Software Engineering Institute

SEPG Europe 2012 Conference Proceedings

Jose Maria Garcia, Javier Garcia-Guzman, Javier Garzas, Amit Arun Javadekar, Pat Kirwan, Joaquin Lasheras, Fuensanta Medina-Dominguez, Erich Meier, Arturo Mora-Soto, Ana M. Moreno, Radouane Oudrhiri, Fabrizio Pellizzetti, Alejandro Ruiz-Robles, Maria-Isabel Sanchez-Segura, Prasad M. Shrasti, Aman Kumar Singhal

September 2012

SPECIAL REPORT
CMU/SEI-2012-SR-005

Software Engineering Process Management

http://www.sei.cmu.edu

Carnegie Mellon

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

This report was prepared for the
SEI Administrative Agent
AFLCMC/PZE
20 Schilling Circle, Bldg. 1305, 3rd floor
Hanscom AFB, MA 01731-2125

SEI markings v3.2/ DM-0000022 / 30 August 2011

Table of Contents

List of Figures

List of Tables

Abstract

The 17th SEPG Europe conference was organized by the Software Engineering Institute (SEI) in collaboration with *IEEE Software* magazine and took place from June 5–7, 2012 in Madrid, Spain. SEPG Europe's goal is to bring together software and systems professionals who have a common passion to improve the processes they use to create the products and services they develop. This year, collaboration between SEPG Europe 2012 and *IEEE Software* magazine gave technical session presenters the chance to have a paper selected by the SEPG Europe 2012 Technical Committee for inclusion in this SEI report, as well the opportunity to be considered for inclusion in a future issue of *IEEE Software* magazine. This report contains the seven papers selected for publication by the SEPG Europe 2012 Technical Committee.

1 Introduction

Pat Kirwan, Software Engineering Institute

The 17th Annual SEPG Europe conference took place June 5–7, 2012 in Madrid, Spain. The conference was organized by the Software Engineering Institute (SEI) in association with *IEEE Software* magazine. The theme of the conference was a *¡A Passion for Process!*, which attracted over 160 delegates from 26 countries around the world.

The Technical Program Committee, composed of an international group of members, assembled the technical program from a record number of abstracts submitted by the software engineering community. The program included five engaging keynote speakers and technical tracks on multi-model process improvement, high maturity, agility, small settings, organizational change, risk and resilience, innovation, professional development, and more.

New in 2012, a collaboration between SEPG Europe 2012 and *IEEE Software* magazine gave technical session presenters the chance to have a paper selected by the SEPG Europe 2012 Technical Committee for inclusion in this report, as well the opportunity to be considered for inclusion in a future issue of *IEEE Software* magazine. An *IEEE Software* article review committee convened to determine the award for the best paper submission, and the announcement was made in the plenary session on Wednesday, June 6, 2012.

The winners of the best paper award were Radouane Oudrhiri and Fabrizio Pellizzetti for their paper **SPC, Six Sigma, and CMMI: Integration and Deployment Challenges**. This paper will be published in an upcoming issue of *IEEE Software*.

This report contains the seven papers selected for publication by the SEPG Europe 2012 Technical Committee. An overview of these papers is presented below.

Software Economics: A Framework for Process Improvement (by Fabrizio Pellizzetti and Radouane Oudrhiri) presents a synopsis of the state of the art in the software engineering industry in relation to the maturity of investment decision-making processes. It argues that the introduction of sound economic reasoning in software process improvement could help define the foundations of a true software improvement engineering discipline.

Influencing Change Management Decisions for Large Enterprise Transformation Programs (Prasad M. Shastri) identifies ten dynamic areas of organizational change and demonstrates how their correlation with solution enablers provides a strong qualitative relationship matrix for decision making.

Using CMMI for Software Improvement in Small Organizations. A Case Study (by Dr. Joaquín Lasheras, Dr. Javier Garzás, and José María García) examines whether CMMI is useful and practical for carrying out software process improvement efforts within small software organizations and discusses the experience of a pioneering project, IMPULSE CMMI, in the Murcia region of Spain.

