**Software Engineering Institute**

# Moving Up the CMMI Capability and Maturity Levels Using Simulation

David M. Raffo, PhD
Wayne Wakeland, PhD

**January 2008**

**Carnegie Mellon**

# Table of Contents

# List of Figures

## List of Tables

# Acknowledgements

# Executive Summary

Process Simulation Modeling (PSIM) can be used to evaluate issues related to process strategy, process improvement, technology and tool adoption, project management and control, and process design. It is a flexible tool and can aid in quantitatively testing ideas about how to configure a process or how to configure a software acquisition supply chain (consisting of contractors and sub-contractors) that would be too expensive or too risky to construct and evaluate in any other manner.

Recent developments in PSIM tools have drastically cut the costs of developing models to evaluate process issues. Moreover, new models and more systematic and repeatable methods have been developed for applying PSIM tools within organizations, enabling PSIM to provide greater business value. For example, new methods used to calibrate these models have reduced the amount of organizational and project-specific data required to achieve useful results.

Competition in the software industry and the continuing pressure from low-cost economies is pressing companies to improve their efficiency and to find ways to optimize their development and quality assurance activities, both locally and globally. Furthermore, as companies improve their operations and establish metrics in order to achieve higher levels of the CMM Integration[SM] (CMMI[®]) framework, the data collected can facilitate the construction of quantitative models.

As a result of these forces and trends, organizations regard PSIM as an attractive tool that can provide significant business value. This report presents the goals, motivations, and benefits associated with the application of PSIM within an organization. Many specific examples are provided to show some of the different ways that PSIM has been implemented within industry and government organizations to provide high value. Typically, process simulation more than pays for itself when it is used to evaluate even one decision.

Some of the tangible benefits that PSIM can provide include

- selection of the best possible development process, quality assurance strategy, or set of tools for a given project/situation/circumstance
- improved project planning through the use of an objective and quantitative basis for decision making
- enhanced project execution and control through PSIM's ability to quickly evaluate alternative responses to unplanned events
- a deeper understanding of the many factors that influence success for complex software development projects
- enhanced ability to communicate process choices and alternatives

---

[SM]    CMM Integration is a service mark of Carnegie Mellon University.

[®]    CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

- an objective way for project managers to answer difficult questions such as, "What inspection and testing approach will work the best for my project?" and "Will adding resources early in the project really reduce overall project cost?"

This report also describes in detail how PSIM supports each of the various CMMI Process Areas from level 2 through level 5. Some of the key areas of CMMI that PSIM directly addresses include

- Project Planning: defining project life cycles and identifying project risks; determining which criteria to use for deciding when to take corrective action
- Organizational Process Focus (OPF): identifying process improvements and developing implementation plans
- Risk Management (RM): identifying process risks, setting appropriate risk thresholds and assessing the risk associated with proposed process changes; determining when to perform risk mitigation
- Decision Analysis and Resolution (DAR) and Integrated Project Management (IPM): providing decision guidelines, processes, evaluation criteria, and alternative solutions and evaluating process alternatives and making specific recommendations
- Organizational Process Performance (OPP): selecting processes, establishing measures, setting specific performance objectives, and establishing baselines and performance models
- Organizational Innovation and Deployment (OID): evaluating, selecting, and deploying incremental and innovative improvements that can bring measurable gains to the organization's processes and technologies. Evaluating costs and benefits (and ROI) of process changes
- Causal Analysis and Resolution (CAR): identifying causes of process problems and evaluating alternative action plans to resolve them
- Quantitative Project Management: identifying suitable subprocesses that compose the project's defined process, based on historical stability and capability data, and composing the process with the appropriate verification and validation activities to maintain a project's quality

The time has clearly arrived for PSIM technology. It is a perfect fit for organizations that want to improve process planning, speed technology adoption, optimize process improvement, step up to quantitative project management, and move to the higher levels of CMMI.

Because this report is aimed at practitioners, especially software development project managers, process improvement champions and professionals, and also is likely to interest those researching software development processes, a considerable degree of detail and many examples are provided.

# Abstract

Process Simulation Modeling (PSIM) technology can be used to evaluate issues related to process strategy, process improvement, technology and tool adoption, project management and control, and process design. Recent developments in PSIM tools have drastically cut the costs to develop models for evaluating such issues, and new methods have been developed to apply PSIM, enabling it to provide greater business value. At the same time, trends within the software industry towards improving operations and reducing costs have heightened the need for tools to better plan and manage processes. As a result, organizations regard PSIM as an attractive tool that can provide business value today. This report shows examples of how PSIM has been implemented within industry and government organizations to improve process consistency and results. The report also shows, via many examples, exactly how PSIM supports Capability Maturity Model® Integration Process Areas from level 2 through level 5.

# 1 Introduction

The purpose of this report is to show how process simulation modeling (PSIM) can help companies improve their processes and achieve higher levels of process maturity and capability as called for by the Capability Maturity Model® Integration (CMMI®)[1] [SEI 2006]. CMMI was developed by a team consisting of members from industry, government, and the Software Engineering Institute (SEI).This report is aimed at practitioners, especially software development project managers, and researchers studying software development processes. The report describes a variety of PSIM applications and discusses how PSIM has helped organizations to improve their implementations of CMMI areas toward higher levels of process capability, maturity and performance.

CMMI is the leading industry standard for measuring software development processes. There are six capability levels (CLs), five maturity levels (MLs), and 22 process areas (PAs); within each PA are one or more specific goals (SGs) and multiple specific practices (SPs). Important aspects of CMMI include process documentation, process measurement, process repeatability, process predictability, and process consistency. An excellent overview of CMMI can be found at http://www.sei.cmu.edu/cmmi/general/index.html.

Software development organizations value CMMI and strive to move to higher CMMI MLs because implementing process improvements based on CMMI should be good for the business—better processes lead to improved business results.

There are many ways an organization might choose to fulfill or implement these PAs, SGs, and SPs. This report focuses on the role of PSIM in helping organizations to move to higher levels of process maturity and capability. The body of this report is organized into three major sections.

Section 2 describes PSIM and its potential benefits. It clarifies the types of processes that can be simulated, especially the ability of PSIM to handle processes that are complex and non-linear. An overview is provided to explain and contrast the different types of process simulation models (PSIMs) currently available. A summary table is provided that shows how PSIM can enable/fulfill or support various aspects of CMMI. This section concludes with an overview of the central and recent PSIM literature, including a discussion of recent developments that have made PSIM easier, faster, and less costly to apply.

Section 3 of the report describes nine example PSIM applications from the literature that illustrate how PSIM has helped software and systems development organizations improve their project and process management activities. These examples show how PSIM strongly supports a number of PAs, SGs, SPs, generic goals (GGs), and generic practices (GPs) of CMMI. The examples include using PSIM to

- architect, design, and document processes
- support bottom-up cost estimation
- compose processes, improve process planning and tradeoff decisions
- analyze and evaluate process improvement opportunities quantitatively

---

- assess the costs and benefits of applying new tools and technologies on a project
- support quantitative process management and control
- optimize the process and quality assurance strategy for a project
- support training
- study global software development

Each example is structured as follows:

- introduction/overview
- description of the model
- summary explaining how the model was applied and the results achieved
- discussion of how this particular PSIM application fulfilled and/or supported CMMI

This section closes with a summary of how these applications illustrate the role that PSIM can play in helping organizations to improve their process maturity and capability.

Section 4 describes how PSIM supports the CMMI PAs from levels 2 through 5. First, one of the following ratings is given to indicate the degree to which use of PSIM might enhance each PA and SG: Strongly supports, Supports, or Partially supports. For the PAs rated as 'strongly supported' we provide detailed tables and commentary describing how PSIM supports or strongly supports each CMMI ML, PA, SG, and SP. The commentary specifically indicates how PSIM might be used support or fulfill each SP.

The appendices present additional details for comparing different simulation modeling paradigms, providing selection criteria for choosing simulation tools, identifying key components of a discrete event simulation (DES) model, and providing a detailed review of the relevant literature.

The main message of this report is that PSIM is a technology that has matured to the point where it can provide value to organizations at all levels of CMMI from levels 2 through 5. Not only can PSIM help support and fulfill a number of important PAs, SGs, and SPs, but it also has been applied in industry and government to provide tangible business value.

# 2  The Benefits of Process Simulation Modeling

PSIM can play a key role in supporting the overall CMMI framework. In this section, we define PSIM and describe some of the potential benefits that an organization might realize by applying PSIM to their software and systems development operations. We also provide an overview of PSIM methodologies, discuss historical disadvantages of PSIM, and describe how recent advances have ameliorated these disadvantages. The final section summarizes how PSIM can support CMMI implementation.

## 2.1  WHAT IS SIMULATION? WHAT IS PROCESS SIMULATION MODELING?

Simulation models have been used for years in many industries such as automotive, chemical, manufacturing, and information technology to predict the performance of complex systems.

Simulation models are computerized representations of systems that are designed to display significant dynamic features of the systems they represent. These models often focus on reproducing the system's behavior over time or replicating the system's *performance* in terms of specific measures of interest. From a manager's perspective, simulation is a tool that supports decision making, forecasting, and the analysis of tradeoffs and "what-if scenarios."

Simulation models are often employed in the types of situations where

- behavior over time is of particular interest or significance, and this behavior is difficult to anticipate due to the influence of unexpected feedback. In this situation the consequences of an event or action trigger a sequence of responses that loops back and either reinforces or offsets the original event or action. (Software development projects are often characterized by a complex web of these feedback loops.)
- system performance is less effective than desired and difficult to estimate due to the high degree of uncertainty or randomness present in the system
- alternative configurations of the system are being considered, but piloting or implementing a new configuration is likely to involve considerable risk or cost. Configuration refers to both product configuration and development process configuration.
- live experimentation is impractical due to the economic and/or logistical cost of manipulating the actual system

Simulation models are often used to

- improve understanding of the system being modeled
- provide a basis for experimentation
- estimate system performance
- answer what-if questions
- determine the likely impact of parametric and structural changes

A PSIM is a specific type of simulation model that focuses on replicating the behavior and performance of organizational processes, such as insurance claim processes, expense reimbursement processes, new product development processes, and physical processes such as loading and unloading an airplane.

In the CMMI context, PSIMs focus on the dynamics of software and systems development, acquisition, and maintenance. They can be used to represent the process as currently implemented (the as-is, as-practiced, as-documented process) or as planned for future implementation (the to-be process) based on process improvement activities, applying new tools and technologies, selecting alternative processes, and so forth. PSIMs have been used to help companies achieve industry certification and to support process improvement programs such as CMMI, Six Sigma, and ISO 9000. They have also been used to help management select and define process measures and to analyze and redesign corporate process performance measurement programs.

## 2.2  POTENTIAL BENEFITS OF PSIM

Potential benefits of PSIM include reduced risk, reduced effort and cost for evaluating new technologies, better (more reliable, faster) decision making, identification of improved or optimal approaches and processes, and enhanced communication.

Specific benefits of PSIM with regard to product development include

- **selection of the best possible development process** for a given situation/circumstance
- **improved project planning** through the use of an objective and quantitative basis for decision making
- **enhanced project execution** (control and operational management) because alternative responses to unplanned events can be quickly evaluated using PSIM before decisions are made
- **a means to answer burning questions being asked by project managers**, such as
    - What is the impact on project performance of increasing or decreasing testing, inspections, or both? What is the risk? What is the return on investment (ROI)?
    - How might changes to my development processes (such as the requirements process or the critical design process) impact performance? What is the risk? What is the ROI?
    - What development phases/steps are essential to success?
    - Which phases/steps could be skipped or minimized to shorten cycle time and reduce costs without sacrificing quality? What is the risk?
    - Are inspections worthwhile on my project? At what point(s) in the project do inspections provide the most value? What if we apply more rigorous quality assurance (in CMMI terminology, verification) only to critical product components? What is the impact? What is the risk? What is the ROI?
    - What is the value of applying automated tools to support development and testing activities? How well does the technology need to perform for it to be worthwhile?
    - In general, what are the likely benefits (and costs) of implementing a particular process change? What is the risk associated with making a particular change? What is the ROI?
    - How do I objectively compare and prioritize process changes?
    - What specific process changes would help me to achieve higher levels of the CMMI standard? Do they provide tangible business value?
- **improved understanding of the many factors that influence project success** for complex software development projects
- **enhanced ability to communicate process choices and alternatives** through PSIM's making "intangible" processes more visible and concrete
- **better training and learning** for project managers, project team members, and executive leadership

- **elevation of project management to a more strategic level by** allowing projects to be analyzed over their full life cycles and with respect to multiple measures of success

Figure 1 illustrates the central role that PSIM can play when evaluating process alternatives.
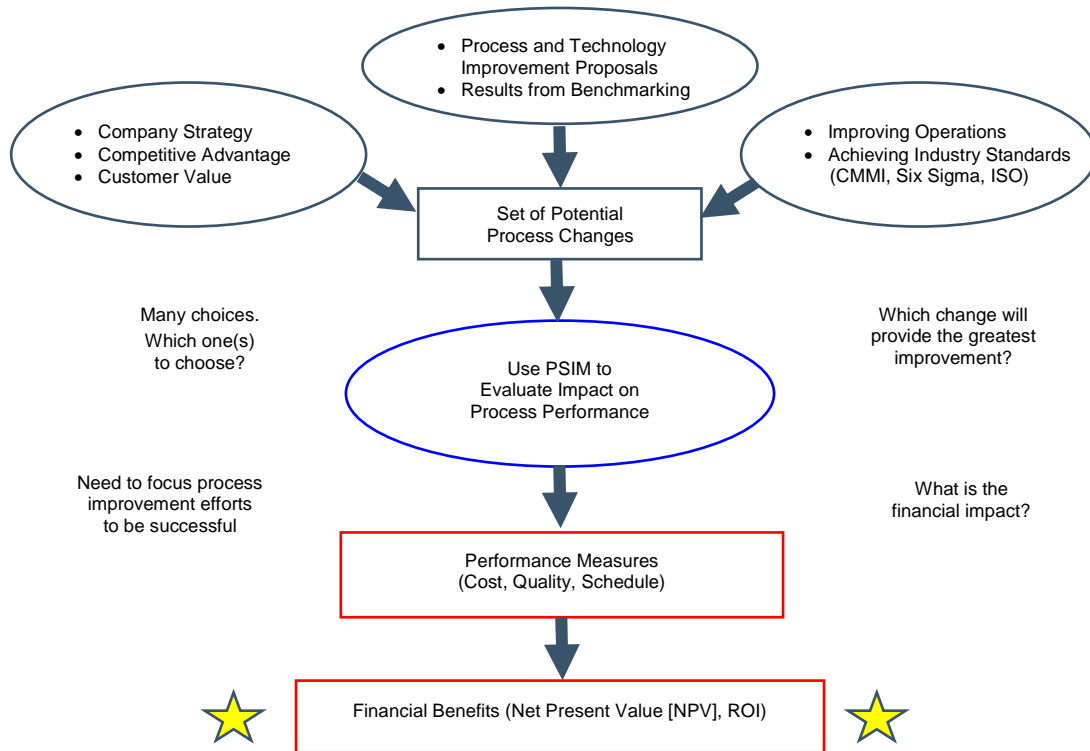


*Figure 1:      PSIM Plays a Central Role in Process Management*

To provide a specific example, Figure 2 shows the results of using PSIM to evaluate three potential process improvements: 1) implementing quality function deployment (QFD) to improve requirements elicitation, 2) implementing a formal *voice of the customer* program, and 3) adding the QuARS requirements checking tool [Ferguson 2006, Lami 2005b]. A fourth case of eliminating inspections entirely is also shown.

| | Option | Effort (PM) f | Project Duration (Months) | ▲Revenue due to ▲Duration | Total Injected Defects | Corrected Defects | Escaped Defects | Implementation Costs | NPV ($) | ROI (%) |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Base Case | 720 | 37.3 | $0 | 5,156 | 4,901 | 255 | $0 | n/a | n/a |
| 1 | Implement QFD | 722 | 36.3 | $12 K | 5,001 | 4,750 | 251 | $100 K | -$130 K | n/a |
| 2 | Implement VOC | 717 | 36.1 | $204 K | 5,053 | 4,801 | 252 | $120 K | $39 K | 23% |
| 3 | Add QuARS Tool | 708 | 35.1 | $368 K | 5,156 | 4,907 | 249 | $80 K | $285 K | 74% |
| 4 | No Inspection at Requirements phase | 741 | 37.7 | -$246 K | 5,156 | 4,890 | 266 | $0 | -$413 K | n/a |

*Figure 2:*      *Sample Results From Using PSIM to Evaluate Alternative Process Tools, Including NPV and ROI*

In this example, Option 3 (adding the QuARS Tool) provided the best improvement in performance and resulting financial benefit (ROI and NPV [i.e., profit]), allowing the project manager to move forward confidently with that decision.

## 2.3 OVERVIEW OF PSIM METHODOLOGIES

### 2.3.1 Overview of Various Methodologies

The goal of this section is to provide an overview of various PSIM methodologies (paradigms) that have been applied to the systems and software development domain. These are

- discrete event simulation (DES)
- system dynamics (SD)
- hybrid simulation (HS)
- state-based simulation (SBS)
- agent-based simulation (ABS)
- knowledge-based systems (KBS)
- qualitative simulation (QS)

The most commonly used methods are DES and SD. A third method that is gaining in interest is HS, which combines DES and SD into one model to achieve the benefits of both methods.

The following paragraphs provide a brief description of each of the listed simulation methods. A more complete overview of several of these methods can be found in Appendix A. A key point to note is that process simulations are quantitative. Process simulation uses a graphical depiction of process workflow as the basis for adding quantitative metrics and creating models to predict process performance. A user can choose from a number of qualitative methods and tools to graphically model the workflow of a process.

**Discrete event simulation (DES)**: DES models are discrete and dynamic. They can use both deterministic and stochastic (random) parameters enabling Monte Carlo simulation. DES models are represented through a network of activities and discrete (individual) items that flow through this network. Activities transform items and update the system state. DES models typically use a large number of different model elements and sometimes rely on programming. DES tools typically have strong graphical capabilities and allow a great deal of flexibility in the form of equations and random variables that can be used. DES is useful for problems that involve fine-grained, detailed processes and many simulated artifacts because the state only requires updating when events occur. DES models have an exceptional capability to model the richness of processes and workflow, work packages (e.g., attributes), and resources, which allows for quantitative assessment of process performance [Mueller 2006, Neu 2003, Raffo 2003, 2005a, 2005b, 2005c].

**System dynamics (SD)**: SD modeling is a type of continuous system simulation [Law 2000]. It is also dynamic and uses a set of differential equations to calculate the behavior of a system or process over time. By systematically varying parameters, users can gain a sense for the variability in outcomes, but SD is not inherently stochastic in nature. In contrast to DES models, it typically represents processes at a higher level (less detail). A strength of SD models is their ability to model feedback loops, but SD models do not usually incorporate process stochastics [Forrester 1961, Abdel-Hamid 1991, Madachy 1994, Rus 1998]. SD tools typically provide good graphics capabilities for representing feedback loops. Moreover, recent tools provide rudimentary capability to incorporate some stochastic random variables.

**Hybrid simulation (HS)**: HS models combine two or more simulation techniques. In software process simulation, HS typically refers to simulation models that combine SD with DES. Hybrid models are not widely used due to limited tool support and the increased complexity involved. However, some commercial tools such as Extend by ImagineThat, Inc. (http://www.imaginethatinc.com) do enable combined SD and DES models to be created within the tool. As a result, HS models are both dynamic and stochastic, and are able to reflect feedback loops, and the fine-grained richness of processes, work products, and resources. Applications include project planning and multi-site development [Donzelli 2001, Martin 2002, Lakey 2003, Setamanit 2006, Mizell 2006].

**State-based simulation (SBS)**: SBS models are another type of dynamic simulation. SBS models favor the representation of a system through finite automata. This technique uses activity and state charts to model the behavior of processes. SBS models can represent the richness of processes but not the same richness of work products, resources, and feedback loops as can be done with DES or SD models. At the same time, SBS models do represent the flow of control of work very precisely. Specifically, SBS models are excellent at representing parallel activities and concurrency and can be stochastic. SBS tools typically have exceptional graphic capabilities and enable both static and dynamic model checking. SBS models are analyzable, multi-level, and incorporate multiple perspectives. Currently, they are not widely used for process simulation.

**Agent-based simulation (ABS)**: ABS models represent systems and processes as populations of autonomous interacting parallel "agents." Each agent runs a simple set of rules that describe how it interacts with other agents and its environment. Rules can be deterministic or stochastic. The emphasis is on generating emergent phenomena from the bottom up without having to specify the "equations of motion" for the system (as would be the case for SD) or a specific process flowchart for how the entities move through the system (as would be the case for DES). ABS models are very fine grained, and can represent the detailed interactions between individual entities (e.g., work packages and resources). ABS is just starting to be applied to systems and software development.

**Knowledge-based or rule-based systems (KBS or RBS)**: RBSs are a type of artificial intelligence (AI) system or "expert system" that is referred to as "top down" (as opposed to "bottom up" systems such as artificial neural networks). These systems employ a rules repository in conjunction with an inference engine/mechanism to represent expert knowledge and mimic expert reasoning processes. In contrast to the previous five techniques, RBSs rely more on textual/logical descriptions. They have been primarily used for process understanding and educational purposes. RBS models represent a "person in the loop" and, as a result, can reflect fine-grained process enactment [Drappa 2000, Scacchi 1999, Scacchi 1990].

**Qualitative simulation (QS)**: QS has been used to model software evolutionary processes and trends. QS models operate on the assumption that information about software and systems development processes is fuzzy. QS models accommodate this by representing important model parameters as equations that simply increase, decrease, or stay the same over time. The approximate magnitude of the increase or decrease is also modeled. Using this information allows general trends and emergent behavior of the system to be seen [Ramil 2003, Smith 2005].

Overall, each approach has different strengths and limitations. The selection of the type of modeling paradigm to be used will depend upon the scope of the system that is being modeled and the types of questions that are to be addressed.

### 2.3.2    Simulation Tools for DES, SD, and HS Methodologies

DES tools include

- Arena (http://www.arenasimulation.com)
- Extend (http://www.imaginethatinc.com)
- iGrafx (http://www.imaginethatinc.com)
- Micro Saint (http://www.adeptscience.co.uk/products/mathsim/microsaint)
- Process Analysis Tradeoff Tool (PATT) (based on Extend) (http://www.quantelsystems.com)
- ProModel (http://www.promodel.com)
- Witness (http://www.lanner.com)

SD tools include

- iThink/STELLA (http://www.iseesystems.com)
- Vensim (http://www.vensim.com)
- Powersim (http://www.powersim.com)

Of these tools, only Extend can be easily used to support HS, although most of them provide at least some support for both discrete and continuous logic. A newer tool called AnyLogic (http://www.xjtek.com) supports DES, SD, HS, and ABS, but has not yet been applied to systems and software development. Appendix A offers a list of criteria that one might use to select a simulation tool for a particular context.

### 2.3.3    Comparison of the Two Most Commonly Used Process Simulation Modeling Techniques

In applications of Process Simulation Modeling, SD, and DES have been by far the most frequently used techniques. Table 1 compares these two important techniques in more detail.

*Table 1:       Characteristics of SD and DES*

|  | SD | DES |
|---|---|---|
| **Typical purpose** | Investigation of strategies: policy making, gaining understanding | Investigation of decisions: optimization, prediction, and comparison |
| **Model elements/ symbols** | Few basic elements: flows, stocks, connectors | Many basic elements: items, activities, queues, attributes, resources, routing, generators |
| **Organizational scope** | Rather strategic; systems with wide boundaries | Rather operational and tactical; typically narrower range |
| **Amount of detail represented** | Low; information highly aggregated | High; many entities; considerable process detail |
| **Work breakdown structure** | Partial support at best, due to aggregated information | Activities and artifacts represented as distinct entities |
| **Tool support** | Good | Good |
| **Advantages** | Continuous feedback can be modeled; simulation equations easily readable; few model elements; requires less data than DES models | Can represent specific entities/items; process easy to understand; easy to modularize; tools allow animation and visualization of process flow |
| **Disadvantages** | Difficult to represent unique items; process description less clear than in DES models | Internal computation not transparent; many model elements to learn |

DES continues to serve as the primary tool for studying detailed processes as typified by software development. Additional details regarding the key components of DES models are provided in Appendix B.

## 2.4  HISTORICAL DISADVANTAGES OF PROCESS SIMULATION

Although there are many advantages and benefits associated with PSIMs, historically, there have been a number of potential costs, challenges, and risks.

### 2.4.1    Costs

The main costs associated with PSIMs are the costs to design and develop an initial model, collect model data, and utilize and maintain the model. Designing and developing PSIMs requires effort; one must understand the processes being modeled, collect the data, and build the model using an appropriate simulation tool. This often requires specialized knowledge and skills. Costs of data collection can include costs associated with obtaining process metric data and defining model parameters. Sometimes organizations do not collect the data required for the model. This data then needs to be specifically collected or be estimated. Costs associated with using and maintaining the models include the costs of adapting the model to address new projects and answering new questions asked by management.

Recent developments have helped to alleviate these costs. Section 2.5 reviews these developments and their favorable impact on cost.

### 2.4.2.   Challenges

The main challenges associated with applying PSIMs within an organization are data collection and model complexity. Organizations that have achieved CMMI levels 3, 4, and 5 typically collect sufficient data to support PSIMs. Moreover, recent developments have reduced the data required to obtain useful results from PSIMs.

PSIMs are typically more detailed and therefore more complicated than COCOMO [Boehm 2000, 1981] or spreadsheet models. Their usage requires staff members to learn new tools and to understand the basic concepts of process modeling and simulation. These skills are not taught in typical computer science curriculums, so specific training programs are required. In addition, PSIMs based on the discrete event paradigm provide outputs that are probabilistic rather than point estimates. Understanding and interpreting this output requires additional training. However, training for process improvement, such as that for CMMI and Six Sigma, provides the knowledge necessary to understand and interpret simulation output [Pyzdek 2003, Stamatis 2001]. Moreover, the equations typically utilized in simulation models are less complex than those used to estimate reliability. Simulation models are graphical, compartmentalized by process steps, and decomposable. As a result, once someone has learned the simulation tool, he or she is typically able to understand the models with relative ease. Basic training for most simulation tools is available from the tool providers.

### 2.4.3.  Risks

It is possible for any model or tool to be misapplied or misused and for results to be misinterpreted. PSIMs are no different. Poor data as well as lack of user knowledge and skill can have undesired effects. Recent developments in process modeling and simulation methodologies have created robust tools that provide easy tailoring options, user-friendly interfaces for creating equations, and linkages to databases of industry standard data. However, like all models, the user can modify them in ways that can lead to unintended effects.

## 2.5  RECENT PROGRESS IN PSIM SOFTWARE AND MODELS

Recent developments in the field have significantly reduced **the costs associated with applying PSIMs** within an organization on multiple dimensions—model development, data collection, model use, and model maintenance. New tools have been created that substantially reduce effort required to build PSIMs from weeks to days. These advances are based upon the concepts of design patterns and problem solution templates. The net effect is that reusable models and process building blocks have been created that drastically reduce the time to create PSIMs. For reference, Appendix C provides an overview of the central PSIM literature.

**The costs associated with data collection** have been reduced by (1) developing new methods for model tailoring that require less organizational data, (2) utilizing pre-built industry standard or organizational models as a starting point and then incrementally updating the models with detailed, project-specific data as it becomes available, and (3) recognizing that using approximate data is suitable for some main applications of PSIMs (such as selecting the best process improvement among a set of alternatives).

**The costs associated with using and maintaining PSIMs** have also been significantly reduced by new tools that have (1) generic process building blocks and model structures that enable extensive reuse of models, (2) manager-friendly interfaces that enable process tailoring options via pull-down menus, and (3) methods and training available to assist in the interpretation and use of simulation output.

In recent years, a great deal of work has been done to refine the methods, develop new models, and apply PSIM to address new problems in different development contexts. The cost to purchase simulation software is reduced, the field has become more mature, less time is required to build and validate models, and there are more sample applications.

The reduction in time is due primarily to the development of modeling approaches that feature generic, readily configurable model building blocks, process components (e.g., alternative processes for the requirements phase), or templates of common software development life cycles (e.g., IEEE 12207, spiral, incremental, product-line engineering, agile). Moreover, the associated analytical methods enable users to quickly and effectively draw useful inferences from the models. The latest generation of PSIM tools

- **are based on extensive research into software process modeling** conducted in academia, the SEI, and industry.
- **provide user-friendly graphical interfaces**
- **model software processes at a detailed level**, such as the level of individual requirements or function points, in addition to higher level aspects, such as project resource levels and productivity
- **incorporate SEI methods**, **such as process repositories,** to define processes and support specific CMMI PAs, as well as templates for defining process steps
- **integrate metrics** related to cost, quality, and schedule to create an easily understood picture of project performance
- **predict the project-level impacts** in terms of cost, quality, and cycle time of process improvements
- support business case analysis of process decisions such as ROI, NPV, and quantitative measures of risk
- **reduce the risk** associated with process changes by predicting the likelihood that the changes will result in improved results
- **require less time and effort** than methods that require the user to build specific models from general-purpose building blocks
- **may also reduce the degree of expertise needed** to apply PSIM

Taken together, these recent developments and new PSIM tools have dramatically reduced the costs associated with applying PSIMs within organizations, and, therefore, make it increasingly worthwhile for practitioners to procure and apply PSIM. Moreover, organizations can often recoup their entire investment by using PSIMs to support even one decision.

## 2.6 PSIM AND CMMI

There are many ways that PSIM can assist users to increase process maturity and capability. At each CMMI level, PSIM helps to enable, fulfill or strongly support a number of key process areas, SGs, and SPs. Table 2 lists 50 specific practices that are strongly supported by PSIM.

Further details regarding how PSIM supports CMMI are provided in Section 3, where we describe specific scenarios/cases that illustrate how PSIM is (or can be) utilized by organizations to support, enable, or fulfill CMMI. Further, in Section 4, we methodically consider each PA, level by level, by each SG and each SP, suggesting possible roles that PSIM might play, with additional specifics provided for clarification.

*Table 2:    Specific CMMI Practices*

| ML | Process Area | SG | SP |
|---|---|---|---|
| 2 | Project Planning | • Establish estimates | • Define project life cycle<br>• Determine estimates of effort and cost |
| | Measurement and Analysis | • Align measurement and analysis activities | • Establish measurement objectives<br>• Specify measures<br>• Specify analysis procedures |
| | | • Provide measurement results | • Analyze measurement data<br>• Store data and results<br>• Communicate results |
| 3 | Organizational Process Focus | • Determine process-improvement opportunities | • Establish organizational process needs<br>• Appraise the organization's processes<br>• Identify the organization's process improvements |
| | | • Plan and implement process-improvement activities | • Establish process action plans<br>• Incorporate process-related experiences into the organizational assets |
| | Organizational Process Definition | • Establish organizational process assets | • Establish standard processes<br>• Establish life-cycle model descriptions<br>• Establish tailoring criteria and guidelines<br>• Establish the organization's measurement repository<br>• Establish the organization's process asset library |
| | Integrated Project Management | • Use the project's defined process | • Establish the project's defined process<br>• Use organizational process assets for planning project activities<br>• Manage the project using the integrated plans<br>• Contribute to the organizational process assets |
| | Risk Management | • Prepare for risk management | • Establish a risk-management strategy |
| | | • Identify and analyze risks | • Evaluate, categorize, and prioritize risks |
| | | • Mitigate risks | • Develop risk mitigation plans |
| | Decision Analysis and Resolution | • Evaluate alternatives | • Establish evaluation criteria<br>• Identify alternative solutions<br>• Select evaluation methods<br>• Evaluate alternatives<br>• Select solutions |
| 4 | Organizational Process Performance | • Establish performance base-lines and models | • Establish process performance measures<br>• Establish process performance baselines<br>• Establish process performance models |
| | Quantitative Project Management | • Quantitatively manage the project | • Establish the project's objectives<br>• Compose the defined process<br>• Select the subprocesses that will be statistically managed<br>• Manage project performance |
| | | • Statistically manage subprocess performance | • Select measures and analytic techniques<br>• Apply statistical methods to understand variation<br>• Monitor performance of selected subprocesses |

| 5 | Operational Innovation and Deployment | • Select improvements | • Collect and analyze improvement proposals<br>• Identify and analyze innovations<br>• Select improvements for deployment |
|---|---|---|---|
| | Causal Analysis and Resolution | • Determine causes of defects<br>• Address causes of defects | • Analyze causes<br>• Evaluate the effect of changes |

## 2.7 SUMMARY

The purpose of Section 2 was to describe PSIM and to suggest the types of benefits that an organization applying PSIM to its development operations might reasonably expect to achieve. PSIM enables an organization to architect, compose, and improve its processes at a detailed level. The models are quantitative and can enable bottom-up cost estimation for projects when utilizing relevant project data. This can provide a very useful sanity check vis-a-vis top-down estimation tools such as COCOMO.

