of the United States Department of Defense measurement initiative, served as the base document for the new international standard on measurement: the ISO/IEC 15939 Software Engineering -- Software Measurement Process*. This standard describes the measurement process in terms of the purpose and outcomes of a compliant process, along with associated activities and tasks.

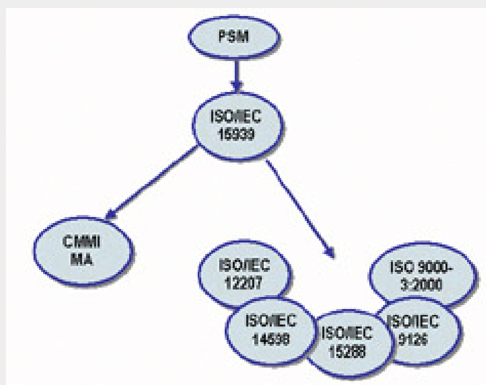


Figure S1: Relationships Among Measurement Standards and Methodologies

PSM provides additional details on the activities and tasks presented in ISO/IEC 15939, including detailed steps to successfully meet these tasks. In addition, PSM provides detailed how-to guidance, sample measures, lessons learned, case studies, and implementation guidance, all based on practical experience.

those that have been identified and analyzed; risks uncovered late in the project are typically more costly to fix. To be successful, you must be committed to identifying and addressing risks early in the lifecycle, when it is possible to address them efficiently. Measurement provides high-quality, objective data that affords visibility to risk areas. For example, requirements creep is a well-known project risk. Measuring and monitoring requirements volatility enables decision makers to determine whether the risk still exists or has been mitigated.

- *Defend and justify decisions*. Managers must do this effectively. Given a choice, the vast majority of managers would choose to base their decisions on objective data rather than subjective data. Measurement provides objective historical performance or trend data as well as current performance data. It provides perspective with respect to time, projects, releases, and so on. This allows decision makers to interpret the measures and decide on appropriate action.

Essentially, measurement provides the data managers need for progress assessment, insight into the quality of an evolving project, and objective decision-making. Although it cannot guarantee a project will be successful, it does help all decision makers take a proactive approach to managing critical issues inherent in software projects.

Attributes of a Successful Measurement Program

Now that you are (we hope) sold on the value of a measurement program, how do you establish one? Let's examine the steps taken by successful projects that established measurement programs and used measurement extensively as a management tool, including the CCPDS-R project[2] , the F/A-18E/F[3] , the Boeing 777 project[4] , and the Space Shuttle program at NASA/Johnson Space Center[5]. What do these successful measurement programs have in common?

- *A well-established and sustained organizational commitment*. Without this commitment, chances are high that the

The purpose and outcomes of the measurement process from ISO/IEC 15939 have been added to the revision of the ISO/IEC 12207, Software Life-Cycle Processes, within a new supporting process entitled Measurement. They have also been added to ISO 9000-3: Application of ISO 9001:2000 to Software. Measurement concepts from ISO/IEC 15939 have also been added to ISO/IEC 15288, System Life-Cycle Processes. Measurement terminology defined in ISO/IEC 15939 has been coordinated with the revisions to ISO/IEC 9126, Software Product Quality, and ISO/IEC 14598, Evaluation of Software Products, so that all of these standards now use common measurement terminology.

The draft international standard ISO/IEC 15939 was also used as input to the Measurement and Analysis (MA) section of the Capability Maturity Model® Integration℠ (CMMI℠)** The MA process area outlines activities for a measurement process and provides a methodology for assessing whether a project's measurement program is compliant with the international standard.

By coordinating these documents, the software and systems engineering communities currently have a consistent set of information-driven standards and guidance for implementing project and process measurement.

**ISO/IEC 15939 Software Measurement Process*. 2002.
** Software Engineering Institute, "Capability Maturity Model® Integrated (CMMI℠) for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing -- Version 1.1 Staged Representation." Carnegie Mellon University, March 2002.

measurement program will fail. Both business and technical managers must believe in and adopt the measurement program and carefully address how measurement results will be used throughout the organization. They should also work to mitigate concerns about using measurement to evaluate individual performance.