Analysis of Coverage of CMMI Practices in Software Engineering Curricula (by Ana M. Moreno, Maria-Isabel Sanchez-Segura, and Fuensanta Medina-Dominguez) presents the results of a study of the qualifications of graduates of programs based on the international software engineering education standards, SE2004 and GSw2009, in implementing practices covered by the different CMMI-DEV process areas.

Estimation Competency Development for IT Project Managers: An Infosys Experience (by Amit Arun Javadekar and Aman Kumar Singhal) describes high-performance practices implemented to improve the estimation competency of various roles involved in project execution, management, sales, and quality assurance functions, in the context of accelerated growth, diverse talents, and the need for global reach and scalability.

Enhancing Process Asset Assessment (by Maria-Isabel Sanchez-Segura, Alejandro Ruiz-Robles, Arturo Mora-Soto, and Javier Garcia-Guzman) examines process assets from a strategic management perspective, shows that value must be determined by aligning process assets with business goals, and assesses how these assets contribute to the achievement of these goals.

The Economics of Process Management: Case Studies and Customer Experiences (by Erich Meier) presents three case studies based on actual customer experiences where economical approaches to process management were successfully deployed and yielded tangible business benefits.

2 Software Economics: A Framework for Process Improvement

Fabrizio Pellizzetti, Systonomy LTD, UK, fabrizio@systonomy.com

Radouane Oudrhiri, Systonomy LTD, UK, radouane@systonomy.com

2.1 Introduction

The objective of this paper is to provide an economics perspective on software and more precisely on the economic foundations of software process improvement (SPI).

Software product development, services delivery, process engineering activities (including software process improvement), and related technologies are investments that are supposed to create value for the stakeholders in the organization. Like any investment, they are based on a decision-making process.

This paper presents a synopsis of the state of the art in the software engineering industry in relation to the maturity of investment decision-making processes. It argues that the introduction of sound economic reasoning in SPI could help define the foundations of a true software improvement engineering discipline. This claim will be reinforced by practical approaches rooted in empirical and experimental studies, which will demonstrate the causal links between investments in change and process improvement, as well as examine the returns on such investments, which is a typical economic concern.

The remainder of the paper is structured as follows:

- Section 2.2 provides a general introduction to the topic of decision making and its relevance in the context of software engineering and SPI.
- Section 2.3 provides an overview of the economics models relevant to SPI.
- Section 2.4 highlights, through a simple case study, the approach to developing a business case for SPI, based on probabilistic approaches.
- Section 2.5 articulates the main implications for the SPI community.
- Section 2.6 outlines the main conclusions and recommendations for future work.

2.2 Software Process Improvement as Decision Making

Software engineering is, in essence, a communication and decision-making process, where different stakeholders constantly exchange information related to the system (and the process used to build or maintain the system) in order to make informed decisions.

These decisions affect the software organization at different levels:

- Product engineering and design: choices related to design and architecture, technologies, integration, interfaces and the evolution of software products
- Process improvement: choices related to the way software engineering professionals organize their collaborative work (i.e., the software process) and constantly improve its effectiveness and efficiency

The latter is the focus of this paper.

Any improvement, evolution, or adaptation is, by definition, a change or alteration to an existing situation. This is a decision-making process in the sense that it can be regarded as a cognitive process resulting in a selection of a course of action among several alternatives, in a situation of perennial uncertainty. For any improvement, there are always at least two alternatives: implementing the change leading to improvement (often with multiple scenarios to choose from) or maintaining the status quo (namely, "do nothing").

Every decision-making process produces a final output, in the form of a chosen action or opinion. As defined by Baker et al. in their 2002 study, "efficient decision-making involves a series of steps that require the input of information at different stages of the process, as well as a process for feedback" [Baker 2002]. Every decision is made up of a composite of information, data, facts, and beliefs. Data by itself does not constitute useful information, unless it is analyzed and processed within its context.

We therefore need to understand, analyze, assess, and measure the risks associated with our change and improvement decisions. In that sense, information has a value [Howard 1966], but the entire process of information acquisition, analysis, and management also has a cost, and it is possible to expect a return on investment from this process.