Primary benefits of using PSIM include (1) selection of the best possible development process for specific situations and circumstances, (2) improved project planning and execution, (3) provision for an objective and quantitative basis for project decisions, (4) reduced risk when implementing process changes, and (5) enhanced understanding of possible outcomes in complex processes and projects.

In Section 3, we show specifically *how* PSIM can be applied by identifying nine application areas and showing specific examples of application in both government and industrial organizations.

# 3 Process Simulation Applications that Provide High Value for Software and Systems Development Organizations

The goal of this section is to present examples or use cases demonstrating how PSIM can be applied within an organization and the benefits that result. These high-value process simulation use cases strongly support a number of process areas, SGs, SPs, and GGs and GPs of CMMI. The use cases shown are drawn from the literature and summarize research and case studies conducted at commercial and government organizations. The goal is to show how PSIM provides significant business value to organizations by improving their project- and process-management activities.

The use cases are presented in the order that an organization would typically use PSIM rather than in order of importance or impact. For instance, typically an organization would first use a process life-cycle template or create a process model using graphical process blocks. This is discussed in Section 3.1, which deals with how to use PSIM to architect, compose, and document processes. Next, an organization would tune the model and use it to make predictions. This is discussed in Section 3.2. After making basic predictions using the model, an organization would then use it to address important issues for process planning and to support tradeoff decisions (discussed in Section 3.3), evaluating process improvement opportunities (discussed in Section 3.4), evaluating the impact of new tools (discussed in Section 3.5) and so forth. The use cases build upon each other.

In terms of impact and benefit, using PSIM to optimize development processes and quality assurance strategies [for both verification and validation (V&V) and also independent verification and validation (IV&V)] is a very high-impact activity (discussed in Section 3.7). Using PSIM to evaluate the impact of new tools and address process tradeoffs (discussed in Sections 3.4 and 3.5) are very high impact as well. Section 3.9 discusses globally distributed development and how PSIMs are being used to evaluate software supply chains. Managing processes quantitatively (Section 3.6) shows the use of simulation to track the project and to re-plan the project when performance falters. Each of these areas can provide very high value to an organization.

Each section is self contained to make it easy for readers to focus on applications that are relevant to their needs. The use cases are organized in the order below, according to their use of process simulation to

1. architect, compose, and document processes
2. estimate project costs from the bottom up
3. improve process planning and tradeoff decisions[2]
4. analyze and evaluate process improvement opportunities quantitatively
5. assess costs and benefits of applying new tools and technologies on a project
6. manage and control projects quantitatively
7. optimize development processes
8. train project staff
9. evaluate strategic issues

---

[2]    As stated in Sections 1 and 2, this report focuses on the use of simulation to model software and systems development **processes**. The technique of simulation itself is broadly applicable to many domains including **product** development.

Each use case section includes

- an introduction/overview
- a brief description of the model (with screenshots)
- model application/results
- a summary of how this sample PSIM supports CMMI processes and practices

For some of the use case applications, other tools exist that also could provide the capability described in a specific use case, but we do not consider these process simulation tools. We make the distinction between process modeling and process simulation in that process simulation can not only graphically represent the process and store textual information about process definition elements (e.g., entry and exit criteria, tasks, roles, tools) but also quantitatively simulate and analyze process performance. Based on this distinction, this section will not discuss process modeling tools such as Lil Jil (http://laser.cs.umass.edu/tools/littlejil.shtml), PML, BPML, UML, and Petri Nets or business process modeling tools that are not tailored to the software and systems development domain (e.g., business process modeling notation).

The advantage of creating a PSIM is that it has quantitative capabilities that enable the applications described in the use cases. The first seven case studies described in this section are based on either of two specific life-cycle models. These are (1) a model of the IEEE 12207 Software Development Life Cycle Process and (2) a model of an incremental waterfall development process (referred to as the Incremental Model). Both models were created in the Process Analysis Tradeoff Tool (PATT) offered by Quantel Systems, Inc. (www.quantelsystems.com). This tool is based on the Extend simulation platform offered by ImagineThat, Inc. (www.imaginethatinc.com). We have chosen this approach to enhance the key points of PSIM and to simplify understanding, as changing from one modeling tool to another and from one process model to another would be confusing. However, Sections 3.8 and 3.9 describe models that are not based on PATT.

## OVERVIEW OF THE MODELS USED FOR CASE STUDIES 3.1 TO 3.7

PSIMs of the IEEE 12207 process life cycle and the incremental development life cycle were created using PATT. These models predict process performance in terms of development cost, product quality, and project schedule for the overall project and for individual process steps, if desired.

Some of the inputs to the simulation models include

- productivity rates for various processes
- volume of work (i.e., thousand lines of new code [KSLOC])
- defect detection and injection rates for all phases
- effort allocation percentages across all phases of the project
- rework costs across all phases
- parameters for schedule overlap (e.g., starting the coding phase before fully completing the design phase)
- amount/effect of training provided
- resource constraints

Actual project- and organization-specific data were used for parameters of both models where possible and were augmented with data from the literature or through interviews of project and process experts where necessary. Models were developed from these data using multiple regression to predict defect rates and task

effort. These distributions and models were then integrated to predict the three main performance measures of cost, quality, and schedule at each process step. In both models, results were summed as appropriate to develop overall project performance measures indicated by the following model equations:

- total effort = $\Sigma\Sigma$ effort$_{ij}$ for all i and j
- effort$_i$ = $\Sigma$ f (productivity$_i$, effort_allocation$_i$, size$_j$, defects$_j$) for all j
- duration$_i$ = last_finish$_{ij}$ – earliest_start$_{ij}$ for all j
- corrected_defects$_{ij}$ = (escaped$_{i-1,j}$ + injected$_{i,i}$)* corr_rate$_{i,j}$

Where:

*Total effort* is the sum of effort expended over all process steps i for all work products j. It includes the effort required to rework latent defects remaining at the end of the project (i.e., field defects).

*Effort$_i$* is the effort for an individual process step i summed over all work products j. Effort$_i$ is a function of productivity$_i$, effort allocation$_i$, work product size$_j$, number of defects j detected or corrected (if an inspection/test or rework step respectively) summed over all j.

*Duration$_i$* is the calendar time that passes while executing process step i. It is the difference between the latest finish time of all work products flowing through process step i less the earliest start time of all work products flowing through process step i.

*Corrected_defects$_{ij}$* is the number of defects that have been corrected in work product j in process step i. The number of corrected defects is the sum of the escaped defects from the previous process step i-1 plus the number of defects that are injected during process step i multiplied by the correction rate (corr_rate) for process step i for work product j.

Productivity, defect injection rates, and defect correction rates are stochastic, with probability distributions derived from actual project data.

The **IEEE 12207 Model** is a full life-cycle model containing 86 process steps modeled using three layers of hierarchy [IEEE 1998]. At the top level are the major life-cycle phases: requirements, design, code, test, and deployment. Lower levels contain process steps within these life-cycle phases. Industry data from the work of Jones has been tailored to large-scale NASA projects in conjunction with data from multiple NASA centers [Jones 1991, 2000, 2007]. To use the model, NASA project managers adjust top-level parameters for productivity, project size, and schedule. Other parameters such as defect injection and detection rates may also be tuned to specific projects if desired, depending upon how the model will be used.

As depicted in Figure 3, this model contains a separate layer to accommodate IV&V of NASA projects. On NASA projects, different subsystems are rated with different criticality levels. The IV&V portion of this model is designed to conduct various IV&V activities on portions of the system that meet or exceed the criticality threshold set for each IV&V activity and shows the full life-cycle version of the IEEE 12207 Model.

The **Incremental Model** contains two levels of hierarchy and 28 individual process steps. As with the IEEE 12207 Lifecycle Model, the process steps were tailored versions of reusable generic process modeling blocks from the PATT. This model also includes preapproved process tailoring options that are accessible from a management dashboard screen. The dashboard can be used to quickly reconfigure the process and examine tradeoffs. Figure 4 shows a screenshot of this incremental version of the model, which was originally

developed at a leading software development firm using project-specific data based on past releases. For example, inspection data was collected from individual inspection forms for the past three releases of the project. Distributions for defect detection rates and inspection effectiveness were developed from these individual inspection reports. Effort and schedule data were collected from the corporate project management tracking system. Senior developers and project managers were surveyed and interviewed to obtain values for other project parameters when hard data were not available. The model was subsequently scaled up to function as an organizational life-cycle process model. The model used for the case studies presented in this report contains pull-down menus with SEPG preapproved composition options. To use this model, project managers simply select the desired process configuration to compose a project's defined process. Tradeoffs between composition options can be easily made using the pull-down menus.

Figure 3: Top-Level View of the Full Life-Cycle Version of the IEEE 12207 PSIM, Including IV&V Layer



Figure 4: Screenshot of the Incremental Version Model

## 3.1  USING PROCESS SIMULATION TO ARCHITECT, DESIGN, AND DOCUMENT PROCESSES TO IMPROVE UNDERSTANDING

### 3.1.1  Introduction/Overview

To get projects up and running quickly, organizations need to be able to architect, design, and document their processes in short order. Typically, projects start with an organizational process that must be tailored or composed to meet the objectives of the project at hand. How can processes be rapidly composed? How can these different process versions be tracked and controlled? What process documentation capabilities are available within existing process simulation tools?

Key capabilities associated with composing, designing, and documenting processes using process simulation include the ability to

- graphically represent the process
- easily create and compose process components and generic process blocks in a "plug and play" fashion to create full life-cycle process models
- store key process measures, indicators and parameters in databases that are tightly linked with life-cycle models
- create rich descriptions to document the process
- create reusable life-cycle process models
- store global model templates in a process asset library (PAL)
- use a PAL as the central configuration point for tracking organizational and project-specific processes as well as process components, preapproved process tailoring options, and process change opportunities. These preapproved tailoring options would be process alternatives previously approved by an organization and previously demonstrated to be compliant with CMMI or another appropriate standard.

Recent developments in DES models enable PSIM creation and tailoring to be done quickly (days and weeks rather than months) [Raffo 2005c].

PSIMs can be created to represent

- **specific projects** within a single organization using detailed data from that project. Models can be developed from scratch, from generic process building blocks using newer simulation tools, or from process components already stored in the PAL. Data can be developed from specific projects.

  The Incremental Model was originally developed as a project-specific model with data from several previous releases. This model was subsequently changed (i.e., slight modifications to the life-cycle process and changes to the data contained in the model) to reflect the organization's CMMI ML 3 Life-cycle Development Process. Additional process components were added to the model, and a pull-down menu tailoring option was added so that project managers could evaluate a variety of preapproved tailoring options and select the best option for their projects. This fulfilled the goals of the organization's PAL very well.

- **organizational processes** created for a specific development domain (e.g., large-scale DoD projects, embedded systems in the automotive domain, maintenance projects, COTS development, small-scale agile projects). These models are created with organization- and domain-specific data with the intention that these models can be tailored for specific projects.

  Both the Incremental and the IEEE 12207 models are used as organizational process models. For the IEEE 12207 Model, manned NASA missions are sufficiently similar that it provides a baseline model that can be tailored to specific projects. In addition, IV&V can be planned for various projects using this model as an organizational template.

- **standard life cycle models**, which are created with industry standard data for specific domains, such as military systems or embedded systems. The life cycle models would include standard waterfall, incremental, rapid prototyping processes as well as industry standard life cycles such as the IEEE 12207 Model.

  The IEEE 12207 Model contains all process steps included in the IEEE standard. The IV&V and V&V portions of the model comply with the IEEE 1012 Standard (in fact, the model as currently constructed provides a greater breadth of IV&V techniques than are included in the standard). Other industry standard life cycles including waterfall, product-line, XP, and spiral have been created or are in development.

Once a PSIM is created, it can function as a reusable process asset and can be stored in an organization's PAL. Process components can be stored in the PAL as well as alternative requirements processes (e.g., voice of the customer, quality function deployment, processes using automated requirements checking tools).

### 3.1.2 Model Description

This case study used the full life-cycle version of the IEEE 12207 Model. The top-level diagram for this model is again shown in Figure 5. Figure 6 then presents the second level of hierarchy by showing the major processes composing the Software Architecture & Detailed Design Phase. Figure 7 shows the generic building blocks composing the Software Detailed Design process.
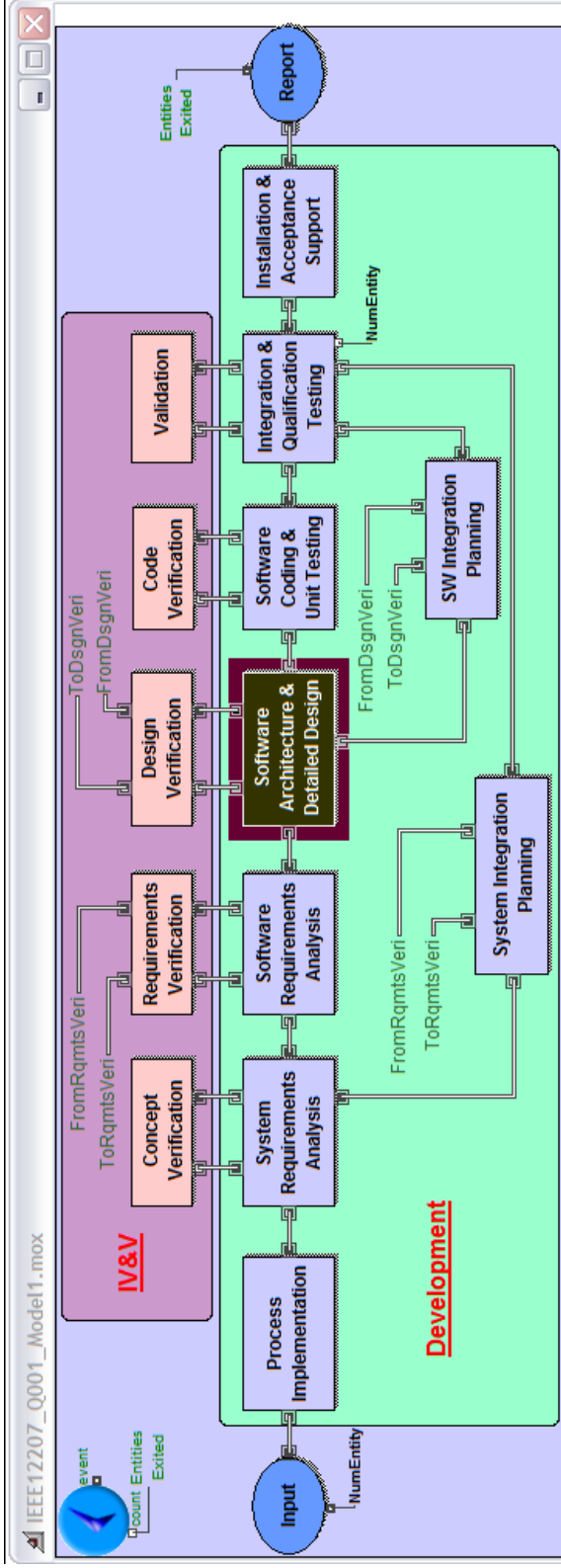
Figure 5:    IEEE 12207 PSIM Showing the Top-Level Life-Cycle Phases
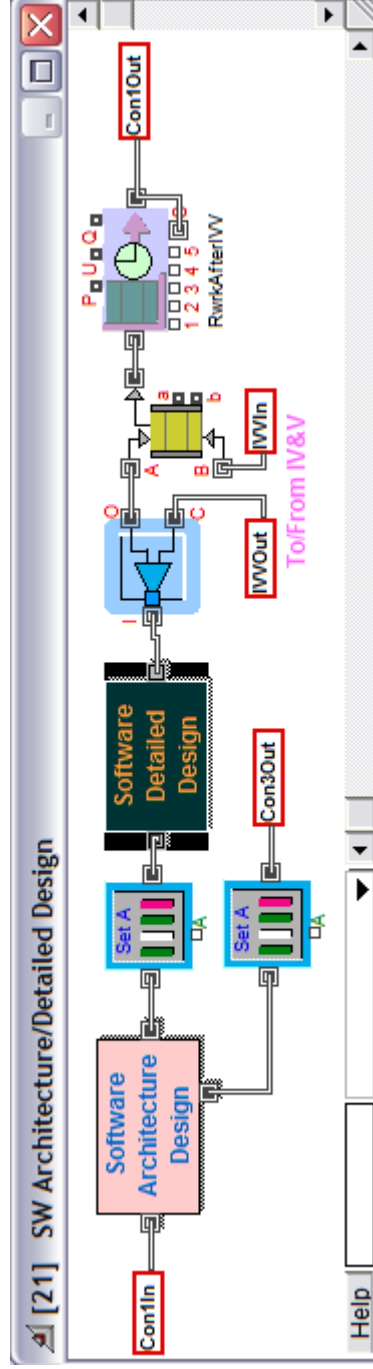


Figure 6:    IEEE 12207 PSIM Showing the Second-Layer of Hierarchy for the Software Architecture & Detailed Design Phase

Figure 6 indicates that the Software Architecture & Detailed Design phase comprises two main processes – Software Architecture Design and Software Detailed Design. In addition, an independent IV&V is performed on critical work products in this model. The final block in Figure 6 is a rework step to correct issues found during IV&V.

Figure 7 shows the workflow for the Software Detailed Design process, which comprises seven process steps, some done in parallel, some done in series.

The entire process shown in Figure 7 could be stored in a PAL to facilitate reuse when creating new PSIMs.

Reusable generic process building blocks available in the PATT tool were used to create this process. For example, the (1) Software Components Design, (2) Interface Detailed Design, (3) Database Detailed Design, and (4) Updating User Documentation process steps were all modeled using the same generic block. The generic model blocks were tailored to represent each specific process step using specific model parameters appropriate for that step, including productivity, defect injection rates, and so forth.

Figure 8 shows the top-level standard organizational incremental process life cycle. Pull-down menus are used to select project-specific tailoring options, such as Use-Case Analysis as the requirements elicitation (RE) method, the QuARS Automated Requirements Checking Tool, an Optimized Requirements Inspection, a Baseline High-Level Design Inspection, a Walkthrough at Detailed Design, and a Full-Fagan Code Inspection.

When a PSIM already exists, it can be easily tailored to a specific project. The Incremental version model shown in Figure 8 was preconfigured with SEPG-approved process tailoring options. As a result, the project manager could quickly select the desired options. In the case shown in Figure 8, the manager selected the options of Use-Case Analysis as the Requirements Engineering method, the QuARS Automated Requirements Checking Tool [Ferguson 2006, Lami 2005b], an Optimized Requirements Inspection, a Baseline High-Level Design Inspection, a Walkthrough at Detailed Design, and a Full-Fagan Code Inspection. Documentation for these processes can be stored in the PSIM database. This example ties in closely with the idea behind QPM SP 1.2 Compose the Process–identifying subprocesses with which to populate our project based on implications for quality, cost, and schedule.
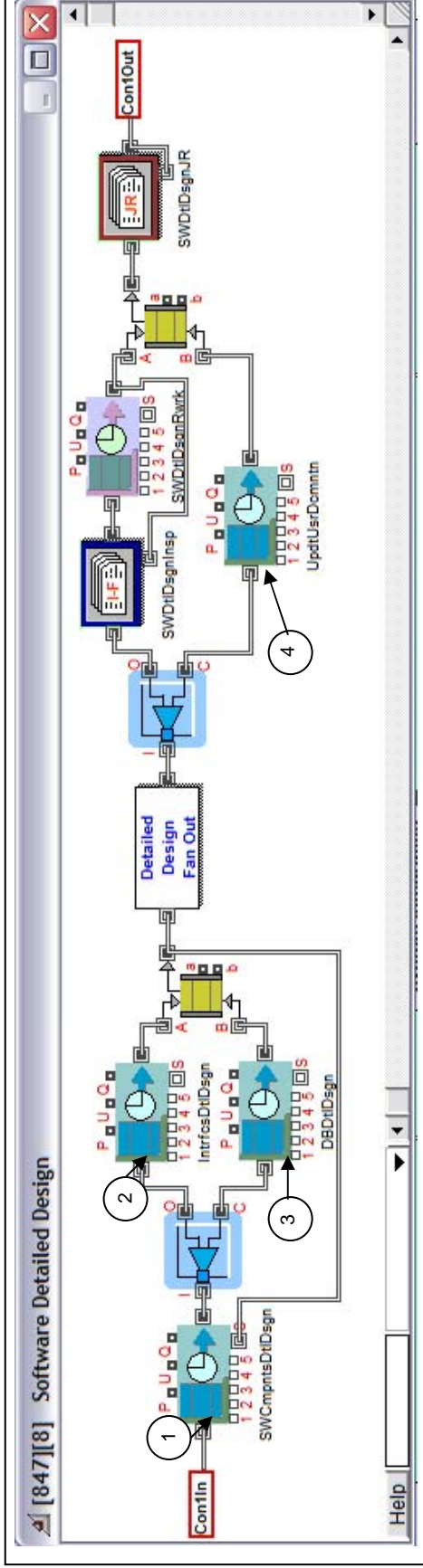
Figure 7: Process Activities Composing the Software Detailed Design for the IEEE 12207 Life-Cycle Process
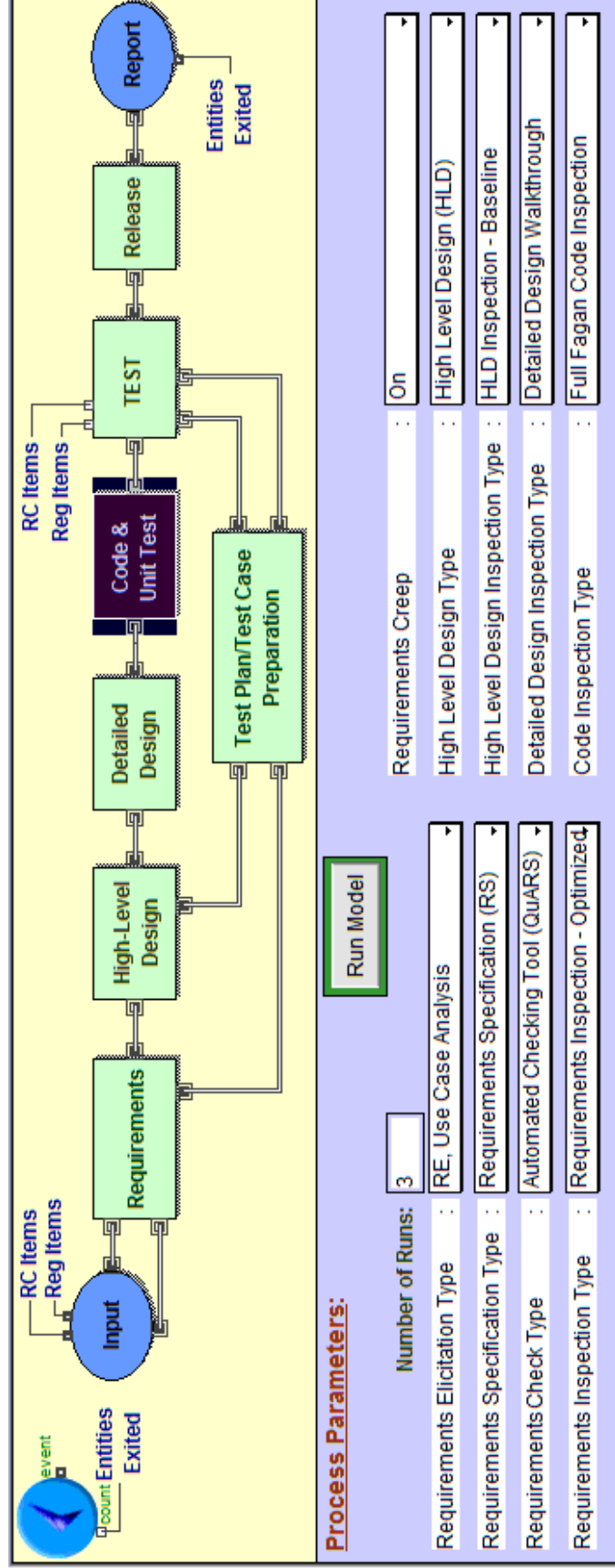


Figure 8: Incremental Life Cycle PSIM Configured for a Specific Project Using SEPG Preapproved Tailoring Options

### 3.1.3 Model Application/Results

PSIMs have been successfully used within multiple organizations to architect, design, and document specific project and organizational processes from generic process building blocks, process components, and compound process components (i.e., major portions of previously built process models). In other words, PSIMs can be highly reusable assets that leverage initial investment costs and provide returns to organizations in project after project.

PSIMs go beyond qualitative process models in that they provide quantitative predictions of the performance of the process. This is a key feature that enables tradeoffs and other high-value uses specified in this section. For the models mentioned, once the process is configured and the model has been appropriately tuned to the target development context, baseline process simulation results can be easily obtained. Table 3 shows top-level baseline results from running the model shown in Figure 8. The columns show the **metric** of interest (size, effort, rework effort, etc.), the **mean** or average value obtained, the standard deviation (**Std. Dev**.) and the **unit** of measure. The metrics shown in this table include

- **size** of the project, given in units of thousands of source lines of code (KSLOC). (This could also be given in function points.)
- total project **effort**, given in person months
- **rework effort**, which is a portion of the total effort, given in person months
- total project **duration**, given in months. The project duration indicates the latest finish-earliest start.
- **average (avg.) duration**, given in months. The average duration indicates the amount of time that would be required to complete the project if waiting times were eliminated.
- **injected (inj.) defects**, the number of defects injected throughout the project of all types
- detected (det) defects, the number of defects that were detected by all the defect detection steps throughout the project lifecycle of all defect types
- **corrected (cor) defects**, the number of detected defects that were reworked and corrected
- **latent defects**, the number of escaped defects that reach the customer of all types

The project database connected with the simulation model provides detailed information about effort, durations and defects (injected, detected, corrected and latent) by phase, by process step, and by defect type.[3]

---

[3]   Defect type can be defined by the user. In the case of the model shown in this example, defect type refers to requirements, design, code, documentation, bad fix, or test defect.

*Table 3:        Performance Measures for the Baseline Incremental Life Cycle PSIM Shown in Figure 8*

| | Metric | Mean | Std. Dev | Unit |
|---|---|---|---|---|
| 0 | Size | 103.1 | 15.7 | KSLOC |
| 1 | Effort | 590.2 | 90.4 | Person Months |
| 2 | Rework Effort | 91.7 | 14.2 | Person Months |
| 3 | Duration | 32.2 | 6.8 | Months |
| 4 | Avg. Duration | 21.7 | 3.4 | Months |
| 5 | Inj. Defects | 4949.6 | 757.2 | |
| 6 | Det. Defects | 4789.7 | 732.3 | |
| 7 | Cor. Defects | 4789.7 | 732.3 | |
| 8 | Latent Defects | 159.8 | 25.2 | |

### 3.1.4        Supporting CMMI Processes and Practices

This case shows how PSIM strongly supports CMMI practices relating to defining and composing processes, planning processes, creating process assets, and objectively assessing processes. Specifically, this example shows how PSIM strongly supports Organizational Process Definition SG1, Integrated Project Management, SP 1.1 and 1.2. It also strongly supports Organizational Process Focus, Quantitative Project Management, SP 1.2, Organizational Process Performance, SP 1.5, and Generic Practices 2.2, 3.1, and 4.1.

Note in particular that PSIMs, as described in this section, support implementation of high maturity practices such as those mentioned above (OPP 1.5, QPM 1.2) as well as the corrective action component of QPM 1.4. We refer to use cases presented later in this section that make the ties more clear.

Note also that because PSIMs are quantitative, they can provide a framework and a focus for metrics programs. Some PSIMs have specific parameters that projects are required to collect. Others rely on industry standard datasets with some modest customization. Some PSIMs provide databases connected with their models and can therefore store project metrics and parameters. Overall, then, PSIMs support establishing and defining process and product measures and estimates. They also support data analysis and storage of data and analysis results. Hence, PSIMs support the Project Planning PA, SG1 (specifically SP 1.1 – 1.4), and the Measurement and Analysis PA, SG1 (SP 1.1 – 1.3) and SG2 (SP 2.3).

### 3.2  USING PROCESS SIMULATION TO SUPPORT BOTTOM-UP COST ESTIMATION

### 3.2.1        Introduction/Overview

Cost estimation is a critical activity for systems- and software-development organizations. Accurate estimates are vital for bids on new projects, plans, hiring decisions, and so forth. Ultimately, good estimates are essential to a successful project. Key questions for which managers want answers include

- Can PSIM be used to make cost estimates?
- How reliable are the cost estimates provided by PSIM?
- How much data is required?
- At what stage in the estimation process should PSIM be used?

Industrial practitioners generally believe that PSIM requires a tremendous amount of data, without which it provides no value. Some projects may not have such data available.

It is true that when PSIMs have been applied in industry to estimate project costs using detailed project data such as defect injection and detection rates, productivity, size, effort, and schedule data, the accuracy of these models can be within 15% [Mueller 2006, Raffo 2005b, 1996].

At the same time, recent works by Mizell and Raffo have applied PSIM in large contexts using limited project-specific data [Mizell 2006, Raffo 2005b]. Mizell's work in particular utilized PSIM to estimate a large-scale NASA project at the pre-concept phase as well as later phases of the project using industry standard data and NASA data from other sites. Her work showed that PSIM can provide accurate and insightful estimates when based upon industry and organizational standard data coupled with a standard organizational process.

As a result, we recommend utilizing PSIMs to create independent bottom-up estimates to be used in conjunction with top-down approaches such as COCOMO or SLIM.

### 3.2.2    Model Description

In this section, we summarize the work conducted by Mizell at NASA Kennedy Space Flight Center [Mizell 2006]. This work utilized a process model of the IEEE 12207 Life Cycle Development Process (see Figure 3). This PSIM used industry-standard data based on work by Jones [IEEE 1998, Jones 1991, 2000] and defect injection rates, productivity, and effort data from the Software Engineering Laboratory at Goddard Space-Flight Center [CeBASE 2003, 2005, Landis 1990, SEL 1993, 1994, 1997]. The model was then used to estimate the costs associated with a large-scale NASA project. This model had previously been used to predict the impact of new technologies on NASA projects [Raffo 2005b] (see Section 3.5).

COCOMO-based estimates of the project produced optimistic estimates of project costs and schedules, due to a variety of factors described by Mizell [Mizell 2006]. PSIM was then used to independently estimate project cost and schedule. The resulting estimates were significantly higher than the COCOMO-based estimates and, in the end, provided much more insight to project managers. Some key factors that Mizell's PSIM took into account were as follows.

- **realistic project size variation at various stages of the project.** Mizell estimated project costs and schedule at the pre-concept phase as well as two other points later in the project using updated information about project size and typical project size variation. This approach showed that optimistic projections made by project management did not prove realistic when historical variations of early-phase project size estimates were considered [Mizell 2006]. This use of PSIMs to re-estimate completion and attainment of project objectives mid-stream is similar to one use made of process performance models (PPM) in the related QPM PA.
- **human factors related to staff turnover**. Mizell incorporated well-known models for staff turnover and learning into the model [Abdel-Hamid 1991]. Staff turnover was a significant factor in the performance of the selected project. Mizell's model provided insight into the added delay caused by underfunding of projects.
- **incremental spiral enhancements**. Mizell's model also enhanced the standard IEEE 12207 Model by representing increments embedded in a spiral development process.

Figure 9 shows an incremental spiral model that was also used to estimate project costs and schedule at NASA. This model added a framework for developing multiple increments and a risk assessment spiral. The model predicted even higher costs than those predicted by the regular IEEE 12207 Model.