- *The organization actually uses the results*. Measurement should be information-driven, and they should be identified and implemented based on whether they address specific information needs as defined by project objectives, issues, and risks. Successful measurement programs evolve over time and should grow at a pace that allows all participants to learn and implement them effectively. A common mistake in measurement implementation is to try to do too much too quickly. To avoid this, teams should start with a few measurements and build up from there. Too many measures can be overwhelming; if you introduce many measures all at once, many of them will not get used.
- *The measurement process is well planned*. Before you decide on what measures you will implement, you should identify, prioritize, and document information needs within your underlying software development process. Then, you can select measures based on these needs. You can also develop procedures for how data collection, analysis, and reporting will be performed. Managers should also define the criteria for evaluating the measures, as well as the measurement process. Effective measures are by-products of the underlying software and system development processes, so the underlying software development process should be followed when applying the measures. Otherwise, as the saying goes, "Bad data in means bad data out."
- *Measurement collection is automated*. Experience has demonstrated that the most successful measures are those that have been collected and reported by automated tools. This does not mean that you cannot collect measurements without automation. But automation reduces the amount of human intervention, which ensures consistency, and thus increases the quality of the data and decreases the cost of collection (compared to manual collection).
- *Measures are objective and unambiguously defined*. Objective measures are concrete numeric representations (such as numbers, percentages, and ratios) as opposed to subjective text representations (such as excellent, good, poor, on-track). Numeric measures minimize ambiguity. They provide an objective view of the project, rather than a view based on impressions, opinions, and subjective data, which is often conflicted and confusing.
- *The measurement process is continuously improved*. The process as well as the measures are continuously evaluated, and improvements are made and widely communicated. If you have measures that are not being used, drop or replace them.
- *Results are communicated*. A key to success is to establish effective communication among managers, work groups, and individual team members. Everyone in the organization should have access to the analysis results. Results can show progress, but they may also show setbacks. Both types of results are valuable, and they both increase the predictability of the outcome. If you see a negative trend but don't communicate about it for a month or two, it could lead to much bigger problems -- not only for the project, but also potentially for the organization as well as the entire company.

The Measurement Process Model -- Activities

Based on the experiences of a wide range of software and systems projects, we have seen that projects with successful measurement programs plan and establish a measurement process, carry out the process activities, and continuously improve the process based on lessons learned.

As Figure 1 shows, the Measurement Process Model as defined by PSM includes four primary activities:

- Establish and Sustain Commitment
- Plan Measurement
- Perform Measurement
- Evaluate Measurement

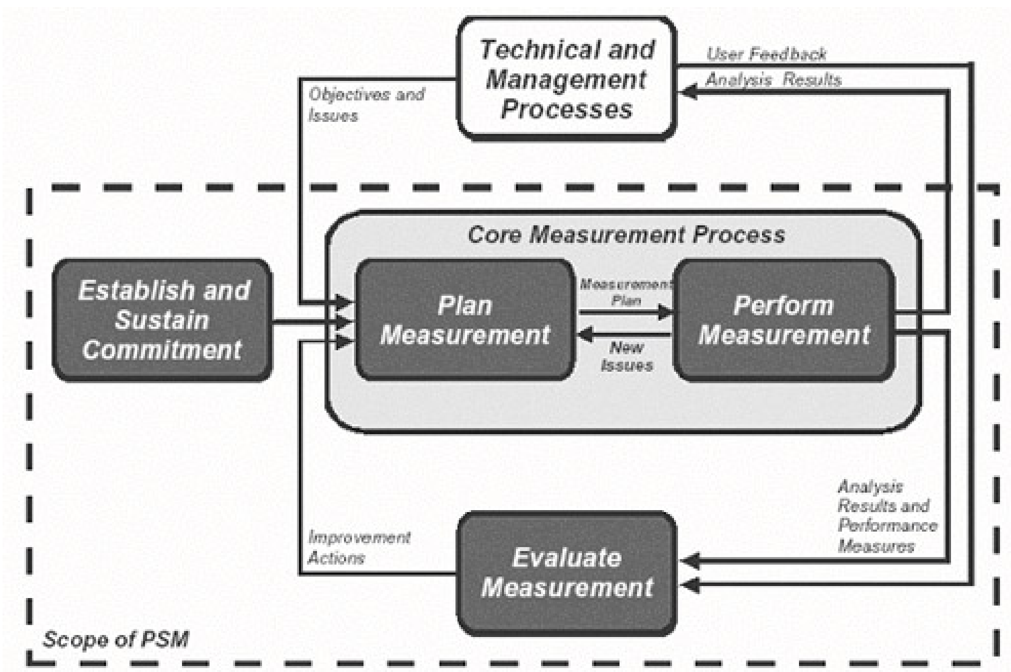


Figure 1: PSM Measurement Process Model

The Establish and Sustain Commitment activity involves obtaining organizational commitment from management as well as staff and resources for the measurement program. It also involves defining the scope of the measurement program -- for example, will it encompass a single project, multiple projects, an organization, or the entire company?

The Plan Measurement activity involves identifying information needs and selecting appropriate measures to address them. It also includes the following additional tasks:

- Define data collection, analysis, and

> **Rational ProjectConsole**
>
> Rational® ProjectConsole enables your project and organization to automatically quantify current project status and assess development trends with up-to-date measures. Measurement data is automatically collected from your Rational Suite development environment and select third-party tools and then stored in a measurement data warehouse. ProjectConsole allows you to

reporting procedures.

- Plan and define criteria for evaluating the information products.
- Define criteria for evaluating the measures and the measurement process.
- Address the resources and technologies that will be required to implement the measurement process.

The Perform Measurement activity encompasses performing and integrating the data collection procedures, performing the analysis procedure (aggregation, transformations, etc.), generating the information products that will present the analysis results, and communicating the information products to the measurement users. Perform Measurement basically implements the measurement plan.

The Evaluate Measurement activity consists of evaluating the information products as well as the measurement process and identifying potential improvements. The Evaluate Measurement activity ensures that the measurement program is continuously updated to address current information needs for the project or organization.

hyperlink measures together into indicators and reports that it then automatically organizes and updates in a project Web site (see Figure S2).