As a dual concept to the value of information, there is a concept that we call the "cost of ignorance" and cost of uncertainty [Pellizzetti 2011]. This is the cost of not detecting the need to change or investing in the wrong improvement initiative or solution—the cost associated with making the wrong decision.

Software improvement processes, business transformation processes, and decision-making processes generally depend on the availability of useful and reliable data, measures, and information. Bateson [Bateson 1972, Bateson 1979] defines information as *"...a difference which makes a difference..."*.

In SPI, there are two minimal sets of data about process performance and costs, which are essential to guide any decision-making process and drive process improvement toward delivering value.

- **Baseline Data**. The baseline is a "snapshot" of the process under study at a given point in time. It consists of both qualitative data and structural descriptions (e.g., the flow of activities or the satisfaction of process stakeholders) and quantitative data (e.g., measureable process attributes, including cost). Contrary to common beliefs, baseline data are not often readily available and may require launching targeted measurement programs that can obtain data in a reliable and useful format. Even software organizations at high levels of maturity may not collect systematic process performance measures and efficiency or cost metrics. In some cases, it is more economically sound to generate baseline data through experiments. And, finally, any baseline will necessarily consist of samples (any measurement is in fact a sampling activity) whose reliability heavily depends on the size and the amount of variation or noise in the data. Therefore baseline data are often estimates surrounded by uncertainty. Baseline data support improvement decisions by highlighting priorities that are based on the relative distribution of costs (i.e., the Pareto effect). However, the basic measures of process quality and cost for software processes (e.g., defect density, bug fixing costs, effort and duration of rework activities, duplication, and waste) often need to be developed through more in-depth analysis of the process and the organization. Software economic models can also be brought to bear in order to develop a meaningful baseline for the cost of poor quality (COPQ) and other costs. In many cases, improvement initiatives are launched regardless of the analysis of baseline cost data and without some fundamental cost benefit analysis.

- **Estimated Improvements and Benefits**. If we accept the definition that process improvement is an investment activity, any decision maker will expect estimates and projections of the performance improvement and cost reductions associated with the investment (i.e., the return on investment [ROI]). Even when a reliable baseline has been developed either through measurement, estimation, or experiments, predictions related to the expected improvements are often regarded as simply guesses, which they may often be. Forecasts about the future have very little meaning, unless a third type of measure is collected to further characterize our estimates, namely, the “quality” of the information used for developing these projections. The quality of information corresponds to our level of knowledge about a process and its behavior and the context in which the process is performed (the organization). This information has a cost, which is mainly the cost to study and analyze the process in qualitative terms (e.g., process modeling and analysis, stakeholders’ management) and to gather or generate (or generate through experiments) hard data related to key attributes of the process, such as capability, stability, resource consumption, and others. The expected benefit is in the form of higher quality information, which has its own value insofar as it supports more informed decision making and increases levels of confidence. In this sense, we define the “value of information” (and its dual concept “cost of ignorance”).

The cost of ignorance is the cost associated with wrong decisions that are made on the basis of unreliable, incomplete, irrelevant, or misleading information. In the context of SPI, this would be equal to the cost of investing improvement resources in projects that will not deliver business benefits or not changing a process on the basis that the need to do so goes undetected (loss of opportunity).

Quality initiatives are often driven by the perception that adherence to a set of so-called “best practices” (often selected following the latest trends and fashions in the industry) will, almost naturally, yield financial rewards. Unfortunately, this is not always the case.

From the perspective of a software economist, the foundation of an SPI discipline would necessarily need to satisfy a set of basic requirements for any improvement initiative.