### 3.2.3    Model Application/Results

A key element of this case study was to examine the variability associated with key process factors–specifically project size, productivity, defect injection and detection rates, and effort. This variability was represented using organizational-specific data from NASA for similar large-scale projects. Understanding the variation provided value to project managers, helping them recognize the basis for higher project cost estimates than planned.

This opportunity to explicitly and systematically look at project size variation and the variation of other critical factors is an important benefit provided by PSIM. Furthermore, using PSIM enabled the project manager to look at contingencies such as V&V effectiveness; the effect of requirements volatility, staff turnover, and areas where the project is not sufficiently staffed (i.e., bottlenecks); and other common project issues.

Figure 10 shows how project size estimates vary over the life cycle of a project [Boehm 2000]. During feasibility studies, project size (a key parameter in almost all cost estimation routines) can vary by as much as four times, while in later phases of the project, project size variation is much reduced.
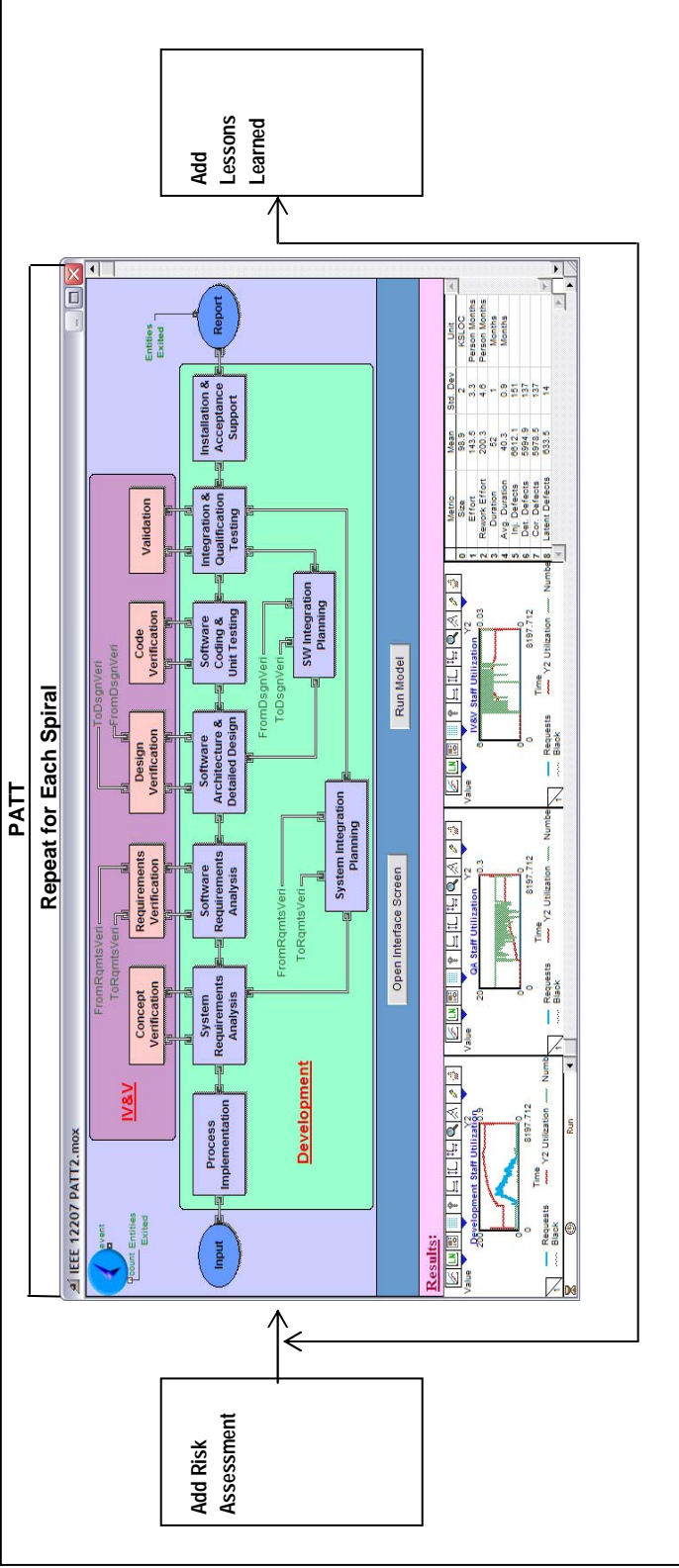
*Figure 9:*    *IEEE 12207 Model Embedded in Incremental Spiral Development Process*
*(adapted from Mizell 2006)*
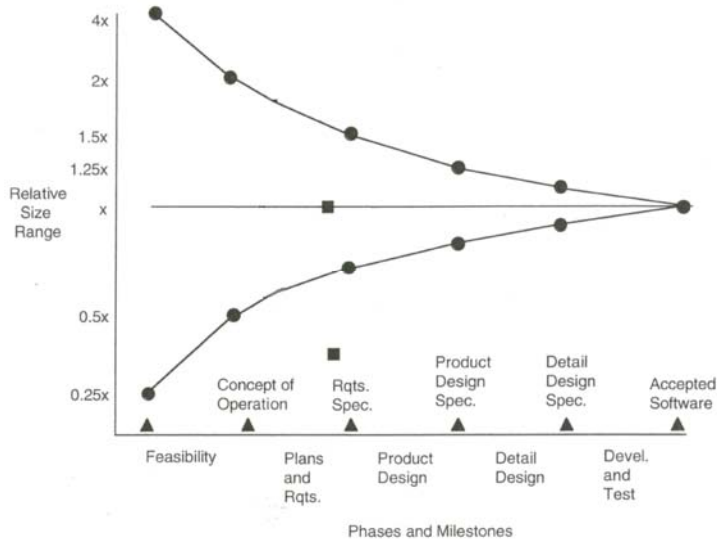
*Figure 10:     Size Estimate Accuracy [Boehm 2000]*

*Estimating Project Costs at Different Points in Time*

Another key benefit of simulation is that models can be reused to provide updated estimates at any point in the project, and the ability to update these estimates supports implementation of QPM SP 1.4. Model parameters can be updated based on new information derived from activities conducted on the project (Section 3.5).

Table 4 shows Mizell's updated project cost estimates, made using PSIM for a large-scale NASA project [Mizell 2006]. The second column shows the size estimates made over time by expert developers working on the project. As the project progressed, the size estimates for the project increased by more than a factor of four.

To estimate project costs, Mizell utilized the size estimates made by project personnel coupled with a project size variability factor derived from Figure 10 [Mizell 2006]. The fourth and fifth columns in Table 4 show Mizell's estimates for project effort and duration made using the IEEE 12207 PSIM. Due to the inclusion of project size variation, historical NASA data, and an industry-standard life-cycle model (IEEE 12207), Mizell's estimates using PSIMs proved to be more accurate than those of the project personnel. In addition, the predicted intervals around these estimates became smaller to reflect the improved information available later in the project. Predictions about project completion were thereby improved.

Overall, this case study demonstrated that project analysis with simulation models provided more reasonable effort and duration estimate intervals for different points in the project. Furthermore, using PSIM provided insight into how the selected life-cycle approach should have affected effort and duration estimates.

*Table 4:        Estimates Made Using PSIM at Different Points in the Project.*

| Date | Size Estimate (Million LOC) | Model Type | Effort Estimate (Labor Years) | Duration Estimate (Years) |
|---|---|---|---|---|
| Feasibility stage, T = 0 | 1.4 | Waterfall | [1466, 2214] | [6.1, 9] |
| T = 4 months | 3.425 | Waterfall | [2123, 2190] | [8.9, 11.5] |
| T = 36 months | 5.8 | Waterfall | [2667, 3083] | [11.3, 13] |

### 3.2.4        Supporting CMMI Processes and Practices

This case study shows how PSIMs can be used to estimate project performance at early as well as later stages of the project. As later sections of this report focus on replanning, we will not address that issue here. At ML 2, PSIMs strongly support SG1 of the Project Planning PA by providing support for establishing estimates. PSIMs also support parts of SG2 during project plan development by helping to establish a budget and schedule (in conjunction with other tools), identifying process-related risks, and providing support for making resource decisions. Moreover, IPM is about managing the project according to an integrated end-to-end defined process, that itself evolves as the project progresses and as project risks and objectives change. Having such a PSIM capability can help the project identify earlier the need for changes or other modifications/improvements in the project's defined process and the associated project plan. This allows projects to more effectively allocate staff and resources to project tasks, improving overall project performance.

### 3.3  USING PROCESS SIMULATION TO SUPPORT IMPROVED PROCESS PLANNING AND TRADEOFF DECISIONS

### 3.3.1        Introduction/Overview

When a new project starts, a process must be selected. An organization may have several preapproved life-cycle development processes. In addition, a variety of process options and alternatives may be available. In this section, we will address the questions: What is the impact of different process tailoring or composition options on overall project performance? How can project performance be improved? For example, how can an organization evaluate the impact of selecting one requirements elicitation process vs. another? Similarly, how can an organization evaluate the impact of using one inspection approach vs. another?

In this section, we explore how PSIM can be used to quantitatively evaluate tradeoffs among different process alternatives.

### 3.3.2        Model Description

In this section, we summarize the work conducted by Mizell at NASA Kennedy Space Flight Center [Mizell 2006]. This work utilized a process model of the IEEE 12207 Life Cycle Development Process (see Figure 3). This PSIM used industry-standard data based on work by Jones [Jones 1991, 1998, 2000] and defect injection rates, productivity, and effort data from the Software Engineering Laboratory at Goddard Space-Flight Center [CeBASE 2003, 2005, Landis 1990, SEL 1993, 1994, 1997]. The model was then used to estimate the costs associated with a large-scale NASA project.

This model had previously been used to predict the impact of new technologies on NASA projects [Raffo 2005b] (see Section 3.5).

The requirements elicitation process has three options:

1.   organization's standard requirements elicitation process
2.   use case analysis
3.   voice of the customer

The other points in the process where choices are available include

- requirements specification (Options include standard requirements specification, concept of operations, and quality function deployment.)
- automated requirements checking tool (Options include apply tool or not apply tool.)
- requirements inspection (Options include no inspection, baseline inspection, walk through, optimized inspection, and formal Fagan Inspection [Fagan 1986].)
- high-level design (Options include standard HLD, HLD after formal concept selection [CS], HLD after failure mode and effects analysis [FMEA], and HLD after a formal CS and FMEA.)
- high-level design inspection (Options include no inspection, baseline inspection, walk through, optimized inspection, and formal Fagan Inspection.)
- detailed design inspection (Options include no inspection, baseline inspection, walk through, optimized inspection, and formal Fagan inspection.)
- coding inspection (Options include no inspection, baseline inspection, walk through, optimized inspection, and formal Fagan Inspection.)

In addition, requirements creep was specifically modeled and its effects can be turned on or off.

As previously mentioned, the incremental PSIM was developed at a large commercial development firm and represents the process used for one of the company's flagship consumer products. At its peak, approximately 60 development engineers staffed this project. The software is in its fifth release. Data from previous releases was utilized to estimate parameters for the PSIM. Data from the literature and expert opinion were used when the development staff did not have experience with particular process options.

### 3.3.3      Model Application/Results

Management clearly recognized that the project's baseline inspections were poor. This put a great deal of pressure on the testing operations. In particular, Unit Test (UT) bore the brunt of finding the high number of accumulated defects in the code. One process improvement considered by the project was to assess the impact of various types of inspections. Some of the developers wanted to eliminate inspections altogether and felt that testing was good enough. The following process alternatives were also considered: basic optimization of the current inspection process with additional training, structured walkthroughs, or full Fagan Inspections to achieve a high defect detection rate. Using the PSIM, management was able to explore all of these options. For the purposes of this section (3.3), we will look at one specific configuration using the implementation costs shown in Table 5, and the inspection effectiveness information shown in Table 6.

*Table 5:    Implementation Costs for Documentation, Training, and Ongoing SEPG Support to Improve Inspections by Inspection Type*

| Process Option Selected | Cost to Change from Baseline Inspection |
|---|---|
| No inspection after either requirements or coding | $0 |
| Continue with baseline inspection | $0 |
| Walk-through inspections | $100K |
| Optimized inspections | $50K |
| Full Fagan Inspections | $200 K |

*Table 6:    Differences in Effectiveness for Each Inspection Type Compared to Baseline Inspections*

| Inspection Type | Number of Staff | Approximate Effectiveness |
|---|---|---|
| Baseline | 4 | 10%–20% |
| No inspection | 0 | 0% |
| Walkthroughs | 2 | 20%–30% |
| Optimized | 3 | 25%–40% |
| Full Fagan | 5 | 40%–60% |

To configure the PSIM, the user selects options from the pull-down menus at each of the choice points in the process. Figure 11 shows the baseline As-Is process options selected and the resulting performance for approximately a 103 thousand lines of code (KLOC) system. Table 10 shows an enlarged view of the performance details provided in the lower–right-hand corner of the One-Screen interface.

Choosing other process configurations results in different performance. Figure 12 shows the baseline As-Is process with full Fagan Inspections after the requirements and coding phases. Table 10  shows the performance results for the current process.

The PSIM can view any option or any combination of options available from the pull-down menus in minutes. These options can be preloaded and the PSIM made available to managers, for planning projects with preapproved tailoring choices. Figure 12 shows one of the options available on the pull-down menus of the PSIM. This was one of the options assessed during evaluation of the overall QA strategy for this process and project.

To evaluate the impact of changing from the baseline As-Is process shown in Figure 11 to the To-Be process shown in Figure 12, we need to determine the difference in performance of the two process configurations. A detailed presentation of how to do this is provided by Raffo [Raffo 1996]. In this report, we will present a summarized discussion evaluating the two process options.

*Evaluating the Differences in Key Performance Measures*

Table 9 shows the performance of the As-Is and To-Be processes from Figure 11 and Figure 12 for a system approximately 103 KLOC in size.
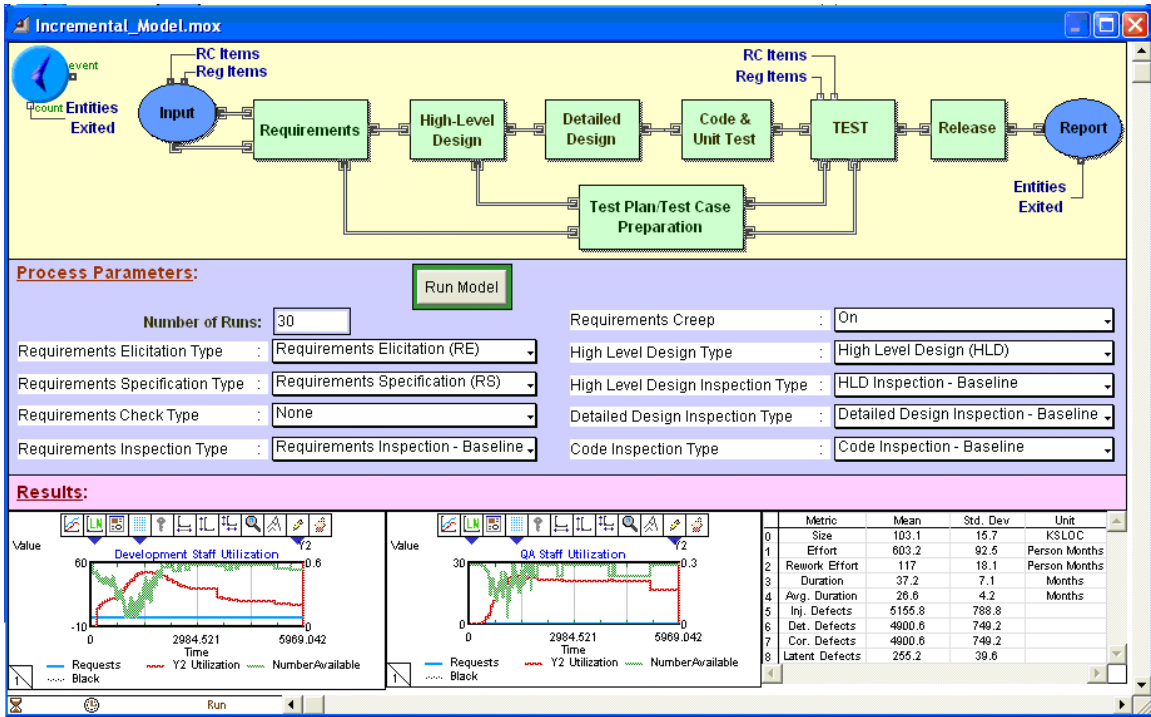
*Figure 11:    Incremental Model Showing Default Baseline Process and Results*

*Table 7:    Performance Measures for Baseline As-Is Process from Figure 11*

|   | Metric | Mean | Std. Dev | Unit |
|---|--------|------|----------|------|
| 0 | Size | 103.1 | 15.7 | KSLOC |
| 1 | Effort | 603.2 | 92.5 | Person Months |
| 2 | Rework Effort | 117 | 18.1 | Person Months |
| 3 | Duration | 37.2 | 7.1 | Months |
| 4 | Avg. Duration | 26.6 | 4.2 | Months |
| 5 | Inj. Defects | 5155.8 | 788.8 | |
| 6 | Det. Defects | 4900.6 | 749.2 | |
| 7 | Cor. Defects | 4900.6 | 749.2 | |
| 8 | Latent Defects | 255.2 | 39.6 | |
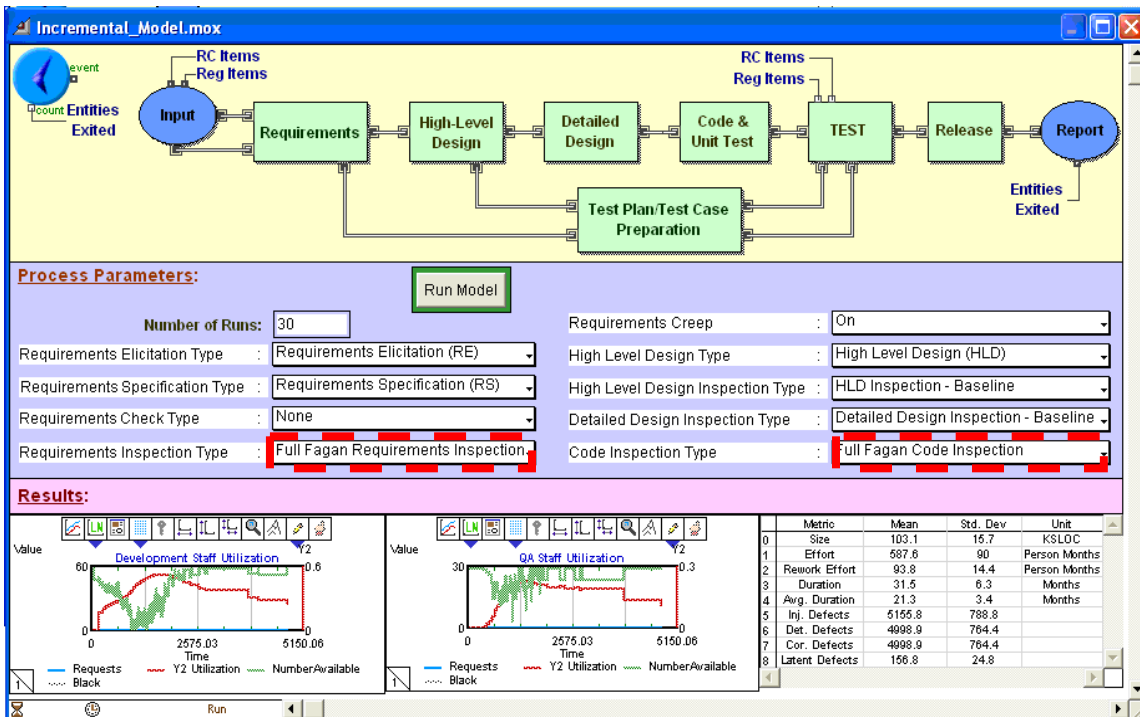
*Figure 12: Incremental Model Showing Default Baseline with Full Fagan Requirements and Code Inspections*

*Table 8: Performance for Incremental Process With Full Fagan Inspections After Requirements and Coding*

| | Metric | Mean | Std. Dev | Unit |
|---|---|---|---|---|
| 0 | Size | 103.1 | 15.7 | KSLOC |
| 1 | Effort | 587.6 | 90 | Person Months |
| 2 | Rework Effort | 93.8 | 14.4 | Person Months |
| 3 | Duration | 31.5 | 6.3 | Months |
| 4 | Avg. Duration | 21.3 | 3.4 | Months |
| 5 | Inj. Defects | 5155.8 | 788.8 | |
| 6 | Det. Defects | 4998.9 | 764.4 | |
| 7 | Cor. Defects | 4998.9 | 764.4 | |
| 8 | Latent Defects | 156.8 | 24.8 | |

*Table 9: Performance Deltas for Process Options Shown in Figure 11 and Figure 12*

| Performance Measure | AS-IS Process (Figure 11 & Table 11) | TO-BE Process (Figure 12 & Table 9) | Difference in Performance * |
|---|---|---|---|
| Total effort (Effort + Rework Effort) | 720.2 PMs** | 681.4 PMs** | 38.8 PMs |
| Project duration | 37.2 Months | 31.5 Months | 5.7 Months |
| Latent defects*** | 255.2 Defects | 156.8 Defects | 98.4 Defects |
| Implementation costs | | | $200K |

* Positive value means improvement in To-Be process
** Person months
*** Predicted number of defects passed on to customers

*Computing Return on Investment (ROI) and Net Present Value (NPV)*

Using the information from Table 9, we can compute the ROI and other financial measures of performance. For information regarding how to compute financial measures, see the texts in Engineering Economics by Grant and colleagues [Grant 1990]. Also, for an interesting treatment of risk and its impact on ROI, see the article by Harrison and colleagues [Harrison 1999].

In computing the ROI the basic idea is to compare the performance of the As-Is Baseline scenario to various To-Be scenarios of interest. The simulation model computes the performance of each scenario. A spreadsheet shows the differences in each of the performance measures, as provided in the last column of Table 9.

Next, each of the performance measures is converted to cash equivalents. This can be done fairly easily with the Total Effort and Latent Defects. The cost of Total Effort comes down to the cost of staff time. For Latent Defects, since many companies do compute the average effort required to correct field defects, this performance measure also converts to the cost of staff time. Implementation costs (consisting of SEPG and staff time, the cost of new tools, the cost of training, etc.) can easily be computed as well, because they comprise the cost of staff time and cash outlays for tools and training. The timing of each of these expenses (or cost savings as in this case) must also be taken into account. For this example, the following assumptions represent a typical case:

- One person month of effort is worth $12,500 per month ($150K annual fully loaded salary).
- Implementation costs are incurred now (T = 0) and savings from effort are incurred at the end of year 2 (T = 2).[4]
- The company knows the average number of work hours associated with repairing defects that are released to the field (0.75 months per defect), which has already been included in the total effort performance measure numbers shown in Table 7, Table 8, and Table 9.

The main challenge for this example is to determine the value of a change in schedule. Usually, this value is set by marketing or some other organization outside the development team. We will further discuss this later in the report.

Once the cash equivalents for the changes in performance measures have been determined along with their timings, financial performance measures can be calculated. There are several financial measures of interest to development managers, including

- Net Present Value (NPV) – determines the amount of cash an investment opportunity is worth over and above the organization's hurdle rate.[5] The MS Excel function call is (NPV). For this example, we will assume that the organization's hurdle rate for making investments is 30%.
- Return on Investment (ROI) – determines the rate of return an organization would receive by investing its money in the opportunity under consideration. It is determined by setting the NPV

---

[4]   It would be more accurate to distribute the labor savings over the life of the project (with most of the savings being realized toward the end of the project). Formulas are available to compute this [Grant 1990]. For the purposes of this example, we chose to place all the labor savings at one point in time toward the end of the project as a useful approximation that may be computed much more simply.

[5]   The *hurdle rate* is the organization's minimum acceptable rate of return for making investments.

equation to zero and solving for the rate of return rather than using the organization's hurdle rate. The MS Excel function call is (IRR)

- Cost-Benefit Ratio (CBR) – the costs in proportion to the benefits. The benefits are summed in the numerator and the costs are summed in the denominator. This financial measure is often used in return on investment computations made by the government.
- Discounted Cost Benefit Ratio (DCBR) – the benefits and costs discounted at the organization's hurdle rate.
- Payback Period (PP) – the amount of time required by an organization to recover its initial investment.
- Discounted Payback Period (DPP) – discounts the cash flows at the organization's hurdle rate when computing Payback Period.

For the purposes of this section, we will show an example for how to compute NPV and ROI only. To calculate NPV, we simply discount and sum all of the relevant cash flows[6] using the organization's hurdle rate. This is shown in Equation (1). ROI is calculated by setting the NPV equation to zero and solving for r as shown in Equation (2).

(1) $\quad NPV_r = IC + \sum Cash\ Flow_{i,j} /(1+r)^i$ for all i and j.

Where:

- i equals the time period
- j equals different cash flows during that time period
- r equals hurdle rate for Equation (1) and the internal rate of return for Equation (2)
- IC equals total implementation costs at time 0.

(2) $\quad 0 = IC + \sum Cash\ Flow_{i,j} /(1+r)^i$ for all i and j.

Calculating the NPV for this example, r=30% and the remaining numbers are taken from Table 9 to give

(3) $\quad NPV(r = 30\%) = -\$200K + 38.8\ PMs*\$12,500/PM/(1+30\%)^2$
$NPV = \$86,983$

To determine the ROI, we solve equation (1) for r:

(4) $\quad 0 = -\$200K + 38.8\ PMs*\$12,500/PM/(1+r)^2$
$r = 55.7\%$

This looks very positive. Furthermore, given that the schedule impact is also favorable, it seems that this is a clear improvement over the As-Is process.

---

[6]    *Cash flows* are the net of all benefits and costs at a particular time period.

However, in a globally distributed development operation, annual salaries may not be as high as $150K/year. It could be that the average salary paid for off-site development is closer to $60K/year. In this case, the economic benefit is negative.

(5)      $NPV(r = 30\%) = -\$200K + 38.8\ PMs*\$5,000/PM/(1+30\%)^2$
         $NPV = -\$85,207$

(6)      $0 = -\$200K + 38.8\ PMs*\$5,000/PM/(1+r)^2$
         $r = -1.5\%$

In this case, evaluating the impact of the To-Be process change is a bit more challenging. The ROI is negative, but latent defects and schedule are still improved.

As mentioned, the rework costs associated with field defects had already been taken into account. In this case, the average cost per defect was 0.75 person-months. However, quantifying the market impact of releasing a product with defects is more elusive. Moreover, determining the impact of schedule differences is also challenging. What is the value in revenue terms of releasing a product with improved quality? What is the value in terms of revenue of delivering a product one month earlier? Six months earlier? The answer depends on the product and the market. As a result, it is necessary for marketing staff to estimate the potential change in revenue and then to use that number.

For the purposes of this example, we will assume that releasing the software on average 5.7 months earlier and improving the quality is worth $250K in increased revenue after the product is released.[7]

Using this information, we can calculate the full NPV and ROI. As mentioned, the financial performance is very sensitive to the cost of labor. If staff salaries are near $150K per staff year, then the To-Be process makes clear sense. If wages are more similar to $60K/year for globally distributed development, and we assume that the market impacts are realized at the end of year 3 (T = 3), the resulting ROI calculation is as follows:

(7)      $NPV(r = 30\%) = -\$200K+38.8*\$5,000/(1+30\%)^2 + \$250K/(1+30\%)^3$
         $NPV = +\$28,584$

(8)      $0 = -\$200K + 38.8\ PMs*\$5,000/PM/(1+r)^2 + \$250K/(1+r)^3$
         $r = 37.2\%$

This shows that when we include the additional factor associated with market impact, the ROI and NPV indicate that this process change would be a good investment.

---

[7]    Most revenue from the release of a new software product is realized over a period of 6 to 12 months depending upon the product and the competitive environment. A model could be created to spread out the revenue over multiple months and to discount the revenue back on a monthly basis. For the purposes of this example and simplicity, we chose to place all the revenue at the three-year mark (10 months after the product was released).

### 3.3.4    Supporting CMMI Processes and Practices

This case study of trading off process alternatives quantitatively demonstrates how PSIMs can contribute to fulfilling the following process areas at maturity levels 2, 3, 4, and 5:

- Project Planning (ML 2), SP 1.3 define project life cycle, SP2.2 identify project risks
- Integrated Project Management (ML 3) SP 1.1 and Quantitative Project Management (ML 4) SP 1.2, respectively, on establishing and composing the project's defined process
- Risk Management (ML 3) by identifying process risks and assessing the risk associated with proposed process changes. In addition to the expected values described above, PSIMs predict stochastic ranges for process performance measures. Using this information, a quantitative assessment of the risk associated with implementing a proposed process improvement can be computed.
- Decision Analysis and Resolution (ML 3) by providing decision guidelines, processes, evaluation criteria, alternative solutions, evaluating alternatives, and making specific recommendations
- Organizational Process Focus (ML 3) by identifying process improvements and developing an implementation plan
- Organizational Process Performance (ML 4) by helping select processes, measures, specific performance objectives, baselines, and performance models
- Organizational Innovation and Deployment (ML 5) by helping evaluate, select, and deploy incremental and innovative improvements that can measurably improve the organization's processes and technologies. The improvements support the organization's quality and process-performance objectives as derived from the organization's business objectives.
- Causal Analysis and Resolution (ML 5) by aiding understanding of the causes of process problems and evaluate alternative action plans to resolve the problem

## 3.4  USING PROCESS SIMULATION TO ANALYZE AND EVALUATE PROCESS IMPROVEMENT OPPORTUNITIES QUANTITATIVELY

### 3.4.1    Introduction/Overview

Some organizations have many different process change ideas and opportunities. These can be generated internally by input from software engineers working directly on projects or from management seeking specific efficiency or performance gains to improve the organization's competitive advantage. Process changes can be generated externally as well from industry standards such as the CMMI- or Six Sigma-based process improvement programs. Some organizations have databases containing many more process improvement opportunities than they could implement. For example, one organization known to the authors had a database containing over 200 process improvement opportunities it had identified and wished to consider. Whatever the source, before a company invests scarce resources and staff effort into implementing process improvements (CMMI-based or otherwise) management will want to know the expected ROI as well as the level of risk involved.

### 3.4.2    Model Description

This case study was conducted at a leading software development firm that utilized PSIM to evaluate the potential impact of an internally generated process change. The problem was that the As-Is Process was releasing defective products and had high schedule variance. The suspected reason was the firm's reliance on Unit Testing (UT) as the main defect removal process. The organization wanted to assess the impact of upgrading its UT process. The management questions addressed by this study included

- Will the process alternative improve overall project performance?
- What is the cost the firm is currently paying by conducting its current UT process in an ad hoc, highly variable manner?
- Is partial implementation of the proposed process alternative possible?
- Would alternative process choices offer a greater improvement?
- Do we gain additional benefits (i.e., reduced cost, increased productivity) when this process change is executed a second or third time?

This last question was reframed into two questions:

1. How would potential learning curve effects affect the new process outcome?
2. Can the project benefit from reuse of process artifacts?

PSIMs were used to assist with a live process improvement decision at the subject firm. Development managers within the organization were very interested in their processes and how these processes could be improved to increase performance and reduce schedule. Managers were concerned with questions ranging from "How do we select between the waterfall or spiral process models for a particular project?" to "What is the impact if we start collecting inspection comments using an automated tool rather than doing it manually?"

The PSIM analysis focused on mid-range process changes that could have a significant impact on process performance but could not easily be studied using cost estimation models such as COCOMO II [CSE 2004], SLIM [QSM 2007], SEER [Galorath Incorporated 2004], or KnowledgePLAN [Software Productivity Research 2007].

The process alternative studied in detail was to **create UT plans** during the coding phase as described in greater detail below. Note that although this case study focused on the UT process, other process alternatives could have also been evaluated using this approach, such as combining HLD and LLD phases, conducting UT before the code inspection, and inspecting only high-risk portions of the product at the design and coding phases.

*Characteristics of the Case Study Site*

Key aspects of the environment in which the case study was conducted are listed below.

- The large company had a number of development sites throughout the world.
- Between 10 and 20 major development projects were underway simultaneously at the study site.