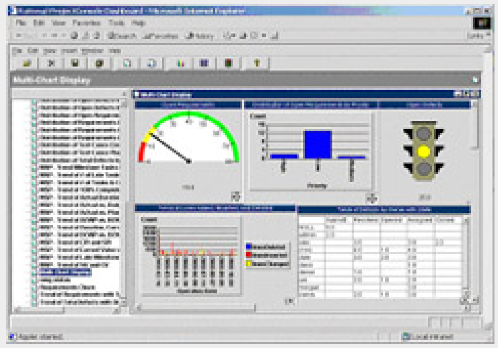


Figure S2: Rational ProjectConsole Displays Measures in a Variety of Formats

For more information, please visit the Rational ProjectConsole Web site.

A fifth activity, the Technical and Management Process, is also depicted in the Measurement Process Model but is not specifically covered in detail in PSM. That is because, technically, it is not a measurement activity. Managers and decision makers operate within this process, defining information needs and using the information products (results) from the measurement program to help them make informed decisions.

Measurement in the Rational Unified Process

The Rational Unified Process (RUP)[6] is a software engineering process. It provides a disciplined approach to assigning tasks and responsibilities within a development organization. Its goal is to ensure the production of high-quality software that meets the needs of its end users within a predictable schedule and budget.

RUP captures many of the best practices in modern software development and provides them in a process framework consisting of guidelines, workflows, templates, examples, and tool mentors.

Included in the RUP process framework is a project management workflow that serves three main purposes:

- To provide a framework for managing software-intensive projects.
- To provide practical guidelines for planning, staffing, executing, and monitoring projects.
- To provide a framework for managing risks.

As you might assume, measurement can be and should be an integral part of the workflow (Figure 2). Monitoring and controlling (through measurement) is performed throughout the complete lifecycle of the project, including managing iterations and phases as well as scope and risks.

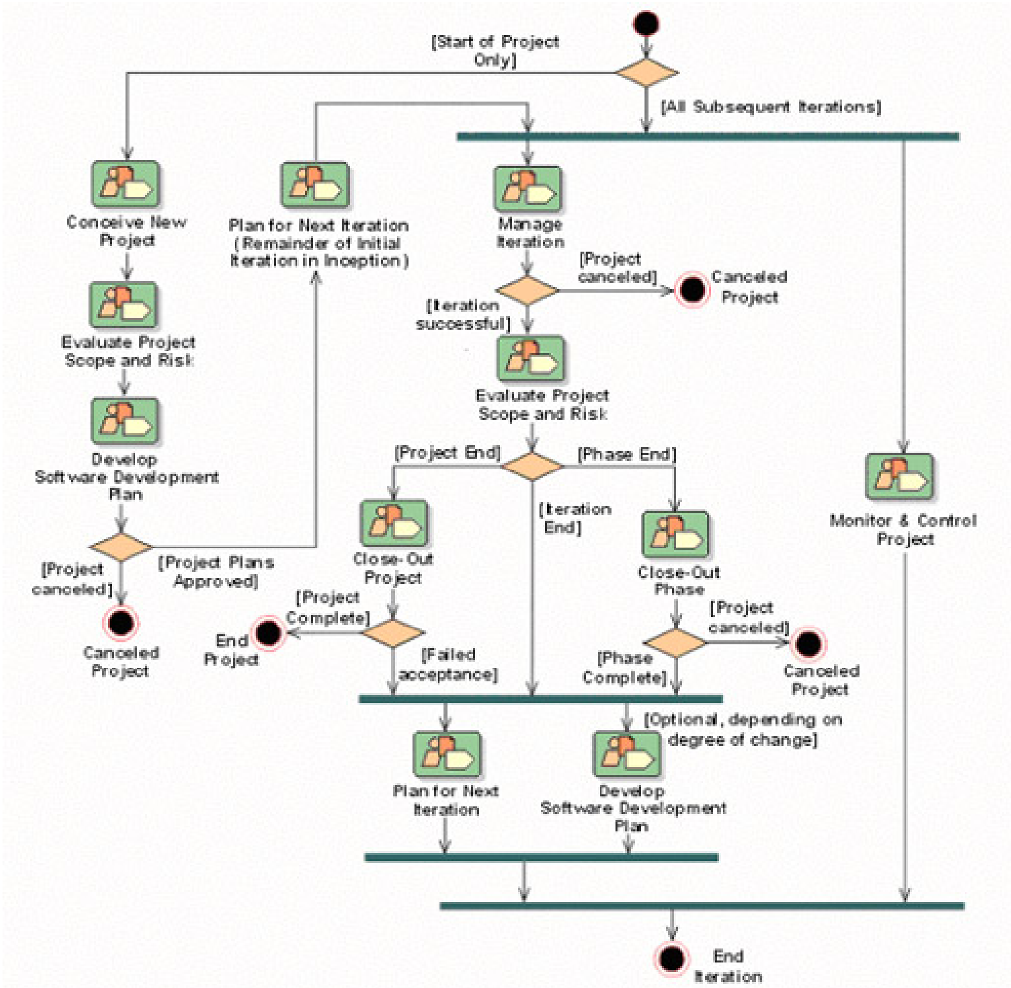


Figure 2: Project Management Workflow[7]

Drilling down further (Figure 3), RUP provides additional details on the activities as well as the artifacts that are produced. Specifically, RUP covers:

- Continuously monitoring the project in terms of active risks, issues, and objective measurements of progress and quality.
- Regularly reporting project status for review by the Project Review Authority (PRA), the organizational entity to which the project manager is accountable.

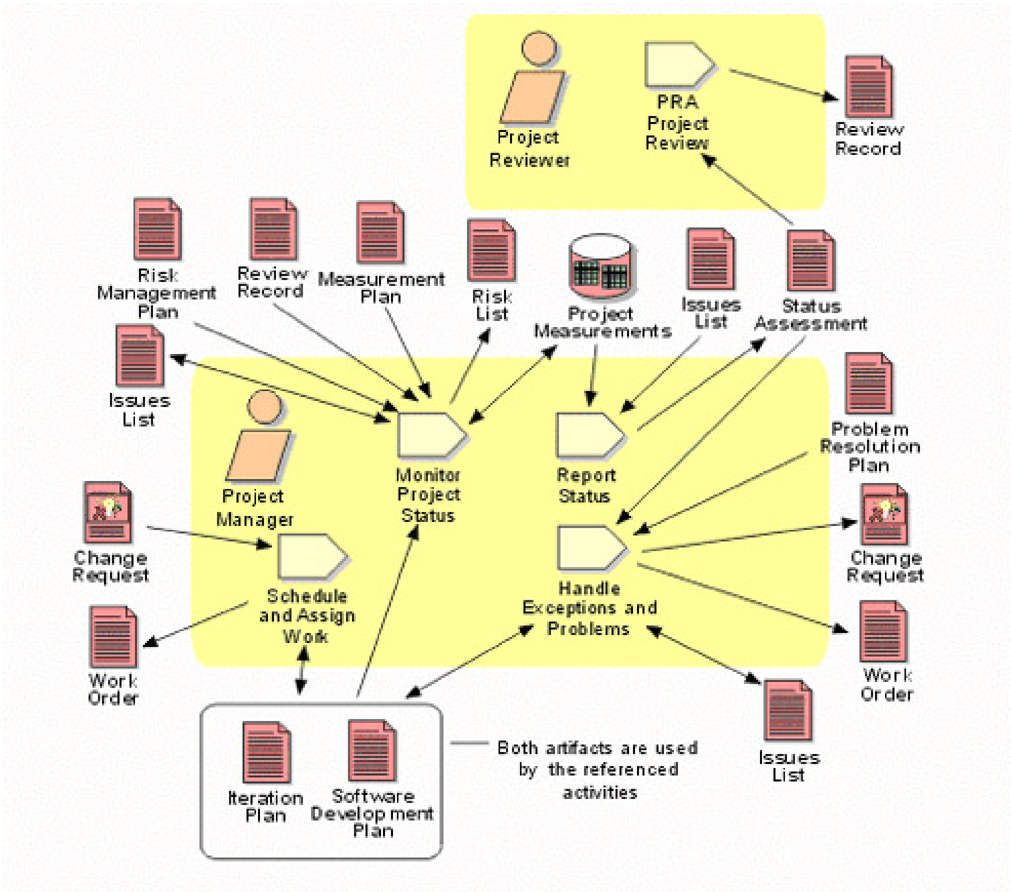


Figure 3: Workflow Detail: Monitor and Control Project

RUP also contains a template for a measurement plan. The measurement plan is an artifact (document) that defines the measurement goals (information needs) for the project, the associated measures, and the primitive (base) measures to be collected in the project in order to monitor progress and quality. In addition, RUP also includes sets of sample measures (a minimal, small, and complete set) to help project managers choose based on project information needs.

RUP is also a process framework that can be adapted and extended to suit the needs of an adopting organization. You can easily choose the set of process component and Plug-Ins[8] that are right for your project needs. In addition, you can extend RUP by capturing your own best practices into Plug-Ins.

Coverage of RUP and PSM

Both PSM and RUP emphasize the importance of measurement as an effective project management activity. At this time, however, there are differences between some of the measurement terminology and artifacts, as well as between the measurement process and activity flows. For example, the term metric, used in RUP, has been replaced by the term *measure*, in PSM (and in ISO/IEC 15939). *Measure* is now the term used in the worldwide software measurement community. Another difference is the measurement construct in PSM, versus the equivalent RUP metric template, which is documented in the measurement plan.

As shown in Table 1, most of the measurement activities in PSM are covered in RUP.