- Work scope included worldwide system support of most of the products being developed at the site (including the one being studied). Support professionals were experienced developers who had been assigned to correct the escalated field defects. When these support professionals had time available, they carried out development tasks.
- The site was ISO 9001 certified and had been assessed at CMMI maturity level 3.
- The product studied had completed five successive major releases when the study began.
- In each release, major new functionality was added, and existing functionality was substantially revised.
- At peak development periods, the project involved over 70 people.

**Overview of the Baseline Process**

The baseline development process studied essentially followed a Waterfall Process Model, including the major phases of Functional Specification (FS), High-Level Design (HLD), Low-Level Design (LLD), Coding (CODE), Unit Test (UT), Functional Verification Test (FVT), and System Verification Test (SVT). Inspections of the FS, HLD, LLD, and CODE were also conducted. After completion of SVT and FVT, the product was released to the public. These phases, as well as a period devoted to field support and maintenance, were captured by the PSIM. In addition, the process segments for developing test plans and writing test cases for functional test and system test were also included. The test plan development and test case writing phases were conducted in parallel to the development phases of functional specification through UT mentioned above.

Figure 13 shows the process being modeled.
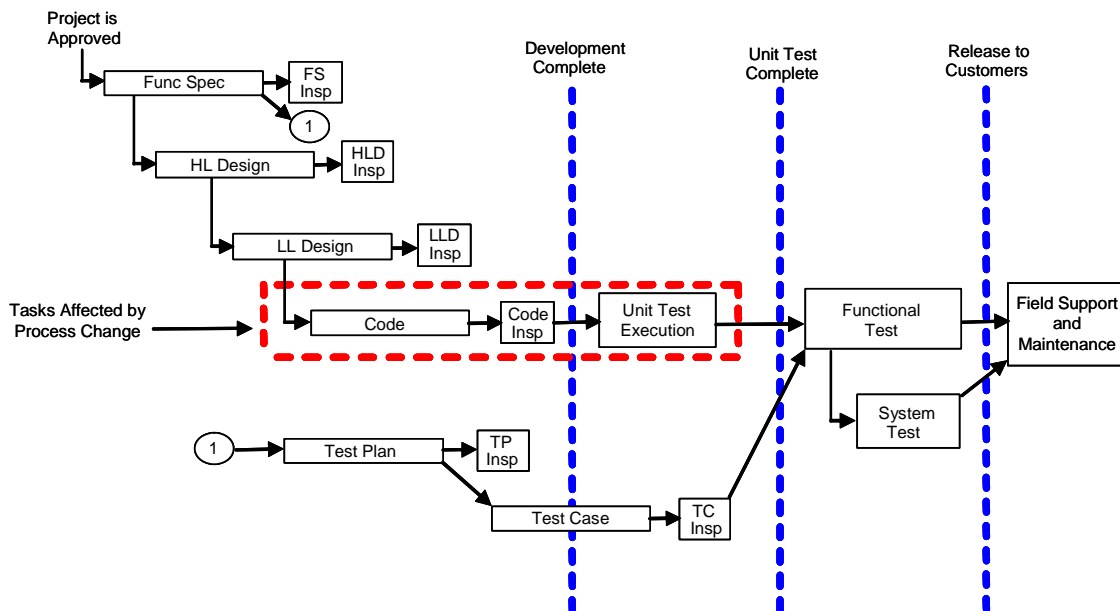


*Figure 13:     Life-Cycle As-Is Process*

*Description of the Process Improvement Being Considered*

Creating UT plans during the coding phase was chosen as the process alternative to enhance the quality and speed of UTs and to encourage developers to resist the temptation of hurrying through UT due to schedule pressures. The change required UT plans to be developed during the coding phase and inspected during code inspection. These plans would be subject to rework if not satisfactory and would be required as part of the exit criteria from the coding phase. The UT plans would specify the methods and scenarios for testing the code. Another added benefit was that, in the process of creating these plans, developers would actually conduct an informal review of their code. The potential benefits of this process choice included

- enhancing effectiveness of the UT phase - in terms of coverage and error detection efficiency
- reducing the temptation of rushing through UT due to schedule pressures
- reducing the number of errors before code inspections
- reducing the effort during UT due to following a planned test approach
- reducing potential downstream defects including FVT, SVT and field defects
- obtaining possible schedule reductions due to removing additional errors earlier in the development cycle
- building an artifact that could be reused for subsequent releases to structure and increase the efficiency of UT

The major costs associated with the process change were

- effort to develop the test plans
- effort to implement the change ( including training staff to write and inspect unit test plans, SEPG effort to create new process documentation, SEPG mentoring and follow-up, etc.)
- inspection effort for inspecting and reworking the test plans
- possible schedule delays associated with the development and execution of the plan

A diagram of the main process steps of the process change are shown in Figure 14.
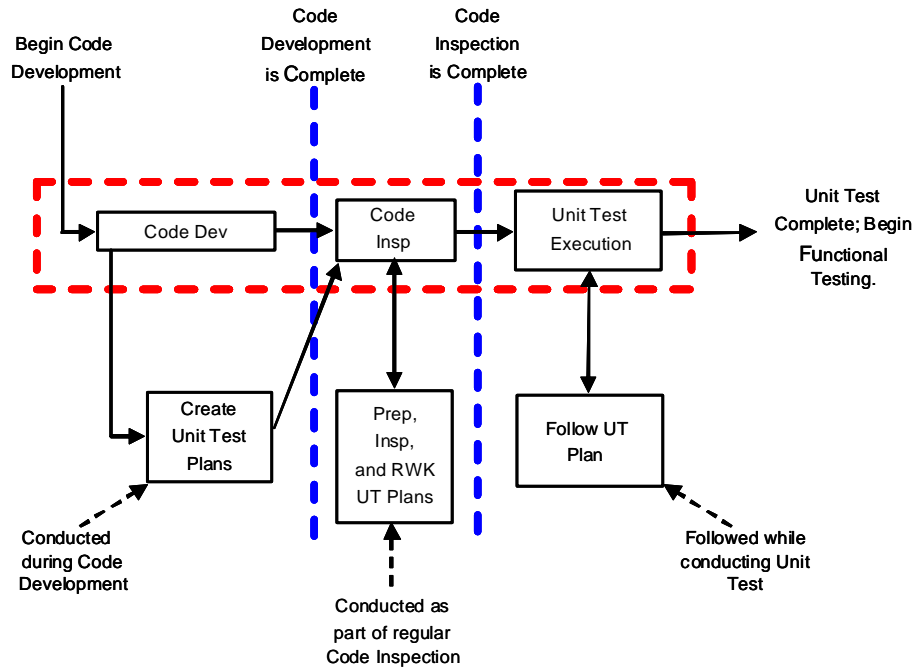
*Figure 14:    Diagram of the To-Be Process: Incorporating the Create Unit Test Plans Process Change*

The Process Tradeoff Analysis Method (PTAM) was followed to develop the PSIM, which was then used to predict the performance associated with the process change. The following project-specific data were obtained for use in the PSIM:

- product size by component
- development productivity rates (hours per KLOC) by phase
- hours devoted to inspections
- error injection and detection rates by phase
- duration data at the component level by phase
- effort allocations for each component
- effort per error costs for inspections and testing

The process definitions for alternative process changes were used as data were gathered and graphical models developed. Figure 15 shows the main phases of the organization's life-cycle process comprising functional specification, HLD, HLD, CODE, UT, test planning/test case preparation, and testing. Figure 16 depicts the As-Is process for CODE and UT, which comprises Code, Code Inspection, Code Rework, and UT. Figure 17 shows the To-Be process for CODE and UT, including the Create Unit Test Plans Process Change. This figure contains the regular code and UT process steps described in Figure 16. In addition, several parallel process steps can be seen.
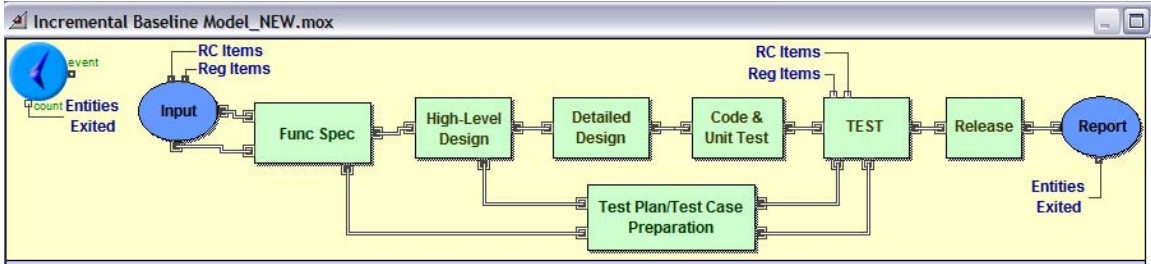
*Figure 15:    Major Phases of the Organizations Life Cycle Development Process*
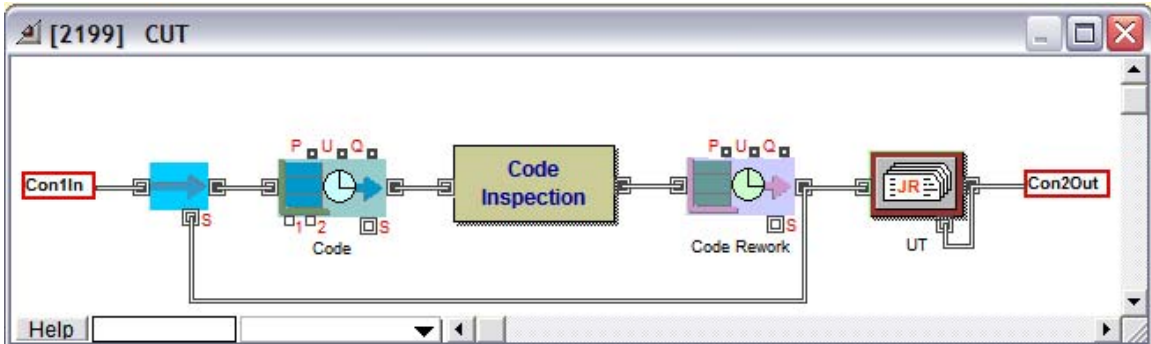


*Figure 16:    As-Is Process for Code and Unit Test*
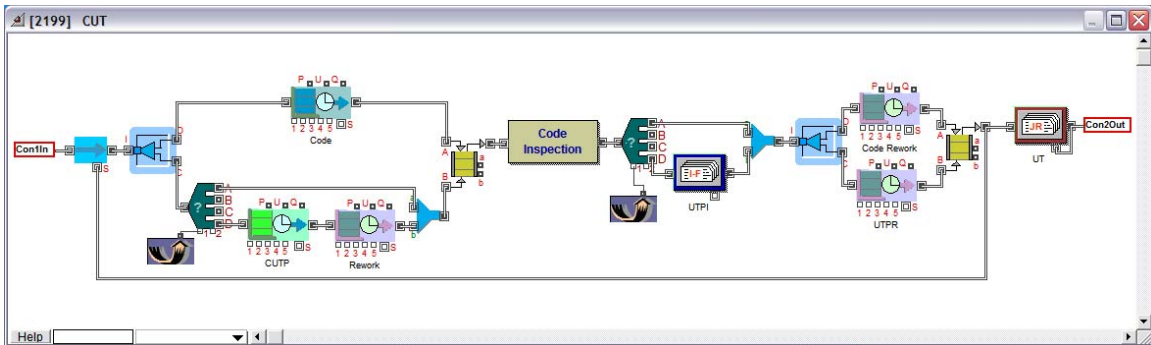


*Figure 17:    To-Be Process for Code and UT Including the Create Unit Test Plans Process Change*

### 3.4.3 Model Application/Results

A multi-attribute decision-making framework (employing utility functions) was utilized to compare the overall performance of the process alternatives. Then, a complete business case was developed that included implementation costs, thereby providing NPV and ROI numbers.

To thoroughly evaluate the process alternatives, extensive sensitivity analysis was done to assess the performance of the processes under a variety of circumstances and to address the above-mentioned questions posed by management.

The results of the sensitivity analysis showed the following:

- The Create Unit Test Plans process offered significant reductions in remaining defects, staff effort to correct field-detected defects, and project duration for this organization on this project. The expected ROI was 56% for a typical 30 KLOC release.
- Compressing UT caused significant increases in schedule (+18%) and effort costs (+8%) due to downstream impacts during later testing phases. Overall product quality was reduced (48% increase in defects).
- Partial implementation of the alternative process was found to be possible. It was recommended that the Create Unit Test Plans process be implemented for complex portions of the code. When implemented in this fashion, the estimated ROI was 72% for a typical 30 KLOC release (compared to 56% for the baseline To-Be case).
- Potential learning curve effects would significantly enhance the performance of the alternative process. Pilot implementations of this process indicated that it would provide a 37% ROI, even under worst-case conditions. With moderate improvements due to learning curve effects, the ROI would almost double to 72%.
- Improving inspections could be a more effective process choice than Creating Unit Test Plans. However, the expected implementation and organizational cost would be much higher. Moreover, improving inspections to rates comparable to those seen in the literature would reduce the expected ROI for the Create Unit Test Plans process change to 17%.
- Reuse of process artifacts, such as the UT Plans, would be likely to yield a significant improvement at a much reduced implementation cost. Reuse of the UT Plans on the next development cycle would provide an overall ROI of 73% (compared to 56% for the baseline To-Be case).

PSIM coupled with the PTAM provided the organization with a structured and quantitative approach for evaluating process alternatives and tradeoff choices. The associated PSIMs were able to estimate process performance in terms of development cost (staff effort), product quality (number of remaining defects), and project schedule (days of task duration); and therefore supported management needs, including

- comparison of process alternatives based on similar process parameters
- go/no-go decisions regarding individual proposed changes
- prioritization and selection from among several proposed changes
- prediction of the cost of quality on a project
- assessment of the potential value of automation and tools for specific projects
- estimation of the impact of a partial implementation of specific process changes
- analysis of the impact of learning curve effects and the reuse of process artifacts

These results were used to develop a solid business case, which was used to obtain management support and "buy-in" for the specific process improvement effort selected.

### 3.4.4 Supporting CMMI Processes and Practices

This case study demonstrates how PSIMs can contribute to fulfilling the following PAs at MLs 2, 3, 4, and 5:

- The planning and estimating practices of PP, IPM, and QPM during which different process tailoring and composition alternatives might be considered
- Organizational Process Focus (ML 3) by identifying process improvements and developing an implementation plan
- Risk Management (ML 3) by identifying process risks and assessing the risk associated with proposed process changes
- Decision Analysis and Resolution (ML 3) by providing decision guidelines, processes, evaluation criteria, and alternative solutions, evaluating alternatives and making specific recommendations
- Organizational Process Performance (ML 4) by helping select processes, measures, setting performance objectives, baselines, and performance models
- Organizational Innovation and Deployment (ML 5) by helping select and deploy incremental and innovative improvements that can measurably improve the organization's processes and technologies. The improvements support the organization's quality and process-performance objectives as derived from the organization's business objectives.
- Causal Analysis and Resolution (ML 5) by aiding understanding of the causes of process problems and evaluating alternative action plans to resolve the problem

## 3.5 USING PROCESS SIMULATION TO ASSESS THE COSTS AND BENEFITS OF APPLYING NEW TOOLS AND TECHNOLOGIES ON A PROJECT

### 3.5.1 Introduction/Overview

New tools and new technologies offer promise for speeding software development tasks, reducing costs, and improving quality along the full development life cycle. Over the years, development organizations have invested heavily in these tools with some success. But there have also been some failures. How can managers determine whether a new tool or technology will benefit their development environment? Under what project conditions would it be beneficial to apply a new tool or technology, and when would it not be beneficial?

PSIM can be used to evaluate new tools and technologies that are being considered. Using PSIM enables an organization to

- plan how a new technology might be applied
- assess the costs and benefits of the new tool or technology
- explore alternative approaches for applying the technology

Using PSIM, an organization can answer the following questions **before** rather than after investing in the technology.

- What is the likely impact of applying new tools and technologies?
- What is the likely economic benefit or value of the tool or technology? What is the ROI?

- When might it be useful and when might it be useless?
- Under what conditions does the tool or technology perform best?
- What performance standards must the tool achieve to have a positive return?
- Are there better ways to apply the tool?

The technology evaluation tool described below is an automated natural language requirements analysis tool called QuARS [Lami 2005b, 2005a]. At the time of this study, the developers of this technology had recently made significant breakthroughs in reducing costs and increasing the effectiveness of this technology. Is the technology now "ready for prime time" on NASA projects? This study sought to address this question.

Software requirements-related defects are the most common and most expensive type of defects to correct. Depending on when this class of defect is found, the cost to find and fix these defects can range between 50-100 times the effort/cost it would have taken to correct the defect in the requirements phase [Leffingwell 2000]. Therefore, it is crucial to detect as many requirements defects as early as possible. The fact that requirements documents are commonly written in natural language makes them prone to errors. There are several human-resource intensive defect detection techniques such as inspection-base techniques and scenario-based review techniques. However, these techniques can be expensive and time consuming. The Quality Analyzer for Requirement Specification (QuARS) is an automated natural language requirements analyzer tool that identifies defects in requirements. QuARS performs expressive analysis on requirements documents and indicates potential defects based on the quality model described in [Lami 2005b, 2005a].

### 3.5.2 Model Description

This case study used the IEEE 12207 Model described at the beginning of Section 3 (see Figure 3). The main life-cycle phases were

- process implementation
- system and software requirements analysis
- software architecture and detailed design
- software coding and unit testing
- software and system integration planning
- integration and qualification testing
- integration and acceptance support

The model includes an IV&V layer that represents the actions of external consultants auditing software artifacts.

The IEEE 12207 process is representative of the life-cycle processes used on large-scale NASA and U.S. Department of Defense (DoD) projects. Predictions made with the model provide similar accuracy to those obtained using COCOMO I (i.e., predictions were within 30% of actual values more than 70% of the time). The PSIM was used to explore the impact of applying the new technology to the project and separately to IV&V.

The QuARS tool was inserted at System Requirements Analysis, Software Requirements Analysis, Concept Verification, and Requirements Verification, as indicated by large arrows in Figure 18. The model was calibrated using NASA project data and industry standard data from Jones [Jones 1991]. The IEEE 12207 Model consists of two layers: 1) Development and 2) IV&V. The development layer represents the systems

and software life-cycle phases based on the IEEE 12207 standard. It comprises nine phases. Each phase consists of one or more process steps. In total, there are 86 steps in the software development process. The IV&V layer represents the activities carried out by external software auditors. This layer consists of five main IV&V phases. Each phase comprises multiple IV&V activities that may be used to verify and validate software artifacts from the corresponding software development phases. The results of this model were validated against the performance data from 12 large-scale NASA projects (with project sizes of 90 KLOC or higher).
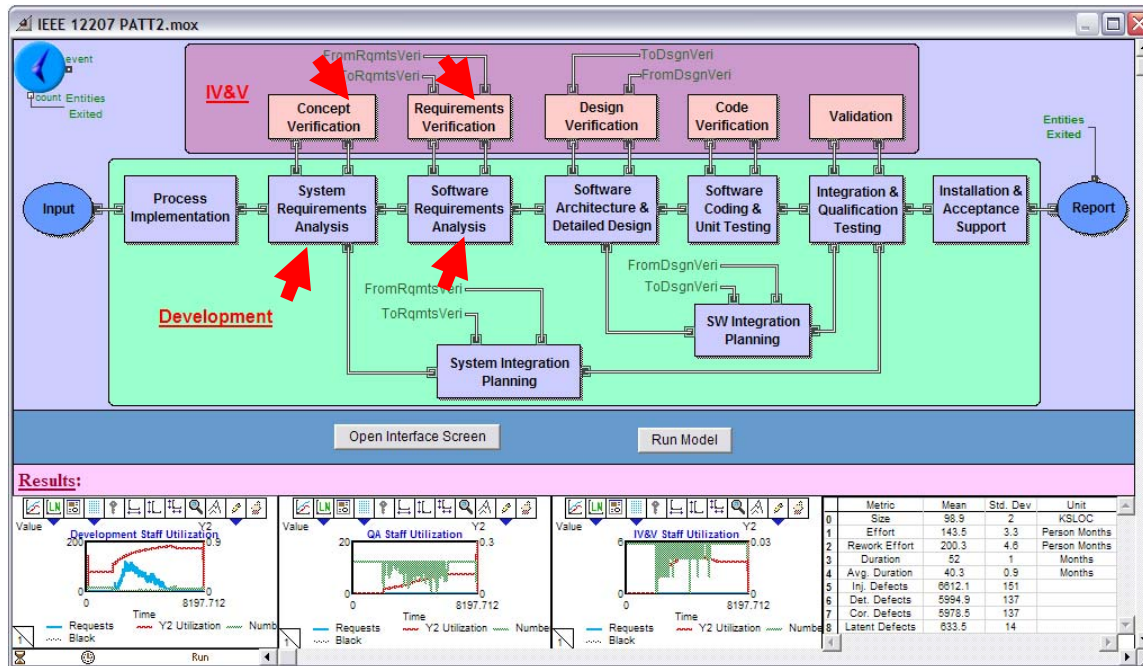


*Figure 18:    PSIM of the IEEE 12207 Systems Development Process
Large arrows indicate potential insertion points for the QuARS
tool that were explored by this study.*

### 3.5.3    Model Application and Results

The general method for using PSIM to assess the impact of new technologies on a project is similar to assessing the impact of a process change. The main steps are

1.   Develop the As-Is PSIM baseline model.
2.   Design each To-Be process scenario and make appropriate changes to the model.[8]
3.   Run each To-Be scenario and conduct sensitivity analysis.
4.   Determine the changes in performance and select the "best" process option.

We discuss each of these main steps as applied to this case study in more detail as follows:

1.   **Develop the As-Is PSIM baseline model**.
     As discussed above, the IEEE 12207 PSIM was used because it is representative of large-scale NASA projects. Moreover, the model was tuned using NASA data to provide representative results. The baseline results from the PSIM are shown in Table 10.

---

[8]    To make each use case section illustrating a particular use of PSIM more "standalone," such descriptions are occasionally repeated from earlier sections.

2. **Design each To-Be process scenario and make appropriate changes to the model.**
   Two To-Be process scenarios were created for this case study.

In this case study, we consider using QuARS during

- quality assurance (i.e., V&V[9]) activities within the project: applying QuARS to analyze the System Requirements, Software Requirements, and then at both phases.
- IV&V activities outside of the project: applying QuARS at Concept Verification, Requirements Verification, and then at both phases.

The key questions that we aim to answer are

- Does using QuARS improve cost, quality, or schedule performance on the project?
- Is QuARS more effective in V&V or IV&V mode?
- What is the amount that the project should be willing to pay for QuARS?

**Assumptions**

To evaluate automated defect detection tools we consider following criteria:

- PD: the probability of detecting faults
- accuracy
- the cost of using the tool
- the probability of false positives

To estimate some key parameters for the model, we utilized empirical study results which were found in reports by Ferguson and Lami[10] to represent QuARS capabilities [Ferguson 2006, Lami 2007]:

- QuARS productivity is 10,000 KLOC per person hour.
- 37% of the requirements defects are QuARS detectable. QuARS defect detection rate is 100% for QuARS-detectable defects.
- Employing QuARS improves the quality of the requirements document, thus the defect detection capability at Requirements inspection improves by 5% to 15% (min = 5%, max = 15%, mode = 10%) if the QuARS detected defects are corrected prior to requirements inspection.
- The cost of training and associated software engineering process group (SEPG) activities is one person-month.

Employing QuARS also provides benefits to other development phases besides the Requirements phase by improving

- clarification of requirements, thus improving design productivity by 5% to 10%

---

[9] Verification and Validation (V&V) activities determine whether development products of a given activity conform to the requirements of that activity and whether the software satisfies its intended use and user needs. This determination may include analysis, evaluation, review, inspection, assessment, and testing of software products and processes.

[10] Lami, G. & Ferguson, R.W. "An Empirical Study on the Impact of Automation on the Requirements Analysis Process," to be published in the *Journal of Computer Science and Technology.*

- engineering design decisions, thus reducing the injection of design defects by 5% to 10%
- test planning and test case generation productivity by 10% to 20%
- the quality of test cases, thus reducing the injection of test case defects by 5% to 15%

**Business Implications of QuARS**

**As-Is Baseline Model Results**

As discussed in the previous section, the IEEE 12207 process model baseline performance was predicted in terms of effort (or cost), duration, and latent defects (or delivered defects). The characteristics of the AS-IS model are as follows:

- The project is 100,000 lines of code.
- The industry standard data [Jones 1991] were used for earned value (percent of effort allocated for each activity) and defect detection rate.
- Organization-specific data were used for productivity and defect injection rates.

The baseline performance for the As-Is process (without using QuARS) is shown in Table 10. The data has been scaled to help protect company confidentiality.

*Table 10:      Baseline Performance for the As-Is Process Prior to QuARS*

|  | Effort incl. IV&V | Effort | Rwrk_Efrt | IV&V Effort | Duration | Avg. Dur | Crctd_Dfcts | Ltnt_Dfcts |
|---|---|---|---|---|---|---|---|---|
| Mean | 71,371.20 | 69,301.61 | 27,404.94 | 2,069.59 | 4,837.69 | 2,423.03 | 6,004.87 | 634.68 |
| Std. Dev. | 1,910.20 | 1,894.25 | 1,037.12 | 246.33 | 195.06 | 92.37 | 227.50 | 24.64 |

**Scenario 1: Applying QuARS in V&V Mode at Different Phases**

In this scenario, changes were made to the model to represent three configurations: 1a) QuARS at the System Requirements phase; 1b) QuARS at the Software Requirements phase; and 1c) QuARS at both phases. Figure 19 shows a flow chart of the As-Is and To-Be processes for configuration 1a) QuARS at the Systems Requirements phase.
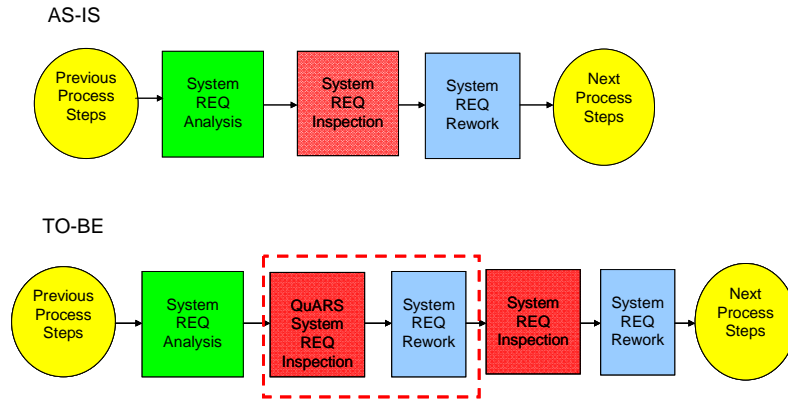
AS-IS

TO-BE

*Figure 19:     Process Flow Chart for As-Is and To-Be Process – 1a) QuARS at Systems Requirements Phase*

Table 11 shows the differences in the model mean for three QuARS V&V configurations compared to the As-Is baseline performance. When QuARS was employed at System Requirements phase, the total effort (including IV&V effort) was reduced by 1,659.07 person hours. Note that a positive value means improvement. The negative numbers in Table 11, indicating that IV&V effort could increase, are not meaningful, as they represent only a very small percentage of overall IV&V effort and are not statistically significant (their p-value is much greater than 0.05). The changes in value for the corrected defects (both positive and negative) also are only a small percentage of the total number of corrected defects, and are not statistically significant.

*Table 11:     QuARS Scenario 1 Performance Comparison to the Baseline*

**Comparison to Baseline**

|  | Effort incl. IV&V | Effort | Rwrk_Efrt | IV&V Effort | Duration | Avg. Dur | Crctd_Dfcts | Ltnt_Dfcts |
|---|---|---|---|---|---|---|---|---|
| **1a) QuARS at Sys Req** | 1,659.07 | 1,669.63 | 1,311.82 | -10.56 | 103.00 | 48.64 | 33.98 | 18.14 |
| p value | 0.00 | 0.00 | 0.00 | 0.87 | 0.05 | 0.05 | 0.56 | 0.00 |
| **1b) QuARS at Sw Req** | 5,141.86 | 5,127.99 | 4,778.59 | 13.87 | 377.28 | 71.50 | -10.12 | 55.12 |
| p value | 0.00 | 0.00 | 0.00 | 0.83 | 0.00 | 0.01 | 0.86 | 0.00 |
| **1c) QuARS at Sys & Sw Req** | 5,267.99 | 5,284.64 | 4,925.64 | -16.65 | 362.00 | 80.63 | -9.89 | 58.54 |
| p value | 0.00 | 0.00 | 0.00 | 0.80 | 0.00 | 0.00 | 0.87 | 0.00 |

One can see that applying QuARS resulted in better overall project performance. In all three cases, the effort expended was lower; the duration was shorter; and the quality (as measured by latent defects) was improved. The effort was reduced because the improved requirements document brought an increase in productivity in subsequent development phases. In addition, QuARS allows development engineers to detect and correct defects early in the process, which resulted in lower rework cost. With better and clearer requirements, the quality of the overall product also improved.

Note that applying QuARS at the Software Requirements phase (1b) yielded a more significant improvement than applying QuARS at System Requirements phase (1a). When QuARS was applied at the Software Requirements phase, the effort decreased by almost 3,500 person hours and the average number of latent defects reduced by more than double (37 defects), as compared to when QuARS was applied at the System Requirements phase. Applying QuARS at both phases resulted in marginal improvement on effort and quality; however, the duration was a bit longer than applying QuARS only at the Software Requirements phases.

We also experimented with the option of applying QuARS before or after requirements inspection. Although we found that applying QuARS after requirements inspection does improve the project performance as

compared to the baseline, the benefit of applying QuARS after a requirements inspection is 10% to 15% lower than when applying QuARS before requirements inspection.

**Scenario 2: Applying QuARS in IV&V Mode at Different Phases**

For this scenario, we examined the impact of QuARS when applied during IV&V activities. Changes were made to the model to represent three different configurations: 2a) QuARS at the Concept Verification phase; 2b) QuARS at the Requirements Verification phase; and 2c) QuARS at both phases. Figure 20 shows flow charts of the As-Is and To-Be processes for configuration 2b) QuARS at Requirements Verification phase.
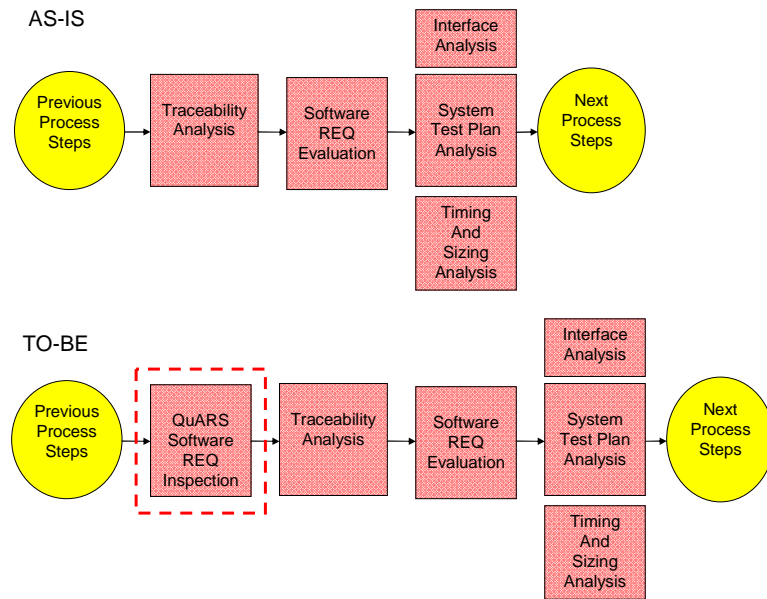


*Figure 20:    Process Flow Chart for the As-Is and To-Be Process – 2b) QuARS at the Requirements Verification Phase*

We made changes to the model to represent each configuration. Table 12 shows the differences in the model for three QuARS IV&V configurations compared to the As-Is baseline performance.

*Table  12:    QuARS Scenario 2 Performance Comparison to the Baseline*

**Comparison to Baseline**

| | Effort incl. IV&V | Effort | Rwrk_Efrt | IV&V Effort | Duration | Avg. Dur | Crctd_Dfcts | Ltnt_Dfcts |
|---|---|---|---|---|---|---|---|---|
| **2a) QuARS at Concept V** | 1,448.16 | 1,679.42 | 1,321.83 | -231.26 | 114.25 | 68.84 | 31.97 | 17.08 |
| p value | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.01 | 0.59 | 0.01 |
| **2b) QuARS at Requirments V** | 2,427.46 | 2,717.04 | 2,340.55 | -289.58 | 190.67 | 64.10 | 18.92 | 28.59 |
| p value | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.01 | 0.75 | 0.00 |
| **2c) QuARS at both** | 2,899.94 | 3,373.50 | 2,975.94 | -473.56 | 236.75 | 97.55 | 10.73 | 35.96 |
| p value | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.86 | 0.00 |

As in Scenario 1, applying QuARS in IV&V did improve project performance as compared to the baseline model for all three configurations. However, the value of QuARS as an IV&V tool is significantly less than the value of QuARS as a V&V tool. The effort decreased by 2% to 4% when we applied QuARS at IV&V mode, while the effort decreased as much as 8% when we applied QuARS at V&V mode. The reason for this is that the secondary effects as discussed in Section 3.5.3. did not emerge in the project when QuARS was employed in IV&V mode.

From the results of these two scenarios, we can conclude that QuARS did add value to the project by reducing effort, shortening project duration, and improving quality. The degree of value added depends on the phase in which QuARS is applied. Applying QuARS in V&V offers more benefits than applying QuARS in IV&V. Applying QuARS at both Systems and Software Requirements phases yield the highest benefit, but the actual sweet spot is to apply QuARS at the Software Requirements phase. In addition, QuARS should be applied before the requirements inspection in order to capture the most benefit.

**Financial Analysis**

In order to weigh the projected benefits received from QuARS against the cost of implementing the tool, we need to convert project performance measures (effort, duration, and quality) to the financial measures. Equations, references and a discussion for doing this financial analysis can be found in Section 3.3 of this report. Several key parameters required for the analysis in this section (which are different from the assumptions made in Section 3.3) are as follows:

- The organization's internal investment rate cut-off (a.k.a. hurdle rate) is 20% annually.
- The cost of development staff is $100 per hour. The cost of IV&V staff is also $100 per hour.
- The cost to correct latent defects after release is 1.5 person-months (or $25,500 per defect).
- There are 170 work hours per month.
- Implementation cost for QuARS is assumed to be incurred at time = 0; development costs can be assessed as a one time cash flow when the project is completed (time = duration); costs to fix latent defects occurs at one year after the project is completed (time= duration + 12 months).
- There is no benefit if the project completes early. Note that this is specific to the organization. Other organizations may gain benefit if the software is released early (i.e., increase in market share/revenue).

Equation (1), shown in Section 3.3 of this report, was employed to make the necessary calculations. Table 13 shows the value of QuARS for the different scenarios discussed earlier in this section. This value may be interpreted as the amount the project would be willing to pay to implement the tool (e.g., for the cost of the tool and training) in order "break even" in a sense.

*Table 13:     Value of QuARS*

| Config. | QuARS Value | |
|---|---|---|
| | Mean | Std Dev |
| 1a) QuARS at Sys Req | $329,350.06 | 41,623.20 |
| 1b) QuARS at Sw Req | $1,012,909.55 | 53,732.26 |
| 1c) QuARS at Sys & Sw Req | $1,094,509.64 | 68,700.92 |
| 2a) QuARS at Concept V | $313,387.99 | 32,630.94 |
| 2b) QuARS at Requirments V | $511,362.33 | 39,002.30 |
| 2c) QuARS at both | $638,714.67 | 50,579.24 |

The probability that the QuARS value is higher than $0 is 100%, which indicates that QuARS helps improve project performance. The probability that the QuARS value is higher than $100,000 is also 100%. This suggests that if the total cost of QuARS implementation is $100,000, the project would gain significantly (between $213,388 and $994,510) should it decide to implement QuARS.

**Conclusion**

The degree of the value added depends on where QuARS is applied. Applying QuARS at the in-project V&V level offers more benefits than applying QuARS externally to the project in IV&V mode. Applying QuARS at both the Systems and Software Requirements phases yields the highest benefit, but the actual sweet spot is to apply QuARS at the Software Requirements phase. In addition, QuARS should be applied before the Requirements Inspection in order to capture the most benefit. The financial analysis shows how one can translate the impact of a new technology into financial value, which makes it easier to decide whether or not to acquire a new tool.

The use of PSIM provided the organization or project manager with specific guidance for identifying distinct project conditions and use cases under which a new technology is likely to be useful or useless. Furthermore, PSIM enables the organization or project managers to set performance benchmarks for vendors of new tools or technologies and to diagnose problems associated with proposed implementations. PSIM can then be used to assess alternative approaches for applying the technology to benefit the organization.

### 3.5.4 Supporting CMMI Processes and Practices

Investing in new technologies is not prudent unless there is a compelling business case for their use. Without such a case, a project manager may not be convinced that he or she should, for example, reallocate scarce resources to implement a new technology or tool on a project. This case study shows how PSIM can be used to evaluate new tools and technologies that are being considered for any project. It also demonstrates how PSIMs can contribute to fulfillment of

- Organizational Process Focus (ML 3) by identifying process improvements and developing an implementation plan
- Risk Management (ML 3) by identifying process risks and assessing the risk associated with proposed process changes
- Decision Analysis and Resolution (ML 3) by providing decision guidelines, processes, evaluation criteria, and alternative solutions; evaluating alternatives; and making specific recommendations relative to process optimization
- Organizational Process Performance (ML 4) by selecting processes, establishing measures, setting specific performance objectives, and establishing baselines and performance models
- Organizational Innovation and Deployment (ML 5) by aiding in selecting and deploying incremental and innovative improvements that can measurably improve the organization's processes and technologies. The improvements support the organization's quality and process-performance objectives as derived from the organization's business objectives.
- Causal Analysis and Resolution (ML 5) by aiding understanding of the causes of process problems and evaluating alternative action plans to resolve the problem

### 3.6 USING PSIM TO SUPPORT QUANTITATIVE PROCESS MANAGEMENT, MONITORING, AND CONTROL

#### 3.6.1 Introduction/Overview

Tracking projects and monitoring progress is a vital management function that profoundly affects all projects. Quantitative metrics and status information show where a project has been. However, to move up to ML 4 of CMMI and to truly control projects against their schedules requires more forward-looking approaches that

- keep overall project performance goals in focus
- provide early indicators that a project is going off track
- predict the potential impact of corrective actions *before* they are implemented
- help to select the best corrective action alternative

PSIM provides key capabilities that assist in monitoring and controlling projects. When used in conjunction with database applications, PSIMs can store project snapshot data and then utilize this information to achieve more accurate, up-to-date predictions of overall project performance. The updated predictions can then be compared to targets preset by management.

If the project is significantly off track, the PSIM can be used to explore the impact of alternative corrective actions to bring the project back on track. The PROMPT method illustrates the use of PSIM in this fashion and integrates timely metrics data with simulation models of the software development process, as described by Figure 21.

The PROMPT method is designed to be an iterative, ongoing, process improvement framework similar to—but refining—Deming's Plan-Do-Study-Act (PDSA) Cycle. PROMPT augments Deming's work by utilizing in-process data and quantitative models to support the Planning, Studying, and Action phases of the Deming PDSA Cycle. Through use of models that predict process performance, PROMPT augments PDSA in the following ways:

- planning: The model supports the planning phase by guiding the selection of process actions and decisions.
- doing: Using the model guides project execution.
- studying: By using timely metrics information to update the PSIM, one can employ the model to study the progress of the plan.
- action: When corrective action is deemed necessary, the model is used to identify corrective actions that can be deployed to bring the project back on track.

PROMPT uses outcome-based specification limits (OBSLs), which represent acceptable ranges or specification limits for project performance that are set by management. Although they are used in a similar manner, OBSLs are distinctly different from traditional statistical process control (SPC) limits in what they represent and how they are set.
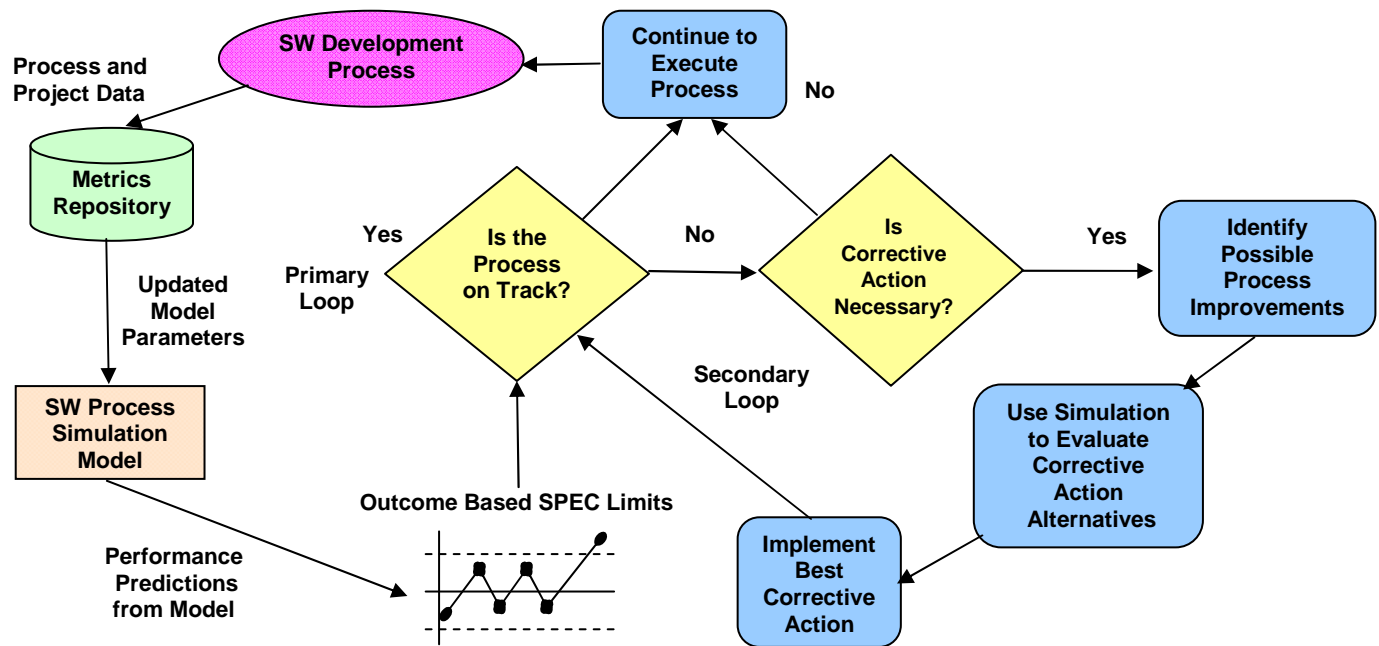
*Figure 21:    Using PSIM in a Feedback Loop to Monitor and Control a Software Development Process*

With traditional SPC models, control limits are derived based upon past process performance. Thus, control limits are linked to process consistency rather than process performance. Because software projects can have high degrees of variability, managers need to get involved at the point when the project is going off track **regardless of whether the project is performing consistently.**

OBSLs identify the targets for project performance and the acceptable ranges of performance for the overall project. OBSLs are used to monitor the process and trigger corrective action when the PSIM estimates based on current process performance fall outside of the targeted outcome range.

Through the PROMPT method, the simulation model is used to map current performance, (which is reflected in the timely metrics that are collected in the projects' or organization's measurement repository), to probable project outcomes. If the predicted performance deviates too much from the desired project outcomes, corrective action may be taken.

### 3.6.2    Model Description

This case uses the incremental life-cycle model described at the beginning of Section 3 (see Figure 4). Figure 22 shows where experienced personnel could be augmented as well as where inspections and tests could be added to the process to address the client's questions.
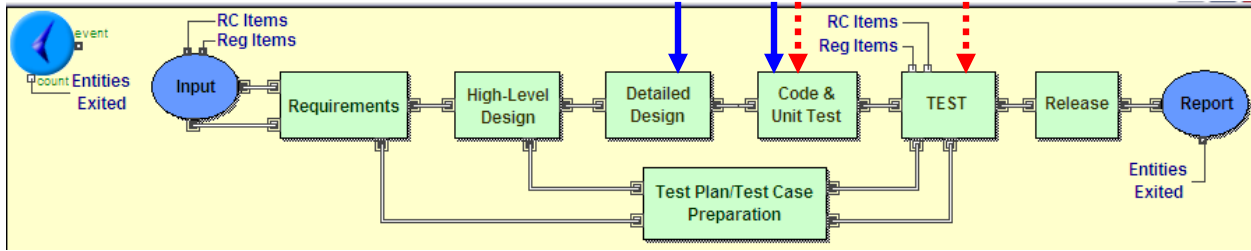
*Figure 22: Incremental Life-Cycle Process Where Potential Changes Were Identified (Solid Arrows Represent Potential Points for Personnel Upgrades; Dashed Arrows Indicate Places for Additional V&V Activities)*

### 3.6.3 Model Application/Results

The information provided in this report has been modified to protect the confidentiality of the client for this project. Thus, the numbers shown may not appear to be fully realistic or consistent.

The specific application was built to fulfill a contract to provide a major enhancement (approximately 32 KLOC) to an existing system. The enhancement consists of modifying six computer software configuration items (CSCIs), with modifications to each CSCI ranging between 4 and 8 KLOC. Within the PROMPT framework, a PSIM was used to estimate cost (project effort), schedule (duration), and quality (delivered defects) for the proposed project using past project data augmented with updated estimates for productivity, earned value, defect injection and detection, and project size (among other parameters). The estimates were used to set the performance targets by which the project would be evaluated.

For this project, the software development firm used a modified Waterfall Development Process with the lifecycle phases of Requirements, Preliminary Design, Detailed Design, Coding, Unit Test, Integration Test, and System Test. Inspections were conducted for each phase: Requirements, Preliminary Design, Detailed Design, and Coding.

First, the PSIM was used to provide probability distributions for the values of each performance measure, which were verified to be normally distributed. The means and standard deviations for the baseline process are shown in Table 14. Again, due to company confidentiality requirements, the numbers shown are based on the modified data set and are different from the numbers obtained in the actual analysis.

*Table 14: Model Estimates of Project Performance*

|  | Mean | Standard Deviation |
|---|---|---|
| Total effort (cost) in person months | 418.5 | 5.54 |
| Project duration (schedule) in months | 26.2 | 2.03 |
| Number of remaining defects (quality) | 77.4 | 3.68 |

OBSLs were determined from the standard deviations in Table 14. Four color distinctions were used, where blue is the highest rating and deemed to be excellent. It indicates that the predicted performance measure of interest is within 5% of the target value. Green is the second highest and deemed to be good performance. It indicates that the measure of interest deviates more than 5% but remains within 15% of the target value. Yellow is marginal performance and indicates that the measure of interest deviates more than 15% but remains within 30% of the target value. Finally, red indicates poor or unacceptable performance and that the measure of interest deviates more than 30% from the target value. Other color-coding schemes and specification limits or thresholds can be used. The performance limits for the project are shown in Table 15.

*Table 15:    Project Performance Targets and Limits*

| | Target | Blue Limits | | Green Limits | | Yellow Limits | | Red Zone | |
|---|---|---|---|---|---|---|---|---|---|
| | (From Table 1) | Upper (+5%) | Lower (-5%) | Upper (+15%) | Lower (-15%) | Upper (+30%) | Lower (-30%) | Above (>+30%) | Below (<-30%) |
| Total effort (cost) in person months | 418.5 | 439.4 | 397.6 | 481.3 | 355.7 | 544.0 | 292.9 | 544.0 | 292.9 |
| Project duration (schedule) in months | 26. 2 | 27.5 | 24.9 | 30.2 | 22.3 | 34.1 | 18.4 | 34.1 | 18.4 |
| Number of remaining defects (quality) | 77.4 | 81.2 | 73.5 | 89.0 | 65.8 | 100.6 | 54.2 | 100.6 | 54.2 |

Next, we evaluate the baseline likelihood that the project will achieve a blue or green rating (see Table 16).

*Table 16:    Probability of Achieving Blue/Green Performance for the Baseline Process*

| | Probability of Blue Rating* | Probability of Blue or Green Rating** |
|---|---|---|
| Total Effort (Cost) in Person Months | 99.9% | 99.9% |
| Project Duration (Schedule) Months | 74.1% | 97.4% |
| Remaining Defects (Quality) | 70.7% | 99.8% |

\*    Probability performance is within 5% of target

\*\*    Probability performance is within 15% of target

Table 16 shows the predicted performance of the project prior to the start. If the baseline process had not obtained a high enough likelihood for achieving the desired outcomes, then changes would have been made to the process, or staff would have been assigned differently to increase the likelihood of acceptable performance. (This is similar to the situations described in CMMI QPM 1.2 and IPM 1.1-1.2.)

Once the project is underway, the PROMPT method calls for using process metrics to update model parameters and assess the project trajectory.

To incorporate the updated project information from the project repository into the model, previously estimated parameters are replaced with actual data from the life-cycle phases of the project that have been completed. This improves the accuracy of the model and reduces variability of the estimated project outcomes because instead of using stochastic parameters for the phases of the project that have been completed, actual data are now available.

In this example, instead of the expected 67 defects detected during preliminary design, 78 were actually detected (a 15.5% increase). Further investigation revealed that although the project was staffed by developers with more than five years experience with the firm, over half of them were new to developing internet applications; thus, the higher-than-expected defect levels are likely to continue. In a real situation, one would expect that differences in productivity, effort expended, and other parameters would also be observed, giving further evidence of potential problems. This example, however, is intentionally limited to the observed increase in defects during preliminary design. Consequently, the mean and standard deviation of defects injected within the model were increased by 10%.

The predicted outcomes for the parameter changes described in the previous paragraph are shown in Table 17.

*Table 17:     Project Performance Using Observed Defect Levels for Requirements and Preliminary Design*

| | Target Values | Model Predictions | | Probability of Blue Rating* | Probability of Blue/Green Rating** |
| --- | --- | --- | --- | --- | --- |
| | | **Mean** | **Standard Deviation** | | |
| Total effort (cost) in person months | 418.5 | 428.3 | 10.6 | 85.2% | 99.9% |
| Project duration (schedule) in months | 26.2 | 28.1 | 3.9 | 43.8% | 69.8% |
| Remaining defects (quality) | 77.4 | 81.7 | 9.8 | 27.9% | 71.7% |

\*   Probability performance is within 5% of target

\*\* Probability performance is within 15% of target

Table 17 shows the target values, along with predicted performance levels obtained from the model using the updated parameter values. The updated probability of achieving blue or green performance for all performance measures is also shown. As can be seen, the new performance levels are not acceptable, given management's goal of having at least a 90% probability of achieving the blue or green performance for all performance measures. Consequently, corrective action was taken.

Using the PSIM, a variety of potential process changes were explored, such as bringing on expert software engineers to help with development and providing additional testing. The results of these corrective actions were compared to the OBSLs to determine if one or a combination of the alternatives would enable the project to achieve the desired level of performance.

Management selected and implemented the corrective action (bringing in expert software engineering consultants) and monitored the process to ensure improvement.

Next, PROMPT will be used again to take a project data snapshot at the end of the detailed design phase. Model parameters will be updated with actual observations from detailed design. The feedback cycle would then be repeated to determine whether the project was performing to the desired level.

### 3.6.4     Supporting CMMI Processes and Practices

Combining metrics and predictive models, rather than using metrics alone, allows for a more comprehensive performance picture of the project. Moreover, the predictive models can support managers as they attempt to re-plan and make process tradeoffs to bring a project back on track.

The PROMPT method using PSIMs squarely addresses and strongly supports the Quantitative Project Management and Organizational Process Performance PAs at CMMI ML 4, in particular, QPM 1.4. Moreover, the PROMPT method provides a medium for Causal Analysis and Resolution (CMMI ML 5 PA). PROMPT is also directly related to and supports the Decision Analysis and Resolution, Integrated Project Management, and Organizational Process Focus PAs at CMMI ML 3. It also strongly supports Risk Management (CMMI ML 3 PA) for process-related risks. At CMMI ML 2, Measurement and Analysis and Project Planning also are supported through use of this approach.

### 3.7 USING PROCESS SIMULATION TO OPTIMIZE THE PROCESS ON A PROJECT

#### 3.7.1 Introduction/Overview

Optimizing a process is different from looking at go/no-go decisions on specific process improvement opportunities. When optimizing a process, one typically evaluates many different process options. Although this can take some time and a substantial number of simulation runs, with PSIM this can be accomplished in a few days to a week (provided the baseline PSIM has already been built). Compare this effort with running pilot studies or controlled experiments, which would take months and be prohibitive in terms of cost, effort, and schedule!

One aspect of software development that is very amenable to optimization is software QA (what CMMI refers to as verification and validation). This is an especially critical issue for large-scale systems development and development projects involving "systems of systems." Typically, these projects are so large and varied that different types of development are concurrently performed on these projects—new development, heritage, COTS, reuse, open source, autogenerated code, and so forth. Using the same quality assurance policy for the different types of development activities does not make sense. Furthermore, different parts of a system have different potentials for defects and different consequences for those defects. What is the best QA strategy for a project having multiple types/modes of development? How can the QA strategy be optimized to deliver the highest quality at the lowest cost? What is the impact of changing QA policies during project execution? How can organizations plan for this? What have organizations done?

PSIM has been used to evaluate alternative QA strategies for large-scale, complex projects. PSIMs can be created to trigger specific QA activities based upon specific characteristics of the work products being reviewed. For instance, specific QA activities could be performed only on highly complex, historically error-prone, or high-consequence work products. PSIM can be used to activate different QA strategies for portions of the project that undergo different types of development (e.g., reuse, new development, COTS, open source). Then, given these different QA policies designated for different parts of a project, the QA strategy for the entire project can be optimized.

To illustrate this approach, three specific applications that were completed recently or are currently underway within commercial and government organizations are described:

1. Commercial Organization – End-User Software

   A leading commercial software development firm has ongoing development activities in multiple countries. PSIM has become the tool of choice for planning process changes and analyzing process tradeoffs. Management asked that the PSIM be used to look at the organization's QA activities to drastically reduce schedule while maintaining or improving quality of the product. Management recognized that only limited improvements could be made on the current project but wanted recommendations. In addition, management wanted recommendations for higher impact opportunities for the next release.

2. Commercial Organization – Embedded Software

   This leading development organization creates embedded software using a product-line software development process across two main locations (one in Europe, one in Asia). The organization's development process handles a high number of concurrent projects with significant reuse, new development, and autogenerated code. The goal was to optimize the QA strategy for the organization's

development group when developing reusable components and customized applications for specific customers.

3.  Government Organization

    Software IV&V, as practiced at the NASA IV&V Facility, is a well-defined, proven, systems engineering discipline designed to reduce risk in major software systems development. NASA practices over 40 different IV&V techniques, and research is ongoing to develop better methods and tools to support IV&V. Certain questions arise, however: What IV&V techniques and technologies should be applied to a given project and in what order or combination? What is the optimal mix of IV&V techniques for a given process and a given project's risk profile? PSIM is being used to support NASA IV&V managers as they plan IV&V for new and existing NASA projects. Thus, NASA will be able to assess the best allocation of IV&V resources for a given project. This work contributes to NASA mission assurance and success by making recommendations as to how V&V and IV&V technologies should be deployed on a given project.

### 3.7.2        Model Description

The incremental PSIM presented earlier (Figure 4) served as the foundation for this model. Figure 24 shows the IV&V module management interface screen for this model. This screen provides a list of IV&V techniques that can be applied to a project. These techniques can be turned on or off in combination as desired to optimize the process. The numbers in Figure 24 represent the criticality level of work products needed to trigger the IV&V technique. The criticality level is rated according to how error prone the work product is and the consequence (i.e., impact) were a defect to occur due to a defect in that work product. The rating of each work product is determined using a confidential assessment approach developed by NASA. The scale is 1 to 5 with 5 being the highest level.

Changes in the IV&V configuration will impact project results either significantly, modestly, or insignificantly. Without PSIM, it can be very difficult to analyze these impacts, much less to determine the best configuration.

### 3.7.3        Model Application and Results

When an existing PSIM serves as a starting point, only a few days of effort are required to identify the changes needed to optimize a process and provide substantial savings to an organization. The PSIM can be used not only to optimize a project's QA strategy, as discussed earlier, but to optimize the entire process. The key is to identify all process options and then use the PSIM to assess the impact of each option on overall project performance. Below is a list of steps for using PSIM to optimize a process.

Step 1: Identifying the Process Options
The list of process options  is the list of IV&V techniques (more than 30 in all). The points in the process at which to employ each option are indicated by the column titles the interface shown in Figure 24.

| | Concept Verification | | Requirements Verification | | Design Verification | | Code Verification | |
|---|---|---|---|---|---|---|---|---|
| | Consequence | Error Potential | Consequence | Error Potential | Consequence | Error Potential | Consequence | Error Potential |
| Management and Planning of Independent Verification and Validation | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Issue and Risk Tracking | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Final Report Generation | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| IV&V Tool Support | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Management and Technical Review Support | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Criticality Analysis | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Identify Process Improvement Opportunities in the Conduct of IV&V | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Reuse Analysis | 3 | None | | | | | | |
| Software Architecture Assessment | 3 | None | | | | | | |
| System Requirements Review | 3 | 4 | | | | | | |
| Concept Document Evaluation | 4 | 5 | | | | | | |
| Software/User Requirements Allocation Analysis | 4 | 5 | | | | | | |
| Traceability Analysis | 4 | 5 | | | | | | |
| Traceability Analysis - Requirements | | | 2 | 4 | | | | |
| Software Requirements Evaluation | | | 3 | 4 | | | | |
| Interface Analysis - Requirements | | | 4 | 3 | | | | |
| System test Plan Analysis | | | 3 | None | | | | |
| Acceptance Test Plan Analysis | | | 5 | None | | | | |

*Figure 23: Partial List of IV&V Techniques That Can Be Selected Using the NASA PSIM Depicted in Figure 3*

Step 2: Modeling Each Option Using the PSIM

Process options can be represented in the PSIM by using unique process steps, unique parameters, or a combination of both. In the PSIM in Figure 3, each IV&V technique listed in Figure 24 has its own process block and set of parameters.

Step 3: Obtaining Model Results

When all process options have been identified, the PSIM can be used to estimate the impact of each option on overall project performance. Most PSIMs can report results for local impacts as well as impacts at the project level. For instance, the PSIM can be used to estimate the change in appraisal costs, rework costs, pretest costs, and so forth if this is of interest to management. Table 18 shows a table that incorporates rework costs along with the overall project performance statistics.

Step 4: Comparing Alternatives

Most PSIMs provide performance results in terms of multiple dimensions of performance. Typically models report performance in terms of development cost (effort), product quality (remaining defects), and project schedule (duration). Other performance measures such as functionality, requirements volatility, and project risk can also be assessed.

Table 18 shows a management tradeoff table ranking five IV&V options and the NPV of each option over the baseline default policy. Note that these figures are hypothetical, as the actual numbers are confidential. Figure 24 shows the specific configuration for option 1 in Table 18.

*Table 18:* *Management Tradeoff Table Ranking the Top Five IV&V Options in Terms of their NPVs Relative to the Baseline Default Policy*

| | Option | Effort (PM) [Dev + Rework] | Project Duration (Months) | ▲Revenue due to ▲Duration | Total Injected Defects | Corrected Defects | Escaped Defects | Rework Effort for Field Defects | Implementation Costs | NPV ($) | ROI (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Baseline policy (BP) | 413.45 | 28.36 | n/a | 6,297 | 5,682 | 615 | 461 | $0 | n/a | n/a |
| 1 | BP + full concept verification–all work products | 411.03 | 28.45 | $0 | 6,297 | 5,687 | 610 | 457.65 | $77K | $18K | 64% |
| 2 | BP + full requirements verification–all work products | 408.38 | 28.21 | $0 | 6,298 | 5,694 | 604 | 453.15 | $180K | $33K | 48% |
| 3 | BP + full design verification–all work products | 410.95 | 28.18 | $0 | 6,299 | 5,699 | 600 | 449.84 | $160K | $36K | 39% |
| 4 | BP + full code verification–all work products | 412.75 | 28.16 | $0 | 6,298 | 5,692 | 606 | 454.46 | $81K | $17K | 35% |
| 5 | BP + full validation–all work products | 414.42 | 29.93 | $0 | 6,299 | 5,698 | 601 | 450.37 | $114K | $7K | 23% |

For optimization, the multiple dimensions of performance shown in Table 18 were combined into a single decision statistic to measure the improvement each option delivers over the baseline. Frequently management prefers to receive financial measures to rank process options. NPV is often recommended as the main ranking criteria for use with financial measures. Grant and colleagues and Harrison and colleagues provide further discussion of using financial measures for ranking investment options [Grant 1990, Harrison 1999]. Once ranked, the best option or a set of attractive options is identified. Management often has additional concerns beyond the financial performance for selecting one option over another. All of these considerations must be taken into account when choosing the final option.

The specific results of using PSIM for the commercial organization developing end-user software involved the creation of a series of scenarios that enabled the project manager to identify an optimal mix of inspections and testing. The optimal mix entailed inspecting those portions of the application with a high-risk profile and adjusting the subsequent testing based upon the risk profiles (which were revised for those portions of the product that were inspected).

The commercial organization developing embedded software was enable through PSIM, to determine an optimal inspection strategy for the different product types that reduced both the cycle time and the cost of the development process.

In the government (NASA) example, IV&V strategies for different types of projects were identified.

Note that the examples described above used a brute-force (exhaustive) optimization approach. Users defined a set of feasible options and the PSIM identified the optimal option or set of optimal options from the set of feasible options. To increase efficiency and coverage, we recommend design of experiments coupled with simulation automation tools.

Design of experiments (DOE) is a well-known method for maximizing the information gained from each run in a set of experiments [Ermakov 1995, Law 2000]. Using DOE to vary model parameters may significantly reduce the number of runs and thereby speed up the analysis. Describing DOE and how it works is beyond the scope of this report. Experimental design is also expedited by a number of statistical packages, such as Minitab and SPSS, which have built-in modules that create experimental designs. These packages go further to create an external experimental file that can be used to automatically run the simulation. For example, Minitab creates a file that may be used in conjunction with Extend to automatically run the experimental design created. The results of the simulation runs can then be exported from Extend back into Minitab for further analysis.

| Activity | Concept Verification Consequence | Concept Verification Error Potential | Requirements Verification Consequence | Requirements Verification Error Potential | Design Verification Consequence | Design Verification Error Potential | Code Verification Consequence | Code Verification Error Potential |
|---|---|---|---|---|---|---|---|---|
| Management and Planning of Independent Verification and Validation | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Issue and Risk Tracking | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Final Report Generation | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| IV&V Tool Support | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Management and Technical Review Support | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Criticality Analysis | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Identify Process Improvement Opportunities in the Conduct of IV&V | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Reuse Analysis | 1 | 1 | | | | | | |
| Software Architecture Assessment | 1 | 1 | | | | | | |
| System Requirements Review | 1 | 1 | | | | | | |
| Concept Document Evaluation | 1 | 1 | | | | | | |
| Software/User Requirements Allocation Analysis | 1 | 1 | | | | | | |
| Traceability Analysis | 1 | 1 | | | | | | |
| Traceability Analysis - Requirements | | | 2 | 4 | | | | |
| Software Requirements Evaluation | | | 3 | 4 | | | | |
| Interface Analysis - Requirements | | | 4 | 3 | | | | |
| System test Plan Analysis | | | 3 | None | | | | |
| Acceptance Test Plan Analysis | | | 5 | None | | | | |

*Figure 24:*     *NASA IV&V PSIM Management Interface Indicating the Changes Required to Implement Option 1 Shown in Table 18*

*The dashed-line rectangle indicates that the Requirements IV&V Activities are now applied on the entire project rather than to only selected CSCIs that had a high-risk classification.*

### 3.7.4 Supporting CMMI Processes and Practices

This case study describes how PSIM can be utilized to optimize the process for a given project and demonstrates how PSIMs can contribute to fulfilling the following process areas at MLs 2, 3, 4, and 5:

- Organizational Process Focus (ML 3) by identifying an optimal process within the set of options considered for implementation and deployment
- Risk Management (ML 3) by identifying process risks and assessing the risk associated with proposed process changes
- Decision Analysis and Resolution (ML 3) by providing decision guidelines, processes, evaluation criteria, alternative solutions, evaluating alternatives, and making specific recommendations relative to process optimization
- Organizational Process Performance (ML 4) by helping select processes, measures, specific performance objectives, baselines, and performance models
- Organizational Innovation and Deployment (ML 5) by helping select and deploy incremental and innovative improvements that can measurably improve the organization's processes and technologies. The improvements support the organization's quality and process-performance objectives as derived from the organization's business objectives.