Table 1: Mapping of PSM and RUP Concepts

PSM	RUP
Plan Measurement: Identify information needs	Covered in Define Monitoring and Control Processes and Develop Measurement Plan activities
Plan Measurement: Select measures	Covered in Define Monitoring and Control Processes and Develop Measurement Plan activities
Plan Measurement: Define data collection, analysis, reporting procedures	Covered in Define Monitoring and Control Processes and Develop Measurement Plan activities
Plan Measurement: Define criteria for evaluating the information products	Covered in Define Monitoring and Control Processes and Develop Measurement Plan activities
Plan Measurement: Define measurement process	Covered in Define Monitoring and Control Processes and Develop Measurement Plan activities
Plan Measurement: Evaluate and select resources and supporting technologies	Covered in Select and Acquire Tools and Develop Measurement Plan activities
Perform Measurement: Integrate procedures	Covered in Monitor Project Status activity
Perform Measurement: Collect data	Covered in Monitor Project Status activity
Perform Measurement: Perform analysis procedures	Covered in Monitor Project Status activity
Perform Measurement: Generate information product	Covered in Monitor Project Status and Report Status activities
Perform Measurement: Communicate information products	Covered in Monitor Project Status and Report Progress activities
Evaluate Measurement: Evaluate information products	Covered in Develop Measurement Plan activity
Evaluate Measurement: Evaluate measurement process	Not covered
Establish Commitment: Define scope	Partially covered in the Initiate Project activity
Establish Commitment: Obtain commitment	Partially covered in the Initiate Project activity

Establish Commitment: Assign resources	Partially covered in Monitor and Control Project guideline

Information Categories

All projects have specific objectives that are typically defined in terms of system capability, resource budgets, milestones, quality, and business or system performance targets. Project success depends largely on how well these objectives are achieved. Project issues are areas of concern that may impact the achievement of a project objective: risks, problems, and lack of information, for example. Decision makers usually identify information needs to help track issues at the start of a project, but these might need to be reassessed as the project progresses, because objectives are different from phase to phase. In addition, changes in requirements, technology, resources, and other factors, often result in new information needs.

Experience has shown that most information needs can be grouped into general areas, called information categories. PSM identifies seven information categories, which represent key areas of concern for the project manager:

- Schedule and Progress
- Resources and Cost
- Product Size and Stability
- Product Quality
- Process Performance
- Technology Effectiveness
- Customer Satisfaction

These information categories help in selecting appropriate measures by allocating each identified project information need to one or more of the seven information categories. The importance of the information need and category is dependent on which phase a project is currently executing.

Table 2 shows the most common categories used in each of the RUP phases.

Table 2: Mapping of PSM Information Categories to RUP Phases

RUP Phase	Information Categories
Inception	Schedule and Progress Resources and Cost Process Performance
Elaboration	Schedule and Progress Resources and Cost Product Size and Stability Product Quality Process Performance Technology Effectiveness

Construction	Schedule and Progress Resources and Cost Product Size and Stability Product Quality Process Performance Technology Effectiveness Customer Satisfaction
Transition	Schedule and Progress Resources and Cost Product Quality Customer Satisfaction

Table 2 shows information categories that are crucial in each phase. Of course, you can also capture measures for information categories not listed within the phases. For example, customer satisfaction is important throughout all phases, but most crucial in Transition.

Integrated Analysis Model

The PSM Integrated Analysis Model in Figure 4 shows the relationship between the information categories. The model helps you visually relate project information needs and issue areas in terms of cause and effect. Consider the following relationships:

- An increase in functional size (product size) could result in more personnel needed, which affects resources and cost.
- Unplanned staff turnover will likely impact schedule and progress.
- Process performance affects the need for personnel resources, influences schedules, and affects quality.
- Shortening schedules causes quality problems.
- Quality problems require rework, which increases the effort required to complete the project, and entails additional effort in future releases.
- Quality problems, of course, also affect customer satisfaction.

Each arrow represents a relationship. For example, as Product Size and Stability changes, so does project Resources and Cost. Problems in one area will affect other areas in the lifecycle, so it is important to monitor and control problems factors early on. This also allows you to identify root causes of problems that appear later on.