## 3.8 USING PROCESS SIMULATION TO SUPPORT TRAINING

### 3.8.1 Introduction/Overview

*A seeker of truth comes before a Zen master, "Oh master, what is the secret of life, the secret we have all been looking for, the secret of happiness and contentment?"*

*The master replies "Good judgment."*

*The student, happy to hear this, responds, "Ah, good judgment. Well said master. Well said. And how does one acquire 'good judgment'?"*

*The master replies, "Experience."*

*The student questions the master further, "And how does one obtain 'experience'?"*

*The master, always economical with words, completes the lesson, "Bad judgment."*

*—Source: Unknown*

How is knowledge of good software engineering principles transmitted? How do new project managers learn how their decisions impact the performance of a project? The answer often given is "experience." But at what cost is this experience gained? How much experience is required for a manager to gain a broad view of all phases and activities for complex development projects? How many projects must a manager complete to gain familiarity with the variety of circumstances and interacting factors that impact large-scale development projects? How many more projects are required for a manager to understand how to steer projects successfully through these circumstances?

Managers and developers are often very clear about their own roles and responsibilities, yet they remain unaware of the activities of others working on different aspects of the project. Furthermore, they are often unaware of how specific factors and early project decisions are likely to impact later phases of the project and its overall performance.

As we have shown in the previous sections, PSIM is a great tool for demonstrating the impact of process decisions and illustrating some of the important interacting factors that strongly impact overall project performance. Moreover, when one is assessing the impact of process improvements or applying new tools or technologies, evaluating QA strategies, or addressing other issues, PSIM does an excellent job in evaluating the tradeoffs. A PSIM is also a great tool for bringing lessons-learned repositories alive and for training people on the development process. In short, PSIM can be used to clarify cause-and-effect relationships between project-management decisions and project results. PSIMs can provide a shortcut for managers to gain the experience required to have good judgment, or at least better judgment, when they must address critical project decisions, thereby helping to mitigate risk.

### 3.8.2        Model Description

Training using PSIMs takes three main forms:

1. using an existing PSIM as a training medium for managers and software engineering process group (SEPG) personnel to learn about possible impacts of their decisions
2. using PSIMs to augment courses in quantitative project management and decision making such as CMMI or Six Sigma
3. having students create PSIMs to gain a deeper understanding of project dynamics within an organization

**Using Existing PSIMs as a Training Medium**

Using PSIMs is a very effective and economical way to provide a shortcut for project managers, SEPG members, and software engineers to gain perspective and experience regarding project causes, dynamics, and interactions that impact project performance. The case studies and scenarios described in Section 3 of this technical report provide examples of high-value ways in which PSIMs have been utilized within a variety of commercial and government organizations. The use of PSIMs has provided significant savings and reduced risk for the project manager and has helped provide a broader view and perspective regarding the entire development project. This capability to provide a full view spanning multiple project phases and a wide variety of interacting issues makes PSIM a powerful tool to support decision making and training.

The case studies provided in Section 3 vary from qualitative work flow management to project cost estimation, process tradeoff analysis, new technology adoption, quantitative project management, and project replanning to QA strategy evaluation, process optimization and global software development and supply chain analysis. Training can be done using PSIMs for all of these situations and more. Moreover, many companies have lessons-learned repositories, and PSIMs can be used to animate and enliven them.

**Using PSIMs in Conjunction With Other Training Such as CMMI or Six Sigma Classes**

Software engineering training courses dealing with quantitative project management often use the case study approach to train students. In these courses, instructors present students with a number of different project scenarios and data sets that reflect the problem being studied. The instructor

then typically asks the students to analyze the data to determine the underlying issue. Students evaluate the data and identify possible causes. The students then make different recommendations for resolving the issue. How can these students **see** the impact of their recommendation? How can they understand whether the training has enabled them to use good judgment? Using a PSIM in conjunction with training classes such as these enables student to "close the loop" on their training and make the final connections to understand whether their recommendations worked or not.

**Having Students Create PSIMs to Gain Deep Insights**

Navaro, as well as Birkhoelzer and colleagues, state that using PSIMs can provide useful insights for managers [Birkhoelzer 2005, Navaro 2005]. But for reaching the next level of understanding of a development project and its various interacting factors, actually creating the PSIM can be very useful training. PSIMs can support process understanding and learning of the workflows, informational exchanges, work products, inputs and outputs, and other process details. Figure 25 shows a top-level screen shot of Navaro's Sim SE environment that students use to create PSIMs. Navaro and van der Hoek describe the experiences of the research group at UCI.

### 3.8.3    Model Application/Results

Several empirical studies have been done to show how PSIM can be used to improve management decision making and learning using computer science students [Pfahl 2004]. The studies focused on improving students' reactions and understanding to help them make better decisions in the areas of project planning and control. These studies showed that quantitative models and PSIMs consistently enable students to better understand important project issues, project dynamics, and knowledge of typical project behavior patterns.

### 3.8.4    Supporting CMMI Processes and Practices

PSIM can be used to create training materials and to deliver content regarding lessons that touch many of the PAs within CMMI, including SG1, SP 1.4 Establish Training Capacity and SG2, SP 2.1 Deliver Training for the CMMI level 3 PA for Organizational Training, and the Generic Practice 2.5 to Train People.
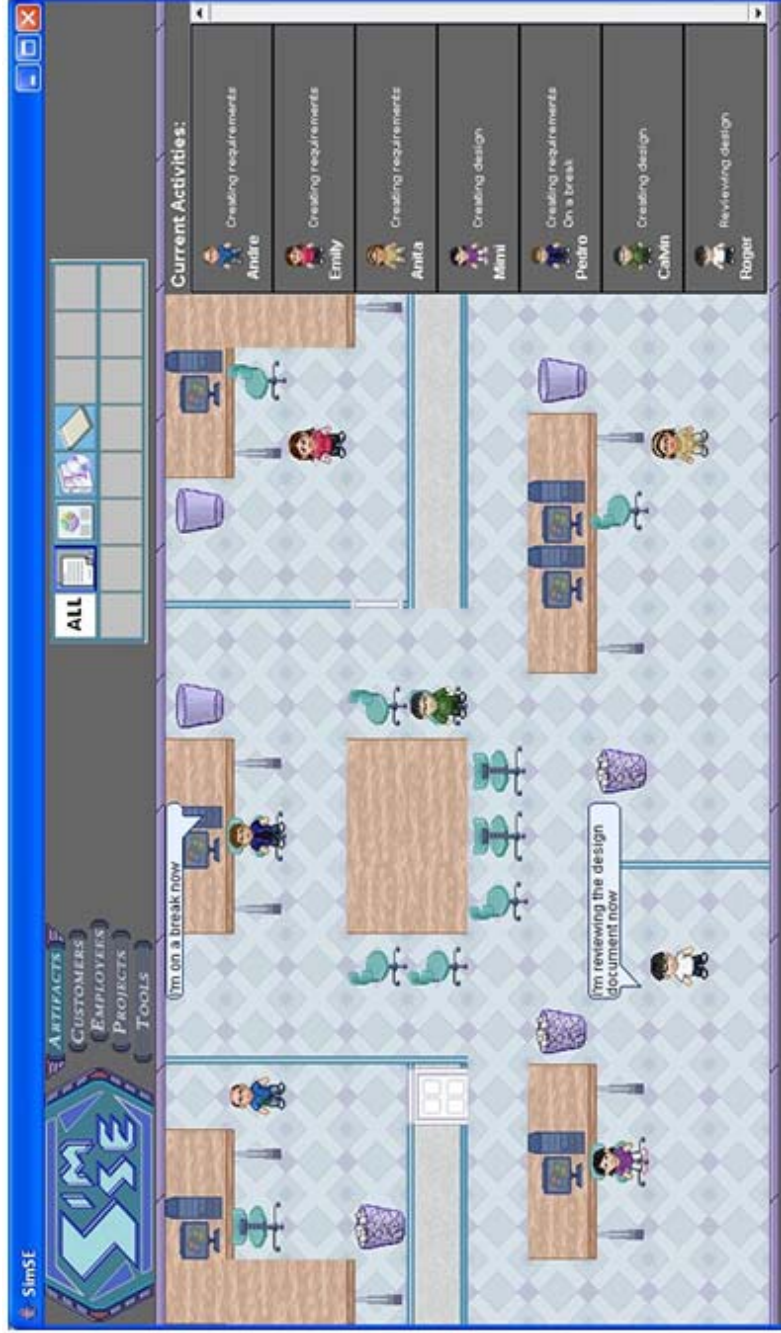
*Figure 25:    Screen Shot of Navaro's Sim SE Environment that Students Use to Create PSIMs*

### 3.9  USING PSIM TO STUDY GLOBAL SOFTWARE DEVELOPMENT

#### 3.9.1        Introduction/Overview

Although several companies have reported success using global software development (GSD), simply deploying GSD as if it were an ordinary project is unlikely to yield positive results because it poses a number of challenges and difficulties as well as significant potential benefits. To be successful, companies must adapt and improve their processes to support this kind of development. Strong project planning and management are also required, along with new methods, processes, tools, tracking, and controls.

As described previously in this report, discrete PSIM has been used to address a variety of issues in software development projects ranging from strategic management, project planning and control, and process improvement to training and understanding. However, to effectively model GSD projects, a hybrid simulation model combining both system dynamics and discrete event models is needed. SD models can easily incorporate continuous factors and their interactions, including those related to communication, coordination, cultural issues, learning curve, changing staff levels, and dynamically varying productivity. Discrete event simulation captures the actual process level details and can represent each work product of the development process as being unique through the use of attributes such as size and complexity. Thus, it provides the ability to explicitly represent the process structure and mechanisms used to transfer work products and to coordinate activities. These two paradigms complement each other; thus, a hybrid model is better able to capture the actual development process executing within a continuously changing project environment.

This section describes just such a hybrid PSIM, used to model GSD. The work described in this section was adapted from that of Setamanit and colleagues [Setamanit 2006].

#### 3.9.2        Model Description

The factors that affect the performance and productivity of GSD projects can be organized into the following three categories:

1.  **Fundamental Factors**
    The fundamental factors relate to the primary characteristics of GSD projects, including communication problems, coordination and control problems, cultural differences, language differences, and time zone differences. A project manager has little or no control over these factors; however, using the right strategy and tool support can reduce their negative impact.

2.  **Strategic Factors**
    The strategic factors are related to high-level issues that the project manager must address when managing a GSD project. Decisions regarding these issues significantly impact the performance of the GSD project. There are five such factors: (1) development site, (2) product architecture, (3) task allocation strategy, (4) distribution overhead, and (5) distribution effort loss.

3.  **Organizational Factors**
    The factors in this category focus on impacts of virtual teams, including team formulation and team dynamics.

All of the above factors are incorporated into this example of a hybrid PSIM.

The model used for this case study has three major components: the DES model, the SD model, and the interaction effect (IE) model.

The DES model includes a **global** DES submodel and a **site-specific** DES submodel for each development site. Each development site may have different process steps, depending on task allocation strategy. The site-specific DES allows the user to capture the impact of these differences. Different time zones also are modeled. Work products are passed from one site to another to capture the effect of distribution overhead and distribution effort loss. The global DES submodel aggregates the information from the site-specific DES submodels to determine overall project progress.

The SD model includes a **global** SD submodel and a **site-specific** submodel for each development site. The global SD submodel captures the overall project environment, including the planning and controlling activities. The global SD submodel has three modules: human resource (HR), planning, and control. The HR module acts as an interface between the HR module from each development site and the other modules within the global SD submodel. The control module receives information about the project progress (from the global DES submodel) and then determines whether adjustments to the schedule or the work rate are needed. The planning module monitors and identifies the workforce level required to meet the overall project schedule.

Each development site has its own site-specific SD submodel. The site-specific SD submodel represents aspects that may be different among development sites, including HR, productivity (PD), manpower allocation (MP), and defect generation and detection rates (QA). The HR module deals with human resource management, which includes hiring, training, assimilation, and transferring human resources within a particular site. The PD module models the rate at which the developers at a particular site can develop software (productivity rate). The MP module assigns the workforce to various activities. The QA module models defect generation, detection, and correction rates. These modules function as if there were only one development site in the project. For example, site-specific productivity assumes that developers are working with others from the same site.

The IE model represents the interaction effects when staff from different sites need to collaborate or work closely together, for example, during follow-the-sun development. When developers work with their colleagues from the same site, information such as productivity and defect rates is sent from the site-specific SD submodel. However, when developers must collaborate with their colleagues from other sites, their productivity will be different. The IE submodel modifies the productivity before sending it to the DES submodel.

Figure 26 shows the overall GSD model structure with two development sites, and Figure 27 provides additional detail regarding the interaction effects associated with productivity.

The data used to drive key aspects of the model were obtained from recently published studies and widely used references [Bass 2004, Carmel 1999, Carmel 2005, Curtis 1988, Herbsleb 1999, 2001, Jones 1998, Software Productivity Research 2007].
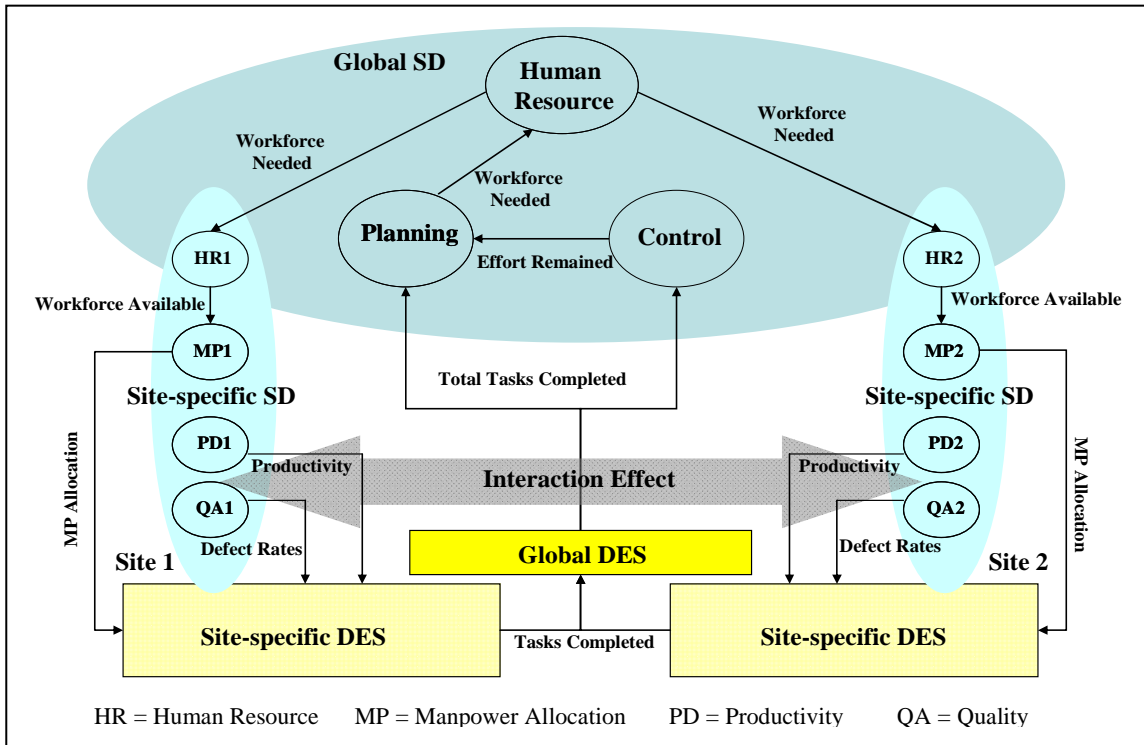
*Figure 26:  Overview of the GSD Model Structure*
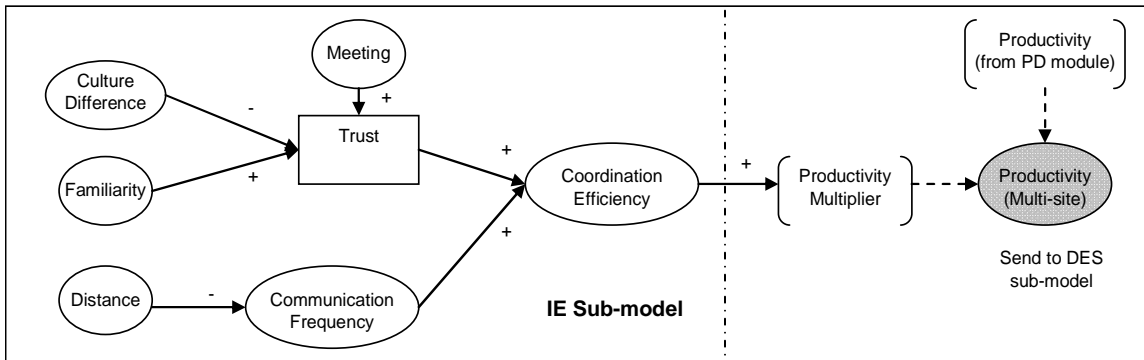


*Figure 27:  The Interaction Effect on Productivity Rate*

### 3.9.3    Model Application/Results

There are five phases in the development process:

1.  Requirements (REQ)
2.  Design (DES)
3.  Coding (CODE)
4.  Testing (TEST)
5.  Rework (RWK)

Due to resource constraints, the main development site would not be able to complete the project within the necessary time window. Therefore, the project manager is considering two additional development sites to help perform the work. Considerations regarding each site are listed below.

**Site A (offshore)**

The time zone difference is eight hours (no overlap working hours).

- The culture and native language of the developers are different.
- The programmer wage is lower than the current site and Site B.

**Site B (near-shore)**

- The time zone difference is only four hours (50% overlap working hours).
- The culture and native language of the developers are the same.
- The programmer wage is higher than Site A.

Both sites have advantages and disadvantages that may or may not offset each other. For Site A, the project can benefit from a 16-hour development window per day by using follow-the-sun development strategy. However, more coordination and communication problems may occur because the developers have different culture and language. This difficulty may require additional effort and time (for issues not addressable by one party alone) to complete the project.

The development window per day is lower with use of Site B. However, the greater overlap in working hours would allow more synchronous communication between the two sites, which could result in better coordination and communication. In addition, since the developers have the same culture and language, miscommunication and coordination problems are likely to be lower, resulting in higher productivity. It is not obvious which development site would be better.

The hybrid PSIM is first configured to represent the current site and Site A; 30 replications were run to obtain the expected project performance, including effort, duration, and quality (number of latent defects). The model was then reconfigured with Site B rather than Site A, and, again, 30 replications were run. Hypothesis tests were performed to determine if the differences in project performance were statistically significant. The results are shown in Figure 28, using the familiar box and whisker plots to show the variation
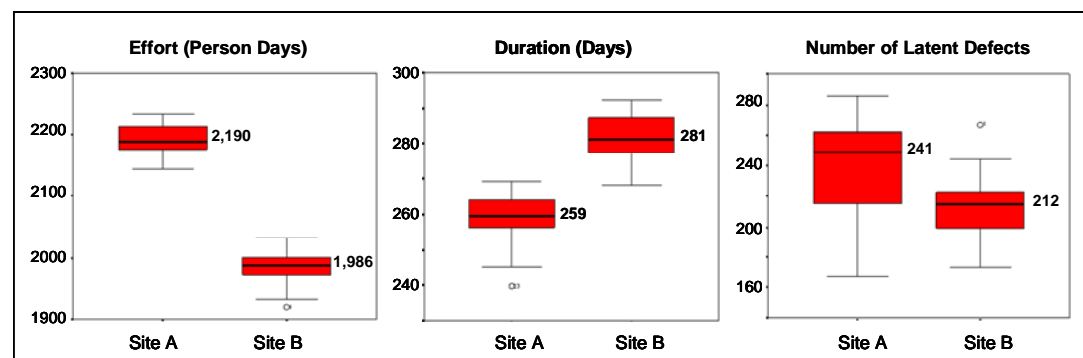


*Figure 28:     Comparison of Effort, Duration, and Defects Using Site A vs. Site B*

The total effort required to complete the project is higher when Site A is used (204 person-days more). The difference is significant at 0.05 level. Working with Site A requires additional effort for coordination. In addition, miscommunication tends to be higher, which results in higher

defects. These defects also require additional effort to rework. However, because the programmer wage in Site A is lower, we cannot automatically conclude that Site B would cost less than Site A. The GSD model also records effort spent at each development site. Figure 29 shows the effort distribution between sites for each alternative (Site A vs. Site B).
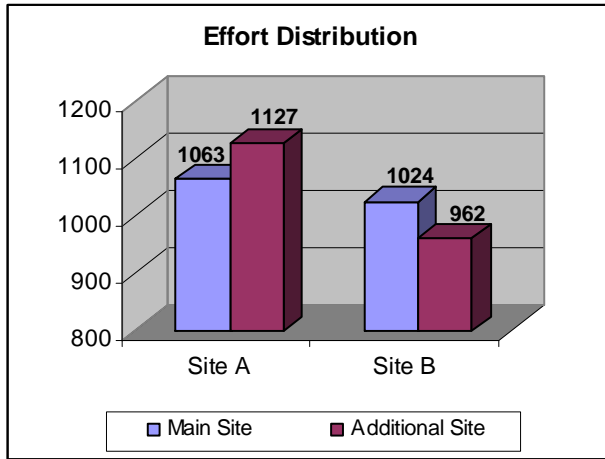


*Figure 29:    Effort Distribution Between Sites for Each Alternative (Site A vs. Site B)*

The effort expended at the main site is approximately the same for two alternatives. The additional effort required for alternative 1 is mainly spent at Site A. In this example, if the wages at Site A are approximately 80% of the wages at Site B, the cost to hire the developers will be about the same.

The duration with Site A is shorter than the duration with Site B (259 vs. 281 days). The difference is statistically significant at 0.05 level. This may be due to the benefits of having more development time per day (16 hours). Although we have to expend more effort when using Site A, the larger number of work hours reduces the project duration. It should be noted that there is approximately a 4% probability that the duration with Site A will be longer than the duration with Site B. This is a rough estimate only, given the uncertainty in the underlying data and model parameters.

The quality of the software was measured as the number of defects escaped, or latent defects. The latent defects with Site B (mean = 212, standard deviation = 21) is lower than the latent defects with Site A (mean = 241, standard deviation = 30). The difference is statistically significant at 0.05 level. As mentioned previously, difference in culture and language and difficulty in communication and coordination are likely to result in more defects. Because both Site A and Site B have about the same capability to detect and correct defects, the site with the higher number of defects injected likely will have the higher number of defects escaped. There is, however, a 15% chance that the quality will be lower (more latent defects) when using Site B versus Site A. This study did not consider the cost to correct the latent defects in the field and, in particular, the difference between sites in coordination cost of correcting such defects.

Neither Site A nor Site B performs best on all three performance measures. Adding Site B to the project results in less effort required and higher quality software (lower defects), but the project duration will be longer. On the other hand, adding Site A contributes to shorter project duration

but higher effort and lower quality. The project manager must make a tradeoff among these three performance measures and determine which alternative will better meet the objectives. For example, if the goal is to reduce the cycle time, adding Site A will work best.

### 3.9.4 Supporting CMMI Processes and Practices

Evaluating the performance of an organization's globally distributed development operations addresses PAs at ML 3 such as Decision Analysis and Resolution (DAR), Organizational Process Focus (OPF), and Risk Management (RM) as well as Project Planning (PP), and Supplier Agreement Management (SAM) at ML 2. Addressing these PAs involves designing a multisite sourcing network and assessing the costs and benefits associated with it, examining the impact of alternative sourcing strategies, and assessing the risk and potential problems. Regarding PP (ML 2), PSIM can be engaged to design the process to be used for the project, establish bottom-up estimates, and examine the impacts of changes to the project plans and sourcing strategy.

### 3.10 SUMMARY

In this section, we showed how PSIM can provide high value using examples from industrial and government organizations. The examples covered issues pertaining to

1. designing, defining, and documenting processes
2. estimating costs from the bottom up
3. planning processes and making tradeoff decisions
4. analyzing and evaluating process improvement opportunities
5. assessing the costs and benefits of applying new tools and technologies
6. managing and controlling processes quantitatively
7. optimizing processes
8. providing training
9. evaluating strategic issues such as globally distributed process tradeoff decisions

For each example, we described the model and the application and how it fulfilled the CMMI PAs in addition to how it contributed to business goals within an organization.

The pattern that emerges from these examples is that PSIM is a perfect fit for organizations that want to improve their process planning, speed technology adoption, optimize process improvement, and step up to quantitative project management. Some of the key areas of CMMI that PSIM directly addresses are as follows:

- PP (ML 2), SP 1.3: defining project life cycles, SP2.2 identifying project risks
- OPF (ML 3): evaluating processes, identifying process improvements, and developing implementation plans
- RSKM (ML 3): identifying process risks, assessing the risk associated with proposed process changes, and evaluating possible mitigations of those risks

- DAR (ML 3): providing decision guidelines, processes, evaluation criteria, and alternative solutions; evaluating alternatives; and making specific recommendations. This is a core value that PSIM provides.
- OPP (ML 4): selecting processes, establishing measures, setting specific performance objectives, and establishing baselines and performance models
- QPM (ML 4): evaluating realism of project objectives, composing the project's defined process, and monitoring and controlling project quality and process performance
- Organizational Innovation and Deployment (ML 5): selecting and deploying incremental and innovative improvements that can measurably improve the organization's processes and technologies
- Causal Analysis and Resolution (ML 5): aiding the understanding of causes of process problems and evaluating alternative action plans to resolve the problems

In Section 4, we assess the degree to which PSIM facilitates and supports each aspect of CMMI, considering each PA, SG, and specific practice SP.

# 4  How PSIM Relates to CMMI

In this report, we have focused on identifying ways in which PSIM can provide value to organizations. We've seen that PSIM helps to increase process innovation, raise quality, and enhance overall project performance through improved process management and decision making. In this section, we identify those CMMI PAs and SGs that are supported by PSIM. We rate the degree to which use of PSIM supports each of these PAs and SGs: Strongly supports, Supports, or Partially supports. For PAs that PSIM supports or strongly supports, we describe how, in some detail.

In developing this material, we have taken a rather conservative approach. For example, we could have asserted much more about how simulation helps satisfy the Verification PA. However, process simulation can be used to evaluate the impact of using different tools to verify products as well as help to design a project's entire verification strategy across multiple locations. Does this have a lot to do with verification? YES (e.g., VER SP 1.1-1.3). Does PSIM fulfill the CMMI PA for Verification? NO. So we have stated, for the purpose of this section, that PSIM only provides partial support for this PA. Other PAs where PSIM makes a related contribution include: Requirements Management, Project Monitoring and Control, Product and Process Quality Assurance, Requirements Development, Validation, and Product Integration. For one CMMI PA, Configuration Management, we noted that there is a distinction between configuration of the product and configuration of the process, and we indicate that PSIM strongly supports configuration of the process, and thereby helps to support implementation of GP 2.6 across the Process Management PAs of CMMI.

Table 19 shows at a glance how PSIM supports or strongly supports each of the CMMI PAs at the PA and SG levels from a staged point of view.

Table 20 provides the same information but organized by process areas area categories, and Table 21 shows ratings of support for the generic goals and practices.

Color codes for the ratings are below. (See next page for an explanation of rating terms.)

| Rating with color code |
|---|
| Strongly supports |
| Supports |
| Partially supports |
| Does not support |

*Table 19:     Rating of How Well PSIM Supports Each CMMI PA (Staged View)*

| CMMI Level 2 | Requirements Management (RM) |
|---|---|
| | Project Planning |
| | Project Monitoring and Control |
| | Supplier Agreement Management |
| | Measurement and Analysis |
| | Process and Product Quality Assurance |
| | Configuration Management (of the *Process*) |
| **CMMI Level 3** | Requirements Development |
| | Technical Solution |
| | Product Integration |
| | Verification |
| | Validation |
| | Organizational Process Focus |
| | Organizational Process Definition |
| | Organizational Training |
| | Integrated Project Management |
| | Risk Management (especially process risks) |
| | Decision Analysis and Resolution |
| **CMMI Level 4** | Organizational Process Performance |
| | Quantitative Project Management |
| **CMMI Level 5** | Organizational Innovation and Deployment |
| | Causal Analysis and Resolution |

Explanations of the ratings are below.

| Rating | Meaning |
|---|---|
| Strongly supports | PSIM is a primary tool that can either fulfill this CMMI PA or one of its components (i.e., SG, SP, GG, or GP) directly or provide considerable support in implementing the PA and when used in conjunction with other tools or methods. |
| Supports | PSIM provides an important part of the solution for effectively fulfilling this CMMI element and can be used as part of a suite of tools and/or methods. |
| Partially supports | PSIM can provide some useful capabilities but does not provide the full solution. |
| Does not support | PSIM does not support this PA. |

*Table 20:     Rating of How Well PSIM Supports Each CMMI PA (Continuous Representation)*

| Process Management | Organizational Process Focus |
|---|---|
| | Organizational Process Definition + IPPD |
| | Organizational Training |
| | Organizational Process Performance |
| | Organizational Innovation and Deployment |
| **Project Management** | Project Planning |
| | Project Monitoring and Control |
| | Supplier Agreement Management |
| | Integrated Project Management |
| | Risk Management |
| | Quantitative Project Management |
| **Engineering** | Requirements Management |
| | Requirements Development |
| | Technical Solution |
| | Product Integration |
| | Verification |
| | Validation |
| **Support** | Configuration Management (of the process) |
| | Product and Process Quality Assurance |
| | Measurement and Analysis |
| | Decision Analysis and Resolution |
| | Causal Analysis and Resolution |

*Table 21:        Rating of How Well PSIM Supports the CMMI Generic Goals and Practices*

| | |
|---|---|
| **GG 1 Achieve Specific Goals** | GP 1.1 Perform Specific Practices |
| **GG 2 Institutionalize a Managed Process** | GP 2.1 Establish an Organizational Policy |
| | GP 2.2 Plan the Process |
| | GP 2.3 Provide Resources |
| | GP 2.4 Assign Responsibility |
| | GP 2.5 Train People |
| | GP 2.6 Manage Configurations |
| | GP 2.7 Identify and Involve Relevant Stakeholders |
| | GP 2.8 Monitor and Control the Process |
| | GP 2.9 Objectively Evaluate Adherence |
| | GP 2,.10 Review Status with Higher Level Mgmt. |
| **Institutionalize a Defined Process** | GP 3.1 Establish a Defined Process |
| | GP 3.2 Collect Improvement Information |
| **Institutionalize a Quantitatively Managed Process** | GP 4.1 Establish Quantitative Objectives for the Process |
| | GP 4.2 Stabilize Subprocess Performance |
| **Institutionalize an Optimizing Process** | GP 5.1 Ensure Continuous Process Improvement |
| | GP 5.2 Correct Root Causes of Problems |

The purpose of this report is to provide insight into the potential capabilities of PSIM and opportunities for applying PSIM to software and systems development projects. Throughout this section, we point out PSIM capabilities directly related to the CMMI component under discussion as well as capabilities that CMMI may not specifically call for but that can nevertheless significantly improve the software development process. The following tables provide details related to the CMMI PAs for which PSIM was rated as "strongly supports" or "supports." The ratings for the individual SPs in the following tables represent our best judgment, although certainly one could make a case for different ratings. Note that IPPD is not a focus of this report. Further, the list of SPs below a given PA includes only those that are at least partially supported using PSIM.

## 4.1   PROCESS AREAS AT MATURITY LEVEL 2

**PROJECT PLANNING**

The purpose of Project Planning is to establish and maintain plans that define project activities.

→ Overall, PSIM provides considerable support for portions of this activity. PSIM can be used in a number of ways, including to design the process to be used on the project, establish bottom-up estimates, examine the impacts of increasing or decreasing functionality or degree of modularity, explore the use of iterative or risk-driven project lifecycles, and determine the impact of requirements volatility. Thus, PSIM supports this PA.