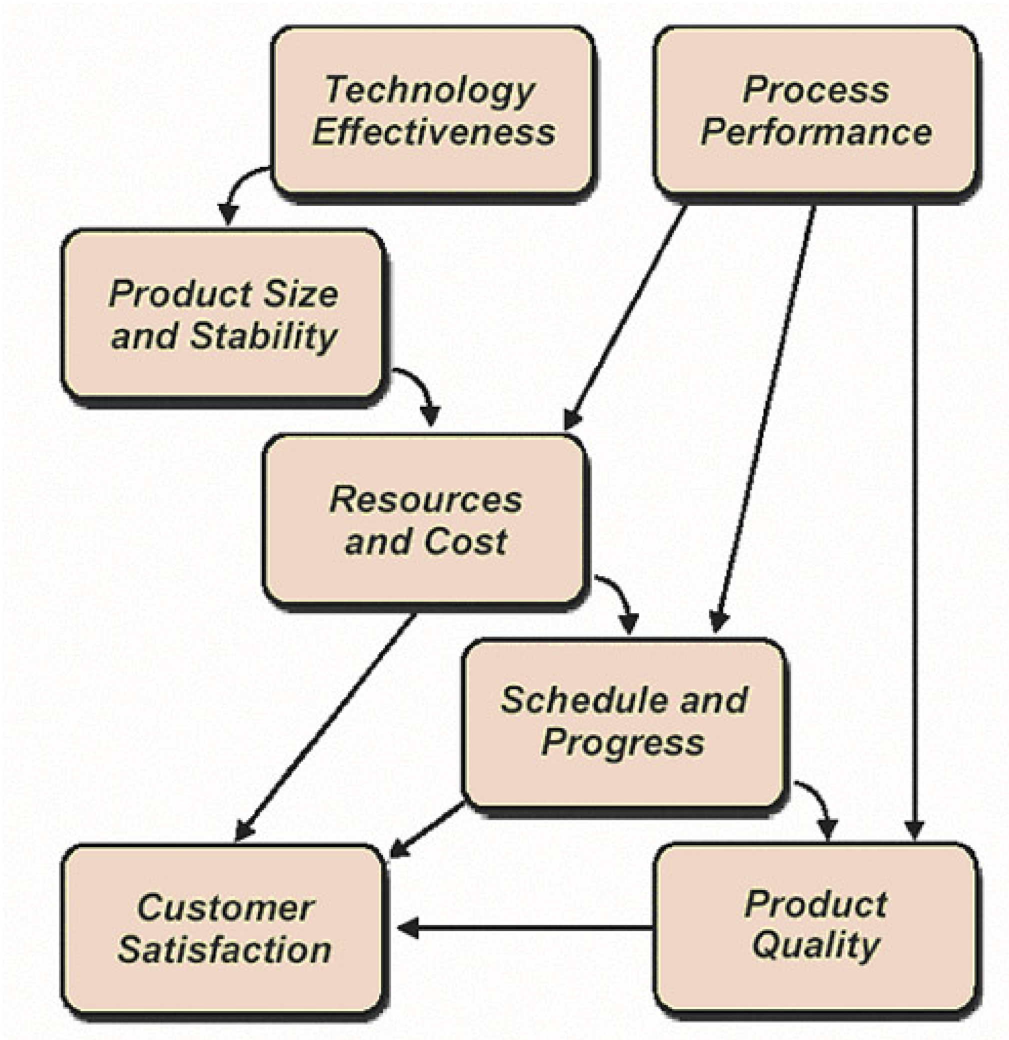


Figure 4: PSM Integrated Analysis Model

The key to controlling costs, meeting schedules, and achieving customer satisfaction is controlling problem factors early in the development lifecycle, which you can do by effectively measuring and monitoring the issues in each category.

Measurement Information Model

The measurement information model from PSM is a structure that links information needs to what can be measured. It describes how relevant measurable attributes are quantified and converted into indicators that satisfy an information need (Figure 5).

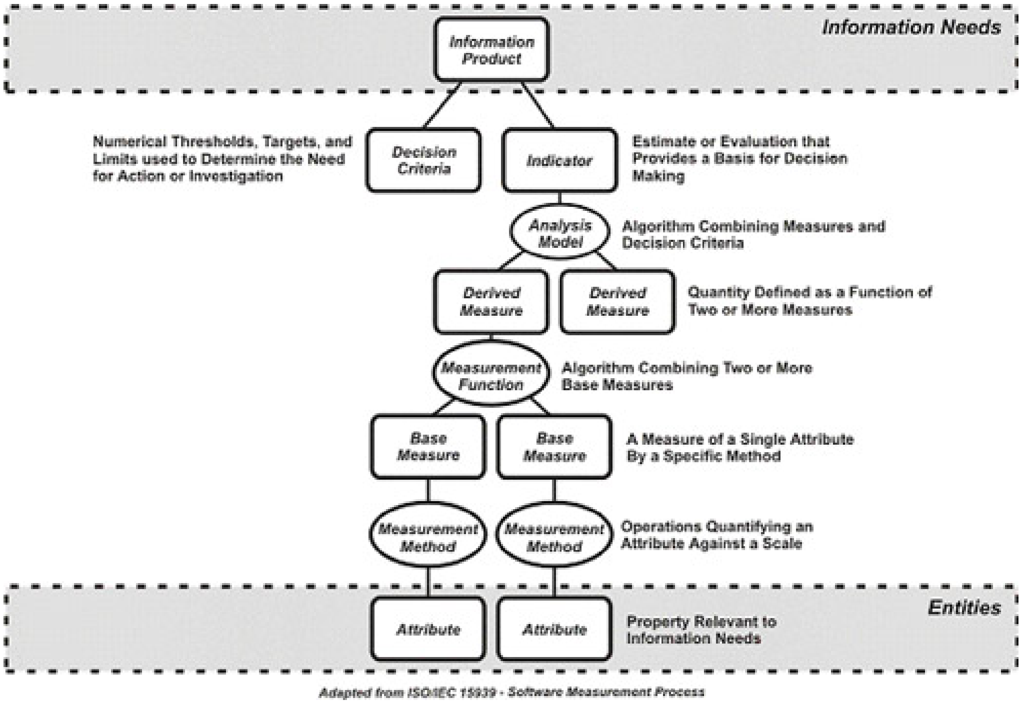


Figure 5: Measurement Information Model

The model formalizes measures into a construct that specifies exactly what will be measured and how the resultant data will be combined to produce results that satisfy a defined information need. The construct shows specific rules for assigning values, and defining the measurement methods, functions, and analysis models associated with each level of measure. From this, measurement constructs can be defined for each prospective measure. Using the model and defining measurement constructs provides many benefits, including:

- Reducing redundancy by helping to identify a core set of base measures that can serve many purposes.
- Increasing accuracy and repeatability by ensuring that all essential aspects of the measurement approach are adequately defined, and that everyone uses the same terminology.
- Maximizing the value of the base measures by creating patterns of derived measures and indicators that can easily be recognized, reused, and adapted.
- Documenting the link between the information need and how it is satisfied.

An example of a detailed measurement construct is shown in Figure 6.

Information Need	Evaluate release readiness
Information Category	Product quality
Measurable Concept	Functional correctness
Indicator	Status of severity 1 defects over time

[redacted]

Analysis Model	This measure is used to track project quality by tracking defects according to their severity. (High-severity defects make the product either unusable or hard to use.) The measure can be used to also track progress in resolving these defects. As the release or milestone approaches, the number of defects still open should decrease, and the number of defects closed should increase.
Decision Criteria	Postpone delivery until the number of open severity 1 defects is zero.
Base Measures	Number of severity 1 open defects
Measurement Method	For all severity 1 defects, cumulate the numbers that are currently open.
Attributes	These include the severity classification and status of defects.

Figure 6: Example Measurement Construct

A Sample Set of Measures

Every project and organization is unique and therefore has a unique set of information needs. Furthermore, as a project progresses from phase to phase, these information needs will change as requirements change, resources and technology change, and the discipline focus (design, development, test, etc.) changes.

PSM's information categories help you select appropriate measures by allocating each information need to one or more of the seven information categories. As Figure 7 shows, RUP breaks down the process into phases and emphasizes different disciplines within each phase.

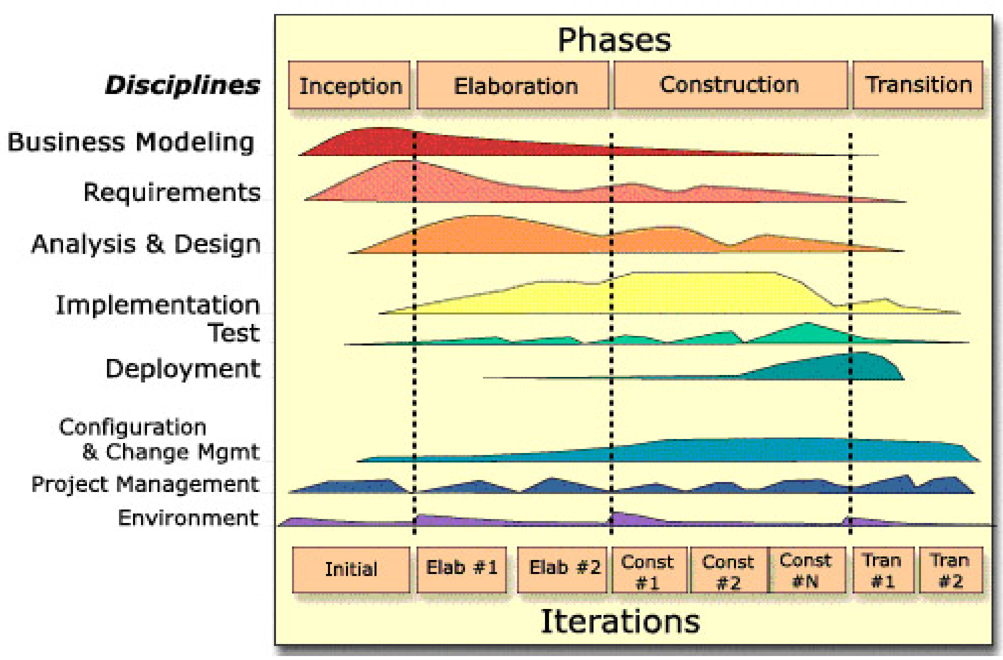


Figure 7: RUP Process Structure

Combining concepts from PSM and RUP results in the kind of mapping shown in Table 3 -- of a sample set of prospective measures in information categories and RUP phases. Many of these measures are further defined in PSM.

Table 3: Mapping Prospective Measures from PSM and RUP

RUP Phase	Information Categories	Measure
Inception	Schedule and Progress	Task Completion Requirements Status Business Use-Case Model Status Use-Case Model Status Design Model Status
	Resources and Cost	Staff Level, Turnover Earned Value - BCWS, BCWP, ACWP, SPI, CPI, SV, CV
	Process Performance	Requirements-Design Traceability Requirements-Test Case Traceability Model Elements (e.g., Activity Diagrams)
Elaboration	Schedule and Progress	Task Completion Requirements Status Requirements Tested Use-Case Model Status Design Model Status Units Designed, Coded, Tested Test Cases Attempted, Passed, Failed
	Resources and Cost	Staff Level, Turnover Earned Value - BCWS, BCWP, ACWP, SPI, CPI, SV, CV
	Product Size and Stability	Requirements Lines of Code Function Points Components Interfaces
	Product Quality	Defects
	Process Performance	Requirements/Model Traceability Requirements/Test Case Traceability