*Table 22:       How PSIM Supports the Project Planning PA*

| SG 1 Establish Estimates | | PSIMs can capture the project WBS at a detailed level. As a result, PSIM is an excellent tool for creating bottom-up estimates for project performance. These bottom-up estimates can be used in conjunction with top-down estimates from tools such as COCOMO, SEER, etc. Furthermore, PSIM models often maintain databases that contain key project and process data that can be used for making estimates and recording their rationales. |
|---|---|---|
| | SP 1.1 Estimate the Scope of the Project | Partially supports | • PSIM models can be used to organize and plan the process that is used to accomplish the project. |
| | SP 1.2 Establish Estimates of Work Product and Task Attributes | Supports | • PSIM models utilize key project data (especially size and other work product attributes such as complexity, error proneness, etc.) as a basis for project performance estimates. |
| | SP 1.3 Define Project Life Cycle | Strongly supports | • PSIM models can be used to design and specify the entire life cycle for a project, or to explore the implications of using a particular life cycle. |
| | SP 1.4 Determine Estimates of Effort and Cost | Strongly supports | • PSIM models can be used to make bottom-up estimates for overall effort and cost based on historical data and estimates of individual task characteristics. |
| SG 2 Develop a Project Plan | | PSIM supports the development of a realistic project plan by showing the project manager the likely results of adopting a given project plan. |
| | SP 2.1 Establish the Budget and Schedule | Supports | • PSIM models can be used to assess the impact of project budget and schedule constraints as well as contingencies or planned changes to the process. |
| | SP 2.2 Identify Project Risks | Supports | • PSIM can be used to assess **process-oriented risks** and to explore **process-focused** risk mitigation strategies, such as changing the project's verification and validation strategy or IV&V strategy to lower technical risk. |
| | SP 2.3 Plan for Data Management | Partially supports | • PSIM models utilize and can serve as repositories for a variety of project data, which could serve as a component of the data management plan. |
| | SP 2.4 Plan for Project Resources | Supports | • PSIMs can be used to evaluate candidate project resource plans, to explore the impact of resource constraints, and to find ways to cost-effectively mitigate those impacts. |
| SG 3 Obtain Commitment to the Plan | | PSIM can provide partial support for obtaining commitment to the project plan, mostly by helping to clearly document the plan and demonstrate its feasibility/effectiveness. |
| | SP 3.2 Reconcile Work and Resource Levels | Partially supports | • PSIM can reveal potential resource bottlenecks and help to evaluate "what if" scenarios regarding resource levels. |
| | SP 3.3 Obtain Plan Commitment | Partially supports | • PSIM output can be used by the PM to determine the realism/feasibility of alternative plans, and create a business case for the recommended choice. |

## MEASUREMENT AND ANALYSIS

The purpose of Measurement and Analysis is to develop and sustain a measurement capability that is used to support management information needs.

→ Overall, PSIM strongly supports measurement and analysis.

Table 23:     How PSIM Supports the Measurement and Analysis PA

| SG 1 Align Measurement and Analysis Activities] | | PSIM strongly supports the alignment of measurement and analysis with the management needs of the organization and project. | |
|---|---|---|---|
| | SP 1.1 Establish Measurement Objectives | Strongly supports | • PSIMs provide a framework and focus for establishing important metrics to be collected, using the Goal, Question, Indicator, Measure (GQIM) approach. |
| | SP 1.2 Specify Measures | Strongly supports | • PSIMs and GQIM can help the project manager to determine which metrics are likely to be the most critical to success and may help to clarify how these metrics should be specified. |
| | SP 1.3 Specify Data Collection and Storage Procedures | Supports | • Metrics repositories connected with PSIMs can be used to store process and product data from disparate company databases [Harrison 2000]. |
| | SP 1.4 Specify Analysis Procedures | Strongly supports | • Established methods and procedures exist for analyzing metrics data, PSIM parameters, and PSIM results. |
| SG 2 Provide Measurement Results | | PSIM strongly supports the goal of providing measurement results, including analysis, storage, communication, and learning. | |
| | SP 2.2 Analyze Measurement Data | Strongly supports | • PSIMs provide an important tool and framework for analyzing the implications of the data being measured, and this can be done almost in real time.<br>• The measured data can be quickly compared to plans, which can help to determine if the discrepancies are due to poor planning, ineffective measurement, or both. |
| | SP 2.3 Store Data and Results | Strongly supports | • Databases connected to PSIMs can store initial projections, rationale, actual results, revised projections, and so forth.<br>• Thus, both the current status and the complete history can be maintained to facilitate planning and enable learning from mistakes. |
| | SP 2.4 Communicate Results | Strongly supports | • PSIMs provide both the high- and low-level information needed by management and staff to support decisions.<br>• PSIMs also provide a consistent format for communicating results. |

## CONFIGURATION MANAGEMENT

The purpose of Configuration Management (CM) is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.

→ Although PSIM does not directly support CM for the *product*, PSIM tools strongly support CM for the *process*, as indicated in the table below.

*Table 24:     How PSIM Supports Configuration Management of the Process*

| SG 1 Establish Baselines | | PSIM supports establishing and maintaining process baselines. |
|---|---|---|
| SP 1.1 Identify Configuration Items | Supports | Models can be built for life-cycle processes, process components, associated measures, approved process tailoring options, and process improvement opportunities as well as process contingencies. These process elements constitute many components of the process asset library (PAL) for an organization. |
| SP 1.2 Establish a Configuration Management System | Supports | Process configurations can be stored as different models. Process components, improvements, and contingencies can also be stored. |
| SP 1.3 Create or Release Baselines | Supports | Organizational baseline models can be created from the PAL. |
| SG 2 Track and Control Changes | | PSIM supports the need to track and control changes to the software development process. |
| SP 2.1 Track Change Requests | Supports | Process improvement options can be easily modeled using PSIMs. |
| SP 2.2 Control Configuration Items | Supports | Process improvement options can be quantitatively evaluated against organizational requirements. Only approved improvement options will be implemented. |
| SG 3 Establish Integrity | | Overall, PSIM supports the establishment of process integrity. |
| SP 3.1 Establish Configuration Management Records | Supports | PSIM tools provide documentation for process baselines and process improvements. |
| SP 3.2 Perform Configuration Audits | Partially supports | Auditors can take advantage of the models and associated process documentation. |

## 4.2 PROCESS AREAS AT MATURITY LEVEL 3

**ORGANIZATIONAL PROCESS FOCUS**

The purpose of Organizational Process Focus is to plan and implement organizational process improvement based on a thorough understanding of the current strengths and weaknesses of the organization's processes and process assets.

→ Overall, PSIM strongly supports an organizational process focus.

*Table 25:      How PSIM Supports the Organizational Process Focus PA*

| SG 1 Determine Process Improvement Opportunities | PSIM supports the identification of process improvement opportunities | |
|---|---|---|
| SP 1.1 Establish Organizational Process Needs | Supports | By allowing processes to be formally described and evaluated, PSIM- enables achievability of needs and priorities to be evaluated. |
| SP 1.2 Appraise the Organization's Processes | Supports | PSIM is an ideal tool for process performance appraisal (see below for more discussion). |
| SP 1.3 Identify the Organization's Process Improvements | Strongly supports | By providing a virtual laboratory for studying bottlenecks and improvement ideas, PSIM enables process alternatives to be assessed, and the best ideas selected for implementation. |
| SG 2 Plan and Implement Process Improvement | PSIM strongly supports the planning and implementation of process improvements | |
| SP 2.1 Establish Process Action Plans | Strongly supports | PSIM provides a consistent environment for considering different process action plans. PSIM helps to determine the most effective sequence of actions. |
| SP 2.2 Implement Process Action Plans | Supports | PSIM facilitates communication of planned process changes—including their beneficial impact. |
| SG 3 Deploy Organizational Process Assets and Incorporate Lessons Leaned | PSIM supports and facilitates deployment and provides a tool for learning and knowledge capture. | |
| SP 3.1 Deploy Organizational Process Assets | Supports | The PSIM process repository encourages process reuse and standardization. |
| SP 3.2 Deploy Standard Processes | Supports | PSIM clearly describes standard processes. |
| SP 3.4 Incorporate Process-Related Experiences into the Organizational Process Assets | Supports | PSIM can serve as the repository where templates and lessons learned based on actual experience are stored. |

SP 1.2: Process performance appraisal can be facilitated by formally and quantitatively describing and analyzing processes. PSIM enables this endeavor and allows for comparison of current process performance to standards and other benchmarks. The formal description includes the detailed flow of work products over the entire product life cycle, including the injection, detection, and correction of errors in requirements, design, coding, and integration. Besides allowing process parameters to be entered quantitatively, PSIM also enables historical and estimated process variability to be fully reflected. This allows for assessment of the risk of negative outcomes in addition to determination of the expected nominal process performance.

**ORGANIZATIONAL PROCESS DEFINITION**

The purpose of Organizational Process Definition is to establish and maintain a usable set of organizational process assets.

→ Overall, PSIM strongly supports organizational process definition.

*Table 26:      How PSIM Supports the Organizational Process Definition PA*

| SG 1 Establish Organizational Process Assets | PSIM provides considerable support. | |
|---|---|---|
| SP 1.1 Establish Standard Processes | Strongly Supports | PSIM tools serve as the process asset library, both descriptively and quantitatively.<br>PSIM encourages the reuse of successful processes. |
| SP 1.2 Establish Life Cycle Model Descriptions | Strongly supports | PSIM incorporates a broad array of process documentation.<br>PSIM encourages consideration of the full product life cycle during project planning. |
| SP 1.3 Establish Tailoring Criteria and Guidelines | Strongly supports | PSIM provides examples of the results of previous tailoring activities.<br>PSIM provides a repository and menu of approved tailoring options. |
| SP 1.4 Establish the Organization's Measurement Repository | Supports | PSIM provides a framework for metrics collection on a project.<br>Actual process measures are used to help create the parameters used in PSIMs. |
| SP 1.5 Establish the Organization's Process Asset Library | Strongly supports | PSIM can serve directly as the process asset library (see below for more discussion). |

SP 1.5: PSIM is an ideal environment for capturing process knowledge and encouraging standardization and reuse. The captured information is both visual and quantitative, as PSIM tools often include a database containing process parameters, policies, and other parameters.

**INTEGRATED PROJECT MANAGEMENT**

The purpose of Integrated Project Management is to establish and manage the project and the involvement of the relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes.

→ Overall, PSIM strongly supports a number of SPs for this activity.

*Table 27:     How PSIM Supports the Integrated Project Management PA*

| SG 1 Use the Project's Defined Process | PSIM strongly supports the use of defined processes and fulfills several of the specific SPs. | | |
|---|---|---|---|
| SP 1.1 Establish the Project's Defined Process | Strongly supports | The project-defined process can be described and fully documented within the PSIM environment through extensive drawing from the process library (see below for more discussion). | |
| SP 1.2 Use Organizational Process Assets for Planning Project Activities | Strongly supports | PSIM provides a virtual laboratory in which the project manager can explore tailoring options (see below for more discussion). | |
| SP 1.4 Integrate Plans | Supports | PSIMs can be aggregated to include multiple projects and organizations as desired. | |
| SP 1.5 Manage the Project Using the Integrated Plans | Strongly supports | PSIM allows project/program managers to project future outcomes and anticipate corrective action proactively rather than reactively. | |
| SP 1.6 Contribute to the Organizational Process Assets | Strongly supports | Data, artifacts, and results from each project's experience can be preserved within a PSIM library (see below for more discussion). | |
| SG 2 Coordinate and Collaborate with Relevant Stakeholders | PSIM provides partial support. | | |
| SP 2.2 Manage Dependencies | Partially supports | May help to reveal critical dependencies that must be closely monitored to assure success | |
| SP 2.3 Resolve Coordination Issues | Partially supports | For issues that can be modeled, stakeholders can use PSIM to help establish a shared set of assumptions and a way to reason through different options. | |

SP 1.1: This is a natural fit, especially if the process asset library is managed using a PSIM tool, in which case the defined process for a specific project is developed by adapting (tailoring) either a generic process template or another similar ready-tailored process model.

SP 1.2: PSIMs provide a specific view into the process. This view includes workflow, process agents or resources, process descriptions, and quantitative measurement. PSIM therefore enables the organization to document and better utilize its organizational process assets.

SP 1.6: PSIMs can themselves be considered a process asset of the organization, as they document the organization's historical processes and can be leveraged to create highly tailored future processes.

**RISK MANAGEMENT**

The purpose of Risk Management is to identify potential problems before they occur, so that risk-handling activities may be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives.

→ Overall, PSIM strongly supports risk management. Using PSIM to evaluate processes can help identify risks. Improving processes can often mitigate risks, and PSIM can also be used to assess candidate risk mitigation strategies.

*Table 28:     How PSIM Supports the Risk Management PA*

| SG 1 Prepare for Risk Management | PSIM provides support for improving understanding of process-related risks. | |
|---|---|---|
| SP 1.2 Define Risk Parameters | Supports | • The impact of product and technical risks on process performance can be represented in the PSIM. |
| SP 1.3 Establish a Risk Management Strategy | Supports | • PSIM allows assessment of the potential impact of identified risks, which can help to form a risk management strategy. |
| SG 2 Identify and Analyze Risks | PSIM strongly supports the analysis of process risks. | |
| SP 2.1 Identify Risks | Supports | • The impact of product and technical risks on process performance can be represented using PSIM.<br>• Risk management is not a simple linear activity (identify, then analyze); instead, risks often emerge from complex interactions. One of the PSIM's strengths is its capacity to represent such risks. |
| SP 2.2 Evaluate, Categorize, and Prioritize Risks | Strongly supports | • PSIM helps to assess overall project risk in terms of schedule, quality, and cost, given specific risk factors and estimates of variability (see comments below for further discussion).<br>• PSIM also allows the quantitative assessment of process risks and process changes. |
| SG 3 Mitigate Risks | PSIM supports the mitigation of risks, especially process-related risks. | |
| SP 3.1 Develop Risk Mitigation Plans | Supports | • PSIM evaluates the impact of alternative risk mitigation strategies<br>• PSIM helps the project manager develop/select a project plan that minimizes overall project risk |

SP 2.2: For example, consider a proposed process change: defect detection capability at the design stage. This change may directly impact effort and schedule by only a modest amount. However, because the change is embedded within a complex process, the overall impact on project performance may be compounded. PSIM can reveal these impacts. Obviously, if defect detection during design strongly impacts overall product quality, then process changes that impact this parameter possess high leverage. PSIM enables the project manager to identify and evaluate this risk/opportunity.

**DECISION ANALYSIS AND RESOLUTION**

The purpose of Decision Analysis and Resolution is to analyze possible decisions using a formal evaluation process that evaluates identified alternatives against established criteria.

→ Overall, PSIM strongly supports this PA by providing an extensive analytical framework for making tradeoffs and developing a business case to support decisions. This is particularly true for decision alternatives involving processes.

*Table 29:      How PSIM Supports the Decision Analysis and Resolution PA for Decisions Involving Processes*

| SG 1 Evaluate Alternatives | PSIM Strongly supports this goal with respect to process decisions. | |
|---|---|---|
| SP 1.1 Establish Guidelines for Decision Analysis | Supports | • by directing the project manager toward the use of a quantitative basis for decision analysis |
| SP 1.2 Establish Evaluation Criteria | Strongly supports | • by encouraging the use of objective "bottom line" criteria, including quality, schedule, and cost (of course)<br>• by providing specific detailed examples of these criteria |
| SP 1.3 Identify Alternative Solutions | Strongly supports | • via the repository of other projects, each of which often represents a different process or approach<br>• by enabling  easy creation of new alternative processes |
| SP 1.4 Select Evaluation Methods | Strongly supports | • by encouraging the project manager to take a business case approach to the evaluation of alternatives |
| SP 1.5 Evaluate Alternatives | Strongly supports | • by providing an extensive analytical framework (see below for more discussion) |
| SP 1.6 Select Solutions | Strongly supports | • by allowing easy comparison of tradeoffs<br>• by making it easy, for example, to incorporate utility functions into PSIM to help select the best alternative from an overall perspective |

SP 1.5: This is one area where PSIM really shines, because it allows the project manager to explore in a virtual laboratory a wide range of process alternatives, resource strategies, decision criteria, and project policies.

## 4.3  PROCESS AREAS AT MATURITY LEVEL 4

**ORGANIZATIONAL PROCESS PERFORMANCE**

The purpose of Organizational Process Performance is to establish and maintain a quantitative understanding of the performance of the organization's set of standard processes in support of quality and process-performance objectives and to provide the process performance data, baselines, and models to quantitatively manage the organization's projects.

→ PSIM strongly supports this activity.

*Table 30:       How PSIM Supports the Organizational Process Performance PA*

| SG 1 Establish Performance Baselines and Models | PSIM can be used to fulfill this goal. | |
|---|---|---|
| SP 1.1 Select Processes | Supports | • PSIM provides the tool for identifying and prioritizing contributing influences/leading indicators/controllable parameters to project success. |
| SP 1.2 Establish Process Performance Measures | Strongly supports | • PSIM provides a framework and a focus for selecting those metrics that would mostly likely be useful and revealing<br>• PSIM can be used to assess critical process performance measures. |
| SP 1.3 Establish Quality and Process-Performance Objectives | Supports | • PSIM can help one to better understand the relationship between quality and process performance vis-a-vis management targets for final quality [Raffo 2003]. |
| SP 1.4 Establish Process Performance Baselines | Supports | • PSIM provides an objective basis to determine likely process performance (see below for more discussion). |
| SP 1.5 Establish Process Performance Models | Strongly supports | • This is PSIM, almost by definition (see below for more discussion). |

SP 1.4: Using PSIM, SP 1.5 would actually come first. Then, the models established in SP 1.5 would be run using different assumptions, from cautious to optimistic, to determine a variety of baselines. What-if scenarios would also typically be run to establish contingency plans.

SP 1.5: Once adopted and implemented, PSIM tools may include a library of models and a database of parametric data that together enable project managers to predict process/project performance under different conditions and using different process configurations.[11]

---

[11]    Note that PSIMs can be used to establish "trial" Process Performance Baselines (PPBs), thereby helping organizations with process data get an "early" start at implementing OPP (in the sense of not needing to wait until QPM is implemented for some projects to establish PPBs).

## QUANTITATIVE PROJECT MANAGEMENT

The purpose of the Quantitative Project Management process area is to quantitatively manage the project's defined process to achieve the project's established quality and process-performance objectives.

→ Overall, PSIM strongly supports quantitative project management.

Table 31:     How PSIM Supports the Quantitative Project Management PA

| SG 1 Quantitatively Manage the Project] | | PSIM strongly supports this goal. | |
|---|---|---|---|
| SP 1.1 Establish the Project's Objectives | Strongly supports | • | by helping one to explore the potential impact of different objectives and to evaluate their feasibility/realism |
| | | • | by helping one to set intermediate and supplier process performance objectives for quality targets (see discussion below) |
| SP 1.2 Compose the Defined Process | Strongly supports | • | by providing the process asset library with both generic and previously tailored process components (see discussion below) |
| SP 1.3 Select the Subprocesses that Will Be Statistically Managed | Supports | • | by revealing which subprocesses have the most impact on project outcomes |
| | | • | by helping the project manager to understand the critical path for the project |
| SP 1.4 Manage Project Performance | Strongly supports | • | by providing a tool for the project manager to use to predict possible impacts of changes in project goals, resources and other what-if questions |
| | | • | by facilitating closed-loop corrective action |
| | | • | by enabling proactive response to changes and deviations |
| SG 2 Statistically Manage Subprocess Performance | | PSIM strongly supports and derives benefits from this activity. | |
| SP 2.1 Select Measures and Analytic Techniques | Strongly supports | • | by identifying problematic aspects where measures are most needed |
| | | • | by providing powerful analytic capability and methods for estimating project performance and evaluating sources of variation |
| SP 2.2 Apply Statistical Methods to Understand Variation | Strongly supports | • | by aiding understanding of the impact of process variation (see comment below) |
| | | • | by fully incorporating statistical methods |
| SP 2.3 Monitor Performance of the Selected Subprocesses | Supports | • | by identifying necessary performance thresholds that subprocesses need to achieve in order to satisfy overall project performance goals [Raffo 2005a, 2003] |
| | | • | by facilitating planning and monitoring at the subprocess level. |
| SP 2.4 Record Statistical Management Data | Supports | • | by recording statistical information for process steps and activities that are captured in the model |

SP 1.1: Management must set targets for process and quality performance at the project level. Once this is done, PSIM can be used to (1) monitor the process using the PROMPT method, (2) set intermediate and supplier quality and process performance thresholds that must be achieved to meet management targets, and (3) to evaluate the feasibility of a particular set of quality and process performance objectives [Raffo 2005a, 2003].

SP 1.2: PSIM provides a repository of standard and previously configured processes that allows the project manager to quickly create and evaluate candidate project processes to fit with the needs of the specific project. After considering alternatives, the project manager determines the best overall process approach that becomes the defined process for the project.

SP 2.2: Users sometimes mistakenly interpret QPM SP 2.2 as referring to use of control charts, but the focus is really on understanding variation, where PSIMs can really help.

## 4.4 PROCESS AREAS AT MATURITY LEVEL 5

### ORGANIZATIONAL INNOVATION AND DEPLOYMENT

The purpose of Organizational Innovation and Deployment PA is to select and deploy incremental and innovative improvements that measurably improve the organization's processes and technologies. The improvements support the organization's quality and process-performance objectives as derived from the organization's business objectives.

→ Overall, PSIM strongly supports innovation and deployment.

*Table 32:     How PSIM Supports the Organizational Innovation and Deployment PA*

| SG 1 Select Improvements | PSIM strongly supports the process of selecting improvements. | |
|---|---|---|
| SP 1.1 Collect and Analyze Improvement Proposals | Strongly Supports | • PSIM provides analysis of the likely impact of potential innovative improvements, resource policies, quality expectations, etc.<br>• The PSIM framework by its very nature encourages process improvement. |
| SP 1.2 Identify and Analyze Innovations | Strongly supports | • PSIMs provide a powerful mechanism for identifying bottlenecks (more generally, TARGETS) for improving product quality and process performance.<br>• PSIM provides analysis of the likely impact of potential innovations, such as automated configuration management, testing methodologies, etc.<br>• The PSIM framework by its very nature encourages process innovation. |
| SP 1.3 Pilot Improvements | Supports | • High-fidelity PSIMs provide an alternative to expensive and time consuming pilots. |
| SP 1.4 Select Improvements for Deployment | Strongly supports | • Results of the simulation can be used to select improvements for deployment.<br>• PSIM can evaluate sets of proposed improvements to better understand their interactions before investing in their deployment. For example, the Process Tradeoff Analysis Method (PTAM) provides an effective approach for trading off among process alternatives. |
| SG 2 Deploy Improvements | PSIM supports the deployment of improvements. | |
| SP 2.1 Plan the Deployment | Supports | • PSIM provides tools for project managers to anticipate both the short-term (adverse) impacts and long-term benefits, and adjust expectations and resources accordingly<br>• Understanding interactions promotes better anticipation of needed resources, likely costs, and other information of interest to PMs. |
| SP 2.2 Manage the Deployment | Supports | • PSIM allows project managers to compare actual results to plan and determine the impact of possible corrective actions.<br>• PSIMs may allow estimates/predictions of PPBs to help in establishing PPMs in support of early stages of deployment of process changes to projects. |
| SP 2.3 Measure Improvement Effects | Supports | • PSIM tools support "evaluation" of deployment; impacts and effectiveness. |

## 4.5 SUMMARY

PSIM strongly supports organizations as they strive to progress to higher levels of CMMI. In Table 32 we see that PSIM supports or strongly supports PAs at all levels of CMMI and a greater portion of the PAs at the higher levels: 3 out of 8 of the PAs at CMMI level 2, 5 out of 11 of the PAs at CMMI level 3, and 3 out of 4 of the PAs at CMMI levels 4 and 5.

PSIM is a technology that companies can engage early and obtain benefits from throughout their progression with CMMI. Even at CMMI level 2, PSIM supports PAs such as Project Planning, Measurement and Analysis, and Configuration Management. At level 3, more PAs deal with process planning, process definition, risk management, and decision making. These activities are at the core of the value that PSIM provides. At levels 4 and 5 of CMMI, organizations have more precise and detailed data. Predictions get sharper, and the project management and process optimization aspects of PSIM come into play. In addition, PSIMs provide collateral benefits to project managers making various process tradeoffs, including QA (verification and validation) strategies.

Overall, using PSIM provides a framework and focus for achieving a number of key PAs in CMMI and making process improvement easier to accomplish. PSIM provides many benefits and core business value to users. Supporting and fulfilling many PAs of CMMI is just one of these core benefits to the business.

# 5  Conclusions

PSIM is a technology that has received increasing attention in research and academic domains but has not yet been widely implemented in practice. Recent developments have made this technology much more attractive, and its application is now considered to be a repeatable practice. The purpose of this report was to introduce PSIM technology to the community of practitioners by identifying the benefits and costs associated with it, outlining different approaches that have been used, and providing examples and case studies that showcase high-value applications of PSIM in specific organizational contexts.

PSIM is a maturing technology that is highly applicable to large-scale software and systems projects. It is a technology for which the time has come. Why?

- **PSIM provides tangible benefits**. Whether the goal is designing and tailoring processes quickly, focusing scarce resources on process improvements that provide the highest financial and overall benefit, assessing risk, or evaluating strategic issues (ranging from optimizing quality assurance strategies to evaluating global software development tradeoffs), PSIM is being used in ways that provide tangible value.

  Specific benefits of PSIM include
  - selection of the best possible development process or quality assurance strategy (V&V or IV&V) for a given situation/circumstance
  - improved project planning through the use of an objective and quantitative basis for decision making
  - enhanced project execution (control and operational management) because PSIM can quickly evaluate alternative responses to unplanned events before decisions are made
  - a means for project managers to answer their burning questions
    > What is the impact on project performance of increasing or decreasing testing, inspections, or both? What is the risk? What is the ROI?
    > What development phases/steps are essential to success?
    > What is the value of applying automated tools?
    > How do I objectively compare and prioritize process changes?
    > What specific process changes would help me to achieve higher levels of the CMMI standard? Do they provide business value?
  - improved understanding of the many factors that influence project success for complex software development projects
  - enhanced ability to communicate process choices and alternatives because intangible processes are made more visible and concrete
  - better training and learning for project managers, project team members, and executive leadership
  - elevation of project management to a more strategic level through support of projects' analyses over their full life cycles and with respect to multiple measures of success
- **Organizations cannot continue to accept the high cost of poor decisions.** Increasingly, organizations cannot absorb the high cost and schedule delays associated with poorly

performing software development projects. Competition is fierce, and leading firms have learned how to most effectively develop software; other firms must follow suit.

- **The benefits of PSIM are increasingly recognized, valued, and articulated.** PSIM can be used to document the To-Be vision of how a project's process could perform. PSIM can quantify the benefits and assess the risks associated with that vision. In addition, PSIMs can be used to document current As-Is processes and show the high cost of inefficient performance. Articulating this information in a business case format is extremely useful—to highlight the need for change and the opportunities change will enable.

- **Companies** and managers have an increased awareness and drive to achieve greater levels of performance. Companies need information at the level provided by PSIM in order to make informed decisions about potential impacts, ROI, and overall financial results. Other approaches (e.g., cost estimation tools) cannot provide information at the necessary level.

- **PSIM supports organizational desire to increase process maturity and capability**. At each CMMI level, PSIM helps to enable/fulfill or strongly support a number of key PAs, SGs, and SPs; Sections 3 and 4 show specifically how this can be accomplished, and provide details as to how PSIM strongly supports CMMI.

- **PSIM technology, tools, and training are better** and more readily available than previously.
  - Process modeling technology is no longer mysterious, and the process to build models and apply models to make decisions is now well understood.
  - Tools are available that dramatically reduce the cost to develop and use PSIMs.
  - Training is available to help staff set up and apply the models within an organization.

- **Flawless data is not required.** For certain types of decisions, using data from a life-cycle model template (e.g., Waterfall, Incremental) using industry standard data that is tuned to high-level organizational data can provide useful and informative results.

- **More data is available within organizations as well as externally from industrial sources to support quantitative models.** As more companies at CMMI level 2 strive toward achieving CMMI level 3, they often find that the data needed for PSIMs to be effective within their organization is already available.

The main challenges associated with applying PSIM are described below.

**Cost:** The main costs associated with PSIMs are the cost to design and develop an initial model, to collect model data, and to utilize and maintain the model. Designing and developing PSIMs requires effort—to understand the processes being modeled, collect the data, and build the model using an appropriate simulation tool. This often requires specialized knowledge and skills. Costs of data collection can include costs associated with obtaining process metric data and defining model parameters. However, recent developments have helped to alleviate these costs. Moreover, the simple fact is that typically PSIM studies more than pay for themselves when used to evaluate even a single decision.

**Data:** Models require data. Organizations that have achieved CMMI levels 3, 4, and 5 typically collect sufficient data to support PSIMs. Also, recent developments have reduced the data required to obtain useful results from PSIMs.

**Complexity:** Models are complex because the real world processes they represent are complex. At the same time, models need not be intimidating and they can be designed to buffer users from complex details, if desired. This design extends from interfaces to output analysis. Recent developments in PSIMs address all areas connected with this concern.

**Risk:** The main risk with PSIM is the same as for any model or tool: It's possible for it to be misapplied or misused and for results to be misinterpreted. Poor data as well as lack of user knowledge and skill can have undesired effects. However, users can avoid these undesired effects by using known organizational data sets or industry standard data parameters. In addition, PSIMs can be used in conjunction with top-down cost estimation models to maintain independence and checks on results.

The time has clearly come for PSIM technology. It is a perfect fit for organizations that want to improve process planning, speed technology adoption, optimize process improvement, step up to quantitative project management, and move to the higher levels of CMMI. PSIM is not a silver bullet. However, industrial and governmental organizations are increasingly applying PSIM to achieve significant benefits, and it is arguably the single most useful tool for improving process maturity and capability and moving up CMMI.

# Appendix A: Selection Criteria for Simulation Tools

The following table was adapted from the work of Mueller [Mueller 2006].

*Table 33:  Selection Criteria for Simulation Tools*

| Criteria | Description |
| --- | --- |
| solution provider/tool vendor | size, history, start of business, customers, experience, number of developers, business model, resellers, training and workshops |
| market presence | number of installations, primary customers, references, independent reports |
| support | hotline, user-group, examples, service contracts, training courses, info news |
| hardware requirements | OS, CPU, RAM |
| software system and other tools | OS, user interface, SW for analysis of results |
| required qualification | PC, programming, simulation |
| prices | single-license, multi-user license, hardware, additional SW, run-only option, test installation, maintenance, training, installation |
| other criteria | future support, compatibility |
| primary application area | business, manufacturing, logistics |
| simulation of | employees, costs, tasks |
| techniques supported | DES, SD, others |
| model size and complexity | number of elements, variables |
| predefined simulation elements | tailored to software and business processes? |
| programming languages | scripting, C |
| ease of use | menus, dialogues, visual interactive simulation environments (VSIM), editor, compiler |
| control elements | distributions, I/O-interfaces, control of warm-up, online help, search function |
| animation | during simulation, after simulation, windows, zoom function, results on screen, dynamic representation of parameters |
| simulation runs | real time, continuous representation, batch function, trace function, step function, interruption, time dimension definable |
| debugging | syntax checker, consistency checker, error messages, debugger |
| analysis of results | automatic statistics of all elements, visualization of output on screen, plot into files, screen hardcopy |
| interfaces to other systems | I/O to external files and databases, dynamic input of data |
| documentation of input data | automatic processing of input data, plot of input data |
| product information | references in journals |

# Appendix B: Key Components of a Discrete Event Simulation Model

The main components of a discrete event simulation (DES) model are shown in Figure 30:.
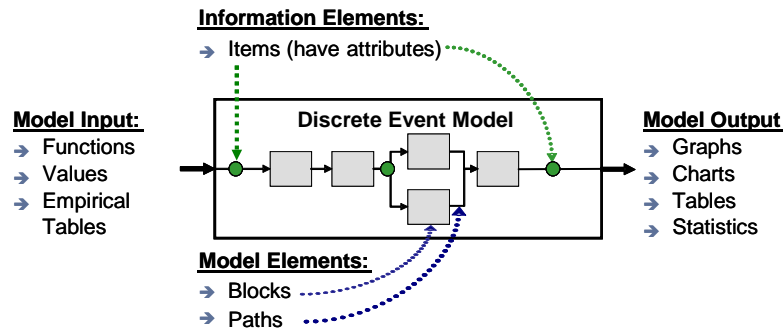


*Figure 30:     Key elements of DES Models [Mueller 2006]*

**Model Inputs:** These are values, functions or empirical tables that specify how items arrive and flow through the model. Inputs can be deterministic as well as stochastic. DES tools provide random number generators for various probability distributions.