	Technology Effectiveness	Requirements Coverage
Construction	Schedule and Progress	Task Completion Change Requests Opened, Resolved Units Designed, Coded, Tested Test Cases Attempted, Passed, Failed
	Resources and Cost	Staff Level, Turnover Earned Value - BCWS, BCWP, ACWP, SPI, CPI, SV, CV
	Product Size and Stability	Requirements Lines of Code Function Points Components Interfaces
	Product Quality	Defects Age of Defects Cyclomatic Complexity Mean-Time-to-Failure
	Process Performance	Defects Contained Defects Escaping Scrap, Rework Effort Requirements/Model Traceability Requirements/Test Case Traceability Change Request/Test Case Traceability
	Technology Effectiveness	Requirements Coverage
	Customer Satisfaction	Customer Reported Defects
Transition	Schedule and Progress	Task Completion Change Requests Opened, Resolved Test Case Progress
	Resources and Cost	Earned Value - BCWS, BCWP, ACWP, SPI, CPI, SV, CV
	Product Quality	Defects
	Customer Satisfaction	Customer Reported Defects

Note that it is not necessary to include all measurements to receive benefit. In fact, it is possible to measure too much and have diminishing returns. Again, we recommend that you start with a small set of measures and build it up, based on information needs.

The same process and measures discussed in this article also apply to small projects. The difference is in the scale of effort that is spent on measurement activities. Small projects typically have fewer information needs, and therefore fewer measures. Plus, less data is typically available for those measures. For example, a small project may comprise only a few dozen activities that require schedule tracking, versus hundreds of activities that need tracking on a large project. The RUP Process Framework also enables development and selection of lighter processes, supplying examples to get people kick-started.

Summary

You will know you have implemented a successful measurement program in your organization when measurement is embedded in your process and becomes a way of doing business. Performance should improve, because your stakeholders will be able to make fact-based decisions. We have learned three main lessons from successful measurement programs:

1. Measurement is a consistent but flexible process that can and should be implemented incrementally. You cannot measure everything on day one of your project.
2. Decision makers must understand what is being measured.
3. Measurements must be used to be effective -- in other words, they must satisfy real information needs.

The measurement concepts described in this article are relatively easy to implement. PSM and RUP provide guidelines to help you get started with implementing a successful measurement program.

References

- For more information about PSM, see the PSM Web site.
- For more information about RUP, see the Rational Software Web site.
- Pamela Bowers, "The F/A-18 Advanced Weapons Lab Successfully Delivers a $120-Millon Software Block Upgrade." *STSC Crosstalk*, January 2002. http://www.stsc.hill.af.mil/crosstalk/2002/01/fa18.html
- Philippe Kruchten, *The Rational Unified Process, An Introduction*, Second Edition. Addison-Wesley, 2000.
- John McGarry, David Card, Cheryl Jones, Beth Layman, Elizabeth Clark, Joseph Dean, and Fred Hall, *Practical Software Measurement -- Objective Information for Decision Makers*. Addison-Wesley, 2002.
- Ron J. Perhrson, "Software Development for the Boeing 777." *STSC CrossTalk*, January 1996. http://www.stsc.hill.af.mil/crosstalk/frames.asp?uri=1996/01/Boein777.asp
- *ISO/IEC 15939 Software Measurement Process*. 2002.
- Rational Software, "Rational Unified Process 2002.05.20." Available at www.rational.com, 2001.

- Walker Royce, *Software Project Management: A Unified Framework*. Addison-Wesley, 1998.
- Software Engineering Institute, "Capability Maturity Model® Integrated (CMMI℠) for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing -- Version 1.1 Staged Representation." Carnegie Mellon University, March 2002.
- Marvin V. Zelkowitz and Ioana Rus,"Understanding IV&V in a Safety Critical and Complex Evolutionary Environment: The NASA Space Shuttle Program." 23rd International Conference on Software Engineering (ICSE '01). http://www.computer.org/proceedings/icse/1050/10500349abs.htm

---

Notes

[1] Walker Royce, *Software Project Management: A Unified Framework*. Addison-Wesley, 1998.

[2] *Ibid*.

[3] Rational Software, "Rational Unified Process 2002.05.20." Available at www.rational.com

[4] Ron J. Perhrson, "Software Development for the Boeing 777." *STSC CrossTalk*, January 1996. http://www.stsc.hill.af.mil/crosstalk/frames.asp?uri=1996/01/Boein777.asp

[5] Marvin V. Zelkowitz and Ioana Rus, "Understanding IV&V in a Safety Critical and Complex Evolutionary Environment: The NASA Space Shuttle Program." In *Proceedings of the 23rd International Conference on Software Engineering* (ICSE '01). http://www.computer.org/proceedings/icse/1050/10500349abs.htm

[6] For more information about RUP, refer to Philippe Kruchten's introductory book on RUP: *The Rational Unified Process, An Introduction*, Second Edition. Addison-Wesley, 2000.

[7] Rational Software, "Rational Unified Process 2002.05.20." Available at www.rational.com

[8] RUP Plug-Ins are designed and implemented using Rational Process Workbench.

Note: Portions of this article are reprinted, with the permission of the publisher, from *Practical Software Measurement: Objective Information for Decision Makers*, by John McGarry, et al., Addison-Wesley, 2002.

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