**Model Elements:** A DES model contains several elements: *blocks*, *paths*, and *items*. Items carry information in the form of attributes. The model is a network of interconnected blocks. Items are passed from block to block along predefined paths.

Several categories of blocks are used to process information in the simulation model:

- Activity blocks process the items. They update the attributes of items and/or consume resources and effort.
- Routing blocks determine the flow of items when there are alternative paths and separate or combined streams of items. With the assistance of attribute blocks, routing blocks determine the path that an item takes.
- Queue blocks hold and store items temporarily.
- Resource blocks reflect resource limits in the simulation model. They allow the modeling of bottlenecks in the process.

**Choosing Paths in DES:** DES provides the concept of routing to model different process and path alternatives (see Figure 31). Routing determines which path an item takes when several alternatives exist. The decision can be made based on a fixed decision rule (e.g., 30% path A, 70% path B) or dynamically based on the item attributes.
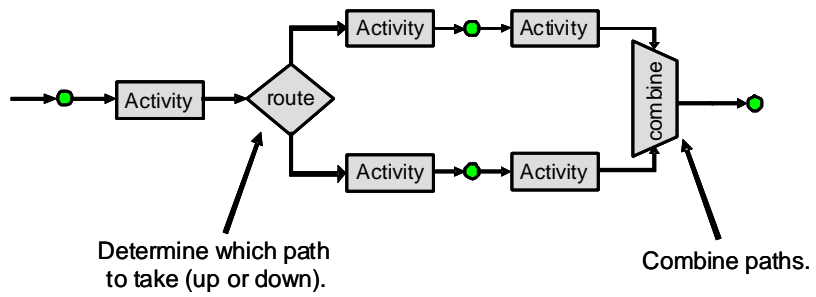
*Figure 31:*     *Routing in DES Models [Mueller 2006]*

# Appendix C: Brief Overview of the Central Literature

Substantial literature is available regarding both simulation and software processes. The literature associated with the intersection of these two fields (PSIM) is much smaller. The main work has been presented and published through the ProSim workshops (www.prosim.pdx.edu) and published through special journal issues in *Software Process: Improvement and Practice* (http://www3.interscience.wiley.com/cgi-bin/jhome/15482) and the *Journal of Systems and Software* (http://www.elsevier.com). Other journals such as *Information and Software Technology* (http://www.elsevier.com), *Software Quality Journal* (http://www.springer.com), *IEEE Software* (http://www.computer.org/portal/site/software/), and others have also carried articles on PSIM within the last five years.

In this appendix, we summarize the scope of the work that has been done and the types of applications that have been reported in the literature. This overview should give the industrial practitioner a sense for the capabilities and maturity of the PSIM field. Section C.1 describes the types of issues or problems that PSIM can be used to address, Section C.2 summarizes the scope of available PSIMs, and Section C.3 summarizes the development of PSIM over time.

## C.1 ISSUES ADDRESSED BY PSIMS

Table 34 presents an overview of the problem domains to which PSIMs have been applied and the key questions addressed within those domains.

*Table 34: PSIM Application Problem Domains and Questions*

| Problem Domain | Questions/Issues Being Addressed Using PSIM |
|---|---|
| Strategic Management (See examples in Sections 3.3, 3.7, and 3.9.) | • What is the impact of staffing policies on development cost, quality, and schedule?<br>• Should the timing of release of systems based upon product quality?<br>• What sites should be involved in development work?<br>• What sites should be involved in testing the product?<br>• What is the best work transfer strategy for a given software supply chain?<br>• What is the most cost effective verification and validation (V&V) or independent verification and validation (IV&V) strategy for a project?<br>• What decisions are relevant to the implications of system evolution?<br>• Acquisition management questions regarding cost, quality, and schedule in context of a variety of suppliers and partner organizations working together towards a set of deliverables |
| Project Planning (See examples in Sections 3.3, 3.4, and 3.7.) | • Which tailoring options should be used by a project? What is the ROI? What is the risk?<br>• How can these processes be optimized? |

| | |
|---|---|
| Cost Estimating (See example in Section 3.2.) | • What is the expected cost and schedule for this project at the pre-contract phase?<br>• What is the expected cost and schedule for this project at other points in the project?<br>• What is the risk associated with these estimates? |
| Project Management and Control (See example in Section 3.6.) | • What is the expected performance of the project?<br>• What is a realistic expectation for the performance of the project?<br>• What is the probability that the project will perform as planned?<br>• Is corrective action needed?<br>• What is the expected impact of the corrective action?<br>• What is the expected impact of other options?<br>• What corrective action is best?<br>• At what level must we perform early in the project in order to achieve management set targets? |
| Process Improvement (See examples in sections 3.3, 3.4, 3.5, and 3.7.) | • What is the likely performance impact of a new process change on a specific software development process?<br>• How robust is a proposed solution to changes in environmental and other uncontrollable factors? |
| Technology Adoption (See example in Section 3.5.) | • What is the likely impact of applying a new technology? What is the risk?<br>• When is a new technology useful? When is it useless?<br>• What is the anticipated ROI for a new technology?<br>• How effective does a new technology need to be in order to provide a financial benefit (positive ROI)? |
| Process Understanding and Documentation (See example in Section 3.1.) | • Which aspects of the process provide the most leverage in terms of making process improvements?<br>• How can the graphical and textual capabilities of PSIMs be used to provide PAL support?<br>• Which processes cause the greatest impact to overall cost, schedule, and quality (absolute performance)?<br>• Which processes drive the greatest variation in the overall cost, schedule, and quality (uncertainty in performance)? |
| Training and Learning (See example in Section 3.8, although most other PSIM applications may also be used for this purpose.) | • What is the impact of using simulations to help train managers about process tradeoffs and process simulation games?<br>• What is the impact of various assumptions regarding learning curves for newly introduced process and technology changes? |

**C.2    SCOPE OF PSIMS**

We define PSIM scope by the extent of the full development life cycle modeled, the type of life-cycle models (i.e., waterfall, incremental, product-line), other key components of models, and model performance measures and inputs.

PSIMs have been developed that represent the full development life-cycle process of the project from conception to release. Some models have even focused on release decisions and post-release impact. Often, major portions of system dynamics and hybrid models contain components that address project level factors such as human resource issues related to hiring, firing, learning, training, motivation, and schedule pressure, that impact productivity over time. Recent work in developing globally distributed development models has also included factors dealing with communication, culture, language issues, time zone differences, trust, and work transfer strategies, among others.

The types of life-cycle models that have been simulated include

- waterfall
- IEEE 12207
- incremental
- incremental spiral
- product line development
- agile (XP)
- open source
- acquisition
- globally distributed development

Performance measures commonly included in PSIM models include

- effort/cost
- cycle time (i.e., interval, duration, schedule)
- defect levels at various stages
- staffing requirements
- staff utilization rate
- cost/benefit, return on investment, NPV
- risk (probability of negative outcome)
- throughput/productivity
- queue lengths (backlogs)

PSIM models require input data in order to predict the above performance measures. Input data can be derived from process documents and assessments, the organization's current baseline figures used for project planning, exemplary projects, pilot projects, expert opinion (especially for SD models), and industry data for comparable organizations. Commonly used data for PSIMs include the following:

- amount of incoming work

- effort based on size (and/or other factors)

- defect detection efficiency

- effort for rework based on size and number of defects

- defect injection, detection, and removal rates

- decision point outcomes; number of rework cycles

- staff hiring rates and turnover rates, by category

- personnel capability/skills, motivation, and productivity; over time, and/or as a function of schedule pressure and other considerations

- resource availability and constraints

- frequency of product version releases

## C.3  SUMMARY OF THE PSIM DEVELOPMENT TIMELINE

- Kellner and Hansen developed a state-based model of a process to update document based on software changes [Kellner 1988]. This model forms the basis of an analysis of opportunities to apply technology to the process. Their model demonstrated the feasibility and value of process modeling as a way to systematically analyze process changes using models.

- Humphrey and Kellner proposed a transformational entity-based model [Humphrey 1989]. Such a model, it was felt, could effectively deal with frequent changes and rework that are typical of real-world software development processes. Their state-based model enabled the representation of multiple concurrent activities.

- Abdel-Hamid described a simulation model of a generic software project in 1991 that initiated the application of simulation to software processes [Abdel-Hamid 1991]. This model contained six main subcomponents: (1) Human Resources, (2) Workforce Allocation, (3) Quality Assurance, (4) Productivity, (5) Software Development, and (6) Project Control. The model estimated project cost and schedule. Other researchers have extended the main model components using the SD paradigm. The main strength of this model is the variety of factors that were incorporated into the model. The development process was not explicitly modeled. Subsequent developments included the following:
  - Madachy added process detail to the Abdel-Hamid and Madnick model to better estimate schedule and cost [Madachy 1994].
  - Madachy provided an SD model focused on software inspection processes [Madachy 1996].
  - Raffo proposed a method to evaluate process improvements called the Process Tradeoff Analysis Method (PTAM) [Raffo 1996]. PTAM provided a systematic approach for developing a quantitative business case for process improvement decisions including calculation of ROI and risk using PSIM. Raffo also developed a full incremental life-cycle PSIM using the SBS paradigm in the Statemate simulation environment by iLogix.
  - Sycamore applied SD modeling to software project management, with a focus on project scheduling [Sycamore 1996].
  - Tvedt used SD modeling to evaluate the impact of process improvements on the software development cycle time [Tvedt 1996].

- Rus used SD modeling to evaluate strategies for software quality and reliability engineering [Rus 1998].
- Plekhanova asserted that one could predict, define, and improve software process performance by considering individual capabilities [Plekhanova 1999]. The paper views human resources as a critical variable in the software process model and presents a novel approach for scheduling resource-constrained processes that considers team and/or individual skills/knowledge/capabilities.
- Collofello and Roehling analyzed the motivations for software outsourcing, including the potential for cost reduction, faster development time, and the increased availability of software engineering talent, using a simulation model to illustrate the dynamics, potential benefits, and potential drawbacks [Roehling 2000]. The paper also discussed the applicability, usefulness, practical benefits, and rationale for using simulation models to address the complex challenges associated with outsourcing.
- Kellner, Madachy, and Raffo summarized the PSIM field as it had developed to that point and put the available research into context [Kellner 1999]. They explained why an organization might want to consider adopting this technology and how it might apply process simulation. A table was provided that presented the scope of the various models versus the problems addressed, based on six broad problem areas for which PSIM has been used.
- Drappa and Ludewig developed a simulation model to be used for training of software engineers on process and project issues in which students act as project managers to see the impact of alternative development strategies [Drappa 2000].
- Houston reported on the use of simulation models to assess the effects of six common and significant risk factors on software development projects [Houston 2000]. His model was designed to represent the risk management activities of assessment, mitigation, contingency planning, and intervention.
- Powell used a measurement system in conjunction with an SD model to examine the impact of time-constrained software development on project results [Powell 2001].
- Pfahl advocated using SD models to increase understanding of the development process and provided a systematic method to build SD models for software processes [Pfahl 2001]. The work focused on using simulation to help project managers learn.
- Martin used a hybrid technique to model the software development process. He combined Abdel-Hamid and Madnick's SD model with Raffo's discrete event model and applied the resulting hybrid model to study processes used by a leading aerospace software development firm [Martin 2002].
- Berling and colleagues proposed the use of SD to investigate the relationship between defect prevention in development phases and defect detection in test phases [Berling 2003].
- Raffo and Setamanit created a bi-directional simulation model. This model estimated specification limits for acceptable project performance during early project life-cycle phases for key project performance measures. This model enables managers to know when the project may be going off track early in the project—even if the process may **appear** to be performing at consistent or acceptable levels [Raffo 2003].

- Ramil and Smith introduced qualitative simulation applied to software process simulation [Ramil 2003].

- Williams focused on the requirements engineering phase of the software development life cycle [Williams 2003]. An SD model was provided that incorporated metrics proposed by Costello and Liu and Davis and colleagues to facilitate process measurement and the prediction of cost, schedule, quality, and customer satisfaction [Costello 1995, Davis 1993].

- Häberlein provided an SD-oriented framework that captures the causal structures common to models of software acquisition and then used the framework to explore alternative acquisition strategies [Häberlein 2004].

- Birkhoelzer and colleagues provided a model with 15 process areas synthesized from CMMI that help to classify process states and potential areas of investment. Twenty-seven business performance indicators form the outputs. The model is capable of reflecting underlying strategies for advancing or maintaining an organization's processes. The iterative and interactive investment-oriented approach, along with the graphical presentation of results vs. historical patterns, can give users insight into the complex process dynamics and interdependencies. The simulator can serve as a tool to enhance the appreciation of software engineering practices [Birkhoelzer 2005].

- Raffo created a quantitative Project Management of Process Tradeoffs (PROMPT) framework for quantitative project management. This work utilizes snapshots of current project data to update a project PSIM that tracks performance. If predicted performance goes outside of the statistical or specified ranges, corrective action may be necessary. The simulation model can then be used to plot the best course to bring the project back on track [Raffo 2005a].

- Mizell extended Raffo's IEEE 12207-based model to create an incremental spiral model that combined SD components [Mizell 2006]. The model was used to estimate project cost and schedule at the concept evaluation phase as well as later in the development life cycle for a large-scale NASA project.

- Mueller created a product-line software development model in a software factory development environment using DES [Mueller 2006]. The work extended previous models by adding non-terminating simulations and a new size index measure. The model was applied at a leading firm in the automotive industry.

- Setamanit, Wakeland, and Raffo created a hybrid simulation for distributed software development. The model incorporated over a dozen key factors associated with multisite, cross-organizational, cross-cultural development [Setamanit 2006]. Each site in the supply chain is modeled explicitly, as are handoffs of work between sites. The model was applied at a global software development organization developing software in Asia and the United States.

# References

[Abdel-Hamid 1991]
Abdel-Hamid, Tarek K. & Madnick, Stuart E. *Software Project Dynamics: An Integrated Approach*. Englewood Cliffs, N.J.: Prentice Hall (ISBN: 0138220409).

[Bass 2004]
Bass, M. & Paulish, D. "Global Software Development Process Research at Siemens," 8-11. *The 3rd International Workshop on Global Software Development*. Edinburgh, Scotland, 2004.

[Behrman 2003]
Behrman, R. & Carley, K. "Modeling the Structure and Effectiveness of Intelligence Organizations: Dynamic Information Flow Simulation." *8th International Command and Control Research and Technology Symposium*. Washington, DC, 2003.
http://www.casos.cs.cmu.edu/publications/papers/dynamic_info_sim.pdf

[Berling 2003]
Berling, T., Andersson, C., Höst, M., & Nyberg, C. "Adaptation of a Simulation Model Template for Testing to an Industrial Project." *Proceedings of the ProSim 2003 Conference*. Portland, Oregon, May 3-4, 2003.

[Birkhoelzer 2005]
Birkhoelzer, Thomas, Navarro, Emily Oh, & Hoek, André van der. "Teaching by Modeling Instead of by Models." *ProSim 2005*. St. Louis, Missouri, May 14-15, 2005.

[Boehm 2000]
Boehm, Barry, Horowitz, Ellis, Madachy, Ray, Reifer, Donald, Bradford K. Clark, Steece, Bert, Brown, A. Winsor, Chulani, Sunita, & Abts, Chris. *Software Cost Estimation with COCOMO II*. Upper Saddle River, NJ: Prentice Hall, 2000 (ISBN: 0130266922).

[Boehm 1981]
Boehm, Barry W. *Software Engineering Economics*. Englewood Cliffs, N.J.: Prentice-Hall, 1981 (ISBN: 0138221227).

[Carley 1991]
Carley, K. "Designing Organizational Structures to Cope with Communication Breakdowns: A Simulation Model." *Organization & Environment, Vol. 5,* 1 (1991): 19-57.

[Carmel 1999]
Carmel, Erran. *Global Software Teams: Collaborating Across Borders and Time Zones*. Upper Saddle River, NJ: Prentice Hall (ISBN: 013924218X).

[Carmel 2005]
Carmel, Erran & Tjia, Paul. *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*. Cambridge: Cambridge University Press (ISBN: 0521843553).
http://www.loc.gov/catdir/enhancements/fy0668/2006619526-d.html

[CeBASE 2003]
CeBASE, Center for Empirically Based Software Engineering. "eWorkshop on Software Inspections and Pair Programming." December 16, 2003. http://www.cebase.org/www/frames.html?/www/AboutCeBASE/News/index.html

[CeBASE 2005]
CeBASE, Center for Empirically Based Software Engineering. "Software Engineering Laboratory Data." 2005. http://www.cebase.org

[Costello 1995]
Costello, R. J. & Liu, D. B. "Metrics for Requirements Engineering." *The Journal of Systems and Software, 29,* 1 (1995): 39-63.

[CSE 2004]
CSE, Center for Systems and Software Engineering. "COCOMO: Constructive Cost Modeling." http://csse.usc.edu/research/COCOMOII/

[Curtis 1988]
Curtis, B., Krasner, H., & Iscoe, N. " A Field Study of the Software Design Process for Large Systems." *Communications of the ACM, 31,* 11 (1988): 1268-1287.

[Davis 1993]
Davis, A., Overmeyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A., Kincaid, G., Ledeboer, G., Reynolds, P., Sitaram, P., Ta, A., & Theofanos, M. "Identifying and Measuring Quality in a Software Requirements Specification," 141-152 *1st International Symposium on Software Metrics*. London, UK

[Drappa 2000]
Drappa, A. & Ludewig, J. "Simulation in Software Engineering Training," 199-208. *Proceedings of the 22nd International Conference on Software Engineering*. Limerick, Ireland, 2000.

[Ermakov 1995]
Ermakov, S. M. & Melas, V. B. *Design and Analysis of Simulation Experiments*. Dordrecht, Boston: Kluwer Academic Publishers (ISBN: 0792336623).

[Fagan 1986]
Fagan, M. E. "Advances in Software Inspections." *IEEE Transactions on Software Engineering, SE-12,* 7 (July 1986).

[Ferguson 2006]
Ferguson, R.W. & Lami, G. "An Empirical Study on the Relationship between Defective Requirements and Test Failures," 7-10 *Proceedings of the NASA/IEEE Software Engineering Workshop* Columbia, Maryland, IEEE Computer Society, 2006.

[Galorath Incorporated 2004]
Galorath Incorporated. "SEER Tools: Advanced Modeling Tools to Estimate, Plan and Control Projects." http://www.galorath.com/

[Grant 1990]
Grant, Eugene Lodewick, Ireson, William Grant, & Leavenworth, Richard S. *Principles of Engineering Economy*. New York: Wiley (ISBN: 047163526X). http://www.loc.gov/catdir/description/wiley033/89033571.html

[Häberlein 2004]
Häberlein, T. "Understanding Software Acquisition through Modeling and Simulation." PhD diss., University of Ulm, 2004.

[Harrison 1999]
Harrison, W., Raffo, D., Settle, J., & Eickelmann, N. "Adapting Financial Measures: Making a Business Case for Software Process Improvement." *Software Quality Journal, 8,* 3 (1999).

[Harrison 2000]
Harrison, W., Raffo, D. M., & Vandeville, J. "A Universal Metrics Repository," 803-809 *Proceedings of the Pacific Northwest Software Quality Conference*. Portland, Oregon, October 18-19, 2000. http://portal.acm.org/ft_gateway.cfm?id=568897&type=pdf&coll=GUIDE&dl=GUIDE,ACM&CFID=42505212&CFTOKEN=61228939

[Herbsleb 1999]
Herbsleb, J. D. & Grinter, R. E. "Splitting the Organization and Integrating the Code: Conway's Law Revisited," 85-95. *Proceedings of the International Conference on Software Engineering (ICSE'99)*. Los Angeles, CA, ACM Press, 1999.

[Herbsleb 2001]
Herbsleb, J. D. & Moitra, D. "Global Software Development." *IEEE Software, March/April* (2001): 16-20.

[Houston 2000]
Houston, D. "A Software Project Simulation Model for Risk Management." PhD diss., Arizona State University, 2000.

[Humphrey 1989]
Humphrey, W. & Kellner, M. "Software Process Modeling: Principles of Entity Process Models," 331-342. *Proceedings of the 11th International Conference on Software Engineering*. Pittsburgh, PA, ACM, 1989.

[IEEE 1998]
IEEE. "Industry Implementation of International Standard ISCO/IEC 12207: Standard for Information Technology - Software Life Cycle Processes."

[Jones 1991]
Jones, Capers. *Applied Software Measurement: Assuring Productivity and Quality*. New York: McGraw-Hill, 1991 (ISBN: 0070328137).

[Jones 1998]
Jones, Capers. *Estimating Software Costs: Bringing Realism to Estimating*. New York: McGraw-Hill Companies (ISBN: 0071483004, 9780071483001). http://www.loc.gov/catdir/toc/ecip0714/2007011911.html

[Jones 2000]
Jones, Capers. *Software Assessments, Benchmarks, and Best Practices*. Boston, Mass.: Addison Wesley, 2000 (ISBN: 0201485427).

[Jones 2007]
Jones, Capers. *Estimating Software Costs: Bringing Realism to Estimating*. New York: McGraw-Hill Companies (ISBN: 0071483004, 9780071483001).
http://www.loc.gov/catdir/toc/ecip0714/2007011911.html

[Kellner 1999]
Kellner, M., Madachy, R., & Raffo, D. "Software Process Modeling and Simulation: Why, What, How." *Journal of Systems and Software, 46,* 2/3 (15 April 1999).

[Lami 2005a]
Lami, G., Gnesi, S., Fabbrini, F., Fusani, M., & Trentanni, G. "An Automatic Tool for the Analysis of Natural Language Requirements." *International Journal of Computer Systems Science and Engineering, Special Issue on Automated Tools for Requirement Engineering, 20,* 1 (January 2005).

[Lami 2005b]
Lami, Giuseppe. *QuARS: A Tool for Analyzing Requirements* (CMU/SEI-2005-TR-014, ADA441307). Software Engineering Institute, Carnegie Mellon University, 2005.
http://www.sei.cmu.edu/publications/documents/05.reports/05tr014.html

[Landis 1990]
Landis, L., McGarry, F. E., Waligora, S., Agresti, W., Card, D., Page, J., Church, V., & Werking, R. *Manager's Handbook for Software Development (Revision 1)* (SEL-84-101). Software Engineering Laboratory Series, NASA Goddard Space Flight Center, 1990.

[Law 2000]
Law, Averill M. *Simulation Modeling and Analysis*. Boston: McGraw-Hill, 2000 (ISBN: 9780072988437, 0072988436). http://www.loc.gov/catdir/toc/ecip0611/2006010073.html

[Leffingwell 2000]
Leffingwell, Dean & Widrig, Don. *Managing Software Requirements: A Unified Approach*. Reading, MA: Addison-Wesley, 2000 (ISBN: 0201615932).

[Madachy 1994]
Madachy, R. J. "A Software Project Dynamics Model for Process Cost, Schedule, and Risk Assessment." PhD diss., University of Southern California, 1994.

[Madachy 1996]
Madachy, R. "System Dynamics Modeling of an Inspection-Based Process." *Proceedings of the 18th International Conference on Software Engineering*. Berlin, Germany, March 25 - 29, 1996.

[Martin 2002]
Martin, R. H. "A Hybrid Model of the Software Development Process." PhD diss., Portland State University, 2002.

[Mizell 2006]
Mizell, C. "Quantitative Assessment of Software Development Project Management Issues Using Process Simulation Modeling With System Dynamics Elements." PhD diss., University of Central Florida, 2006.

[Mueller 2006]
Mueller, M. "Analyzing Software Quality Assurance Strategies through Simulation: Development and Empirical Validation of a Simulation Model in an Industrial Software Product Line Organization." PhD diss., Beim Fachbereich Informatik der Technischen Universität Kaiserslautern, 2006.

[Navaro 2005]
Navaro, E. O. & van der Hoek, A. "Software Process Modeling for an Educational Software Engineering Simulation Game." *Software Process Improvement and Practice, 10,* 3 (July/September 2005): 311-325.

[Pfahl 2001]
Pfahl, D. "An Integrated Approach to Simulation-Based Learning in Support of Strategic and Project Management in Software Organisations." PhD diss., Fraunhofer IRB Verlag, Fraunhofer IESE, 2001.

[Pfahl 2004]
Pfahl, D., Laitenbergerb, O., Ruhe, G., Dorschd, J., & Krivobokovae, T. "Evaluating the Learning Effectiveness of Using Simulations in Software Project Management Education: Results from a Twice Replicated Experiment." *Information and Software Technology 46,* 2 (February 2004): 127–147.

[Plekhanova 1999]
Plekhanova, V. "A Capability-Based Approach to Software Process Modelling." *Proceedings of the ProSim'99 Workshop*. Silver Falls, OR, June 27-29, 1999.

[Powell 2001]
Powell, A. "Right on Time: Measuring, Modelling and Managing Time-Constrained Software Development." PhD diss., Univ. of York, 2001.

[Pyzdek 2003]
Pyzdek, Thomas. *The Six Sigma Handbook: A Complete Guide for Green Belts, Black Belts, and Managers at All Levels*. New York: McGraw-Hill (ISBN: 0071410155). http://www.loc.gov/catdir/description/mh041/2003283371.html

[QSM 2007]
QSM. "SLIM: Software Cost Estimation and Project Management." 2007. http://www.qsm.com/

[Raffo 1996]
Raffo, D. M. "Modeling Software Processes Quantitatively and Assessing the Impact of Potential Process Changes on Process Performance." PhD diss., Carnegie Mellon University, 1996.

[Raffo 2003]
Raffo, D. & Setamanit:, Siri-on. "Supporting Software Process Decisions Using Bi-Directional Simulation." *International Journal of Software Engineering and Knowledge Engineering, 13,* 5 (October 2003).

[Raffo 2005a]
Raffo, D. "Software Project Management Using PROMPT: A Hybrid Metrics, Modeling and Utility Framework." *Journal of Information, Software and Technology, 47,* 15 (December 2005): 1009-1017.

[Raffo 2005b]
Raffo, D. & Menzies, M. "Evaluating the Impact of a New Technology Using Simulation: The Case for Mining Software Repositories," 57-62. *Proceedings of the ProSim 2005 Conference.* St. Louis, MO, May 14-15, 2005.

[Raffo 2005c]
Raffo, D., Nayak, U., & Wakeland, W. "Implementing Generalized Process Simulation Models," 139-143. *Proceedings of the ProSim 2005 Conference.* St. Louis, MO, May 14-15, 2005.

[Ramil 2003]
Ramil, J. & Smith, N. "Qualitative Simulation of Models of Software Evolution." *Software Process: Improvement and Practice, 7,* 3-4 (2003): 95-112.

[Roehling 2000]
Roehling, Stephen T., Collofello, James S., Hermann, Brian G., & Smith-Daniels, Dwight E. "System Dynamics Modeling Applied to Software Outsourcing Decision Support." *Software Process: Improvement and Practice, 5,* 2-3 (2000): 169 - 182.

[Rus 1998]
Rus, I. "Modeling the Impact on Cost and Schedule of Software Quality Engineering Practices." PhD diss., Arizona State University, 1998.

[Scacchi 1990]
Scacchi, W. & Mi, P. "A Knowledge-Based Environment for Modeling and Simulating Software Engineering Processes." *IEEE Transactions on Knowledge and Data Engineering, 2,* 3 (September 1990).

[Scacchi 1999]
Scacchi, W. "Experience with Software Process Simulation and Modeling." *Journal of Systems and Software, 46,* 2-3 (April 1999): 183-192.

[SEI 2006]
SEI, Software Engineering Institute. "Capability Maturity Model Integration (CMMI)." http://www.sei.cmu.edu/cmmi/

[SEL 1993]
SEL. *Cost and Schedule Estimation Study Report* (SEL-93-002). Software Engineering Laboratory Series, NASA Goddard Space Flight Center, 1993.

[SEL 1994]
SEL. *An Overview of the Software Engineering Laboratory* (SEL-94-005). Software Engineering Laboratory Series, NASA Goddard Space Flight Center, 1994.

[SEL 1997]
SEL. *Guide to Software Engineering Laboratory Data Collection and Reporting* (SEL-97-001). Software Engineering Laboratory Series, NASA Goddard Space Flight Center, 1997.

[Setamanit 2006]
Setamanit, W. & Raffo, D. "Planning and Improving Global Software Development Processes Using Simulation." *Proceedings of the Workshop on Global Software Development Practice.* Shanghai, China, May 23, 2006.

[Smith 2005]
Smith, N., Capiluppi, A., & Ramil, J. "A Study of Open Source Software Evolution Data Using Qualitative Simulation." *Software Process Improvement and Practice, 10,* 3 (July/September 2005): 287-300.

[Software Productivity Research 2007]
Software Productivity Research. "Software Productivity Research (SPR) KnowledgePLAN, The Leading Estimation Software Solution." http://www.spr.com/estimation/default.shtm

[Stamatis 2001]
Stamatis, D. H. *Six Sigma and Beyond.* Boca Raton, FL: St. Lucie Press (ISBN: 157444316X). http://www.loc.gov/catdir/enhancements/fy0646/2001041635-d.html

[Sycamore 1996]
Sycamore, D., M. "Improving Software Project Management Through System Dynamics Modeling." PhD diss., Arizona State University, 1996.

[Tvedt 1996]
Tvedt, J. D. "An Extensible Model for Evaluating the Impact of Process Improvements on Software Development Cycle Time." PhD diss., Arizona State University, 1996.

[Williams 2003]
Williams, D. "Challenges of System Dynamics to Deliver Requirements Engineering Projects: Faster, Better and Cheaper." http://www.systemdynamics.org/conf2003/proceed/PAPERS/400.pdf

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

| 1. AGENCY USE ONLY (Leave Blank) | 2. REPORT DATE January 2008 | 3. REPORT TYPE AND DATES COVERED Final |
|---|---|---|

| 4. TITLE AND SUBTITLE | 5. FUNDING NUMBERS |
|---|---|
| Moving Up the CMMI Process Maturity Levels Using Simulation | FA8721-05-C-0003 |

**6. AUTHOR(S)**

David M. Raffo, Ph.D. & Wayne Wakeland, Ph.D.

| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) | 8. PERFORMING ORGANIZATION REPORT NUMBER |
|---|---|
| Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213 | CMU/SEI-2008-TR-002 |

| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER |
|---|---|
| HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116 | ESC-TR-2008-002 |

**11. SUPPLEMENTARY NOTES**

| 12A. DISTRIBUTION/AVAILABILITY STATEMENT | 12B. DISTRIBUTION CODE |
|---|---|
| Unclassified/Unlimited, DTIC, NTIS | |

**13. ABSTRACT (MAXIMUM 200 WORDS)**

Process Simulation Modeling (PSIM) technology can be used to evaluate issues related to process strategy, process improvement, technology and tool adoption, project management and control, and process design. Recent developments in PSIM tools have drastically cut the costs to develop models for evaluating such issues, and new methods have been developed to apply PSIM, enabling it to provide greater business value. At the same time, trends within the software industry towards improving operations and reducing costs have heightened the need for tools to better plan and manage processes. As a result, organizations regard PSIM as an attractive tool that can provide business value today. This report shows examples of how PSIM has been implemented within industry and government organizations to improve process consistency and results. The report also shows, via many examples, exactly how PSIM supports Capability Maturity Model Integration Process Areas from level 2 through level 5.

| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| Process Simulation Modeling, PSIM, Capability Maturity Model Integration Process Areas, CMMI, process evaluation, process improvement, simulation methodologies, discrete event simulation model (DES) | 129 |

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified | 20. LIMITATION OF ABSTRACT UL |
|---|---|---|---|

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18
298-102