- The baseline data for the process in question captures all the essential elements of cost of poor quality, the relative incidence of each category of cost (Pareto analysis) and allows a direct mapping of cost elements to process activities and resources.
- The quality of baseline data is measured. This includes the depth of analysis into process activities, the size of samples collected, the reliability of the data, and the mechanisms used to generate it (historical vs. experimental), as well as the availability of industry data to support or refute the findings from the analysis. If we are able to characterize and measure the quality of information available to guide our decisions, we can also develop the corresponding measure of risk associated with that decision.
- The level of confidence is measured and attached to our estimates. This is a function of the quality of the information available. A low confidence reflects a high risk factor for any decision and is directly related to the rigor followed and, again, the quality of information used in the estimation process. Most importantly, our estimates should be enriched by providing more contextual data about the estimate itself and should include not only a “range” of possible results, but also some appreciation of the probability of each possible outcome, and therefore a measure of the risk of not achieving given targets.

This is typically the case for benefits, usually in the form of ROI and net present value (NPV) figures.

The risk element (e.g., the risk of not delivering the expected ROI) should be one of the key drivers of business and improvement decisions, but the current state of the practice seems to contradict this basic rule. The only possible strategy for process improvement professionals is to quantify and reduce the risk of making decisions based on flawed information and data. As long as the cost of information (the cost to produce better quality information and reduce uncertainty) is lower than the expected potential impact of making the wrong decision (the cost of ignorance), this investment is worthwhile.

A software economics perspective will add this dimension to SPI decision making: the comparison between the cost of information and the cost of ignorance (or value and utility of the information) as a key determinant of improvement decisions.

2.3 Economics of SPI Projects

Process improvement is essentially a knowledge acquisition exercise where, through the analysis and study of the process, purposeful changes are introduced in order to achieve given improvement targets. If the knowledge gained does not improve our confidence in the changes that we intend to introduce, this information has little or no value. However, if we can generate useful data to reduce the risks associated with our decisions, this information is valuable and the cost to obtain this data may be justified.

If we attempt to formalize a simple economic model to support SPI decisions, a set of basic definitions and relationships ought to be developed and agreed upon. We propose the following initial taxonomy of concepts that are relevant to such models:

Cost of information: This is the cost to achieve (through measurement, experiments, or simulation) sufficient data and knowledge to aid decision making. The cost of information is a non-linear function of the effort invested in knowledge acquisition and the maturity of an organization’s measurement processes.

Value/utility of the information: This represents the increase in the level of confidence of our decisions. The value of information is a function of the expected ROI from an improvement initiative and the current level of uncertainty.

Risk appetite: This is the organization’s willingness to accept a certain level of risk. This is defined as a function of the importance of the improvement initiative, the expected payback (higher returns typically tolerate higher level of risk), and cultural factors. The risk appetite is measured by the level of confidence required by decision makers to approve an improvement initiative. This level of confidence is typically related to some measure of profitability of the initiative (ROI, NPV).

The diagram below provides a high-level view of the dynamics of the cost of information and the value or utility of the information, expressed as levels of confidence.

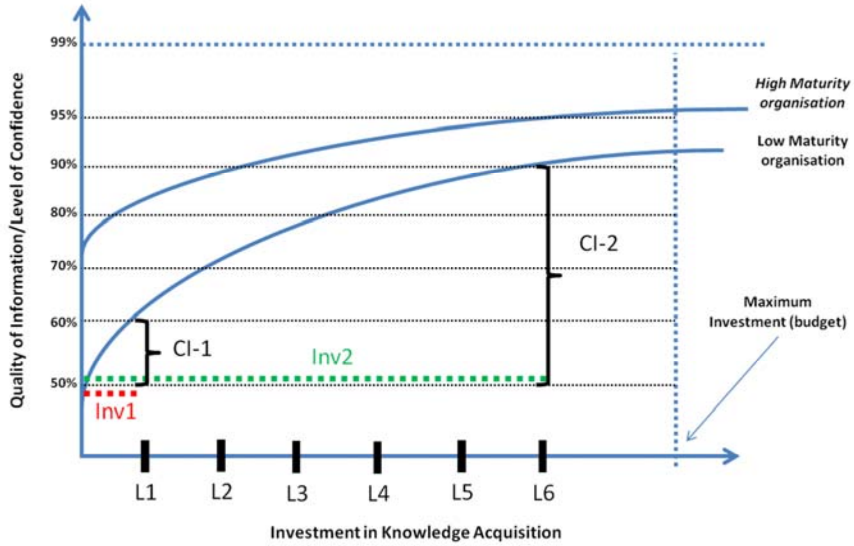

Figure 2.1: The Costs and Benefits of Information

If we plot the hypothetical cost/benefit curves of knowledge acquisition investments, we observe the following:

- Diminishing returns in confidence levels: the quality of information and level of confidence increases with additional investments in knowledge acquisition. This increase in confidence however occurs at decreasing rates; that is, the investment required to achieve higher levels of confidence grows more than proportionally.
- A natural limit exists to the maximum possible level of confidence in a decision. Since any estimate is subject to uncertainty and variability, anything short of having a clairvoyant in the project team will not allow the team to achieve confidence levels higher than 99%. Moreover, the costs to achieve such levels may be prohibitive as compared with the expected returns from the project.
- The elasticity of such curves (the strength of the association between investments in knowledge and increases in confidence) may change, depending on the maturity of the organization and the initial level of confidence.

In our simple model, a low maturity organization willing to move from an initial level of confidence of 50% to 60% (CI-1) would need to invest one unit (Inv. 1) to reach the investment level L1. However to achieve a much more comfortable level of 90% (CI-2), the investment required would increase to more than 6 units (Inv.2) and higher levels of confidence would exceed the maximum possible investment for the improvement project (e.g., the total budget). An organization at higher maturity with established measurement practices will probably start at a confidence level of around 70% and would need an additional investment of less than 3 units to reach a confidence level close to 90%.

The decision on how much additional investment in knowledge acquisition is needed becomes a problem of optimization along the tradeoffs that exist between the cost of information and the level of risk considered acceptable (the level of confidence).

2.4 Developing Measures of Risk and Confidence

Whilst the estimation of the cost of knowledge acquisition that is directly related to the effort spent in gathering, organizing, producing and validating process data may be a relatively straightforward exercise, the actual quantification of the risk (i.e., the level of confidence) requires further and more in-depth analysis. Assuming that an organization has defined its risk appetite at 10% (corresponding to a level of confidence at 90 percent), how can we measure the current level of risk we are taking when making a decision? This becomes the main factor that influences the investment decision in the SPI initiative.

A simple example will clarify the proposed approach.

Imagine a software organization that is facing growing pressure from customers to reduce the number of defects in every new release of the product. The organization does not necessarily have a formal process but follows an “incremental” and “iterative” approach. The company has limited resources to invest in additional testing and is constantly under pressure to deliver new features while struggling to catch up with fixing bugs from previous releases.

The company engaged an SPI specialist with a mandate from management to “fix the leak.” This specialist investigated the issue, performed an initial root cause analysis, and reported the following findings:

- Customers are right—the analysis of trends in the number of customer-reported defects confirms an increase in critical defects, release after release.
- The costs are unknown—developers and testers do not track the time spent fixing bugs—bug fixes and releases of new features are treated as equal, and some initial estimates from developers and testers suggests that the cost to fix a functional defect reported by a customer is in the range of £400 to £1800.
- No defect classification—the classification schema for defects is rudimentary and does not allow developers and testers to discriminate between different types of defects and their criticality.
- Diverse and diffused processes—different teams in the organizations perform different processes and the organization is at a low level of maturity. As a result, processes are neither documented nor enforced. Some teams rely heavily on unit testing and static code analysis and others manage to perform formal code reviews, while others simply bypass any early verification activity to deliver products on time to the sales team.

Key stakeholders recognized this issue as a real problem, but formal commitment from management was subject to an analysis of the expected ROI: a target of £350,000 NPV from the project was (arbitrarily) defined by management.

Based on this information, the SPI team developed an initial business case based on estimates of the current cost of poor quality (COPQ) and projecting the expected improvements associated with the introduction of different review techniques (unit testing, code inspections).

As a first step, SPI team members developed a structural model of the business case, where the different factors influencing the cost and benefit sides were decomposed to help the analysis of their relative contribution to the expected NPV. The structural view of the business case is presented in Figure 2.2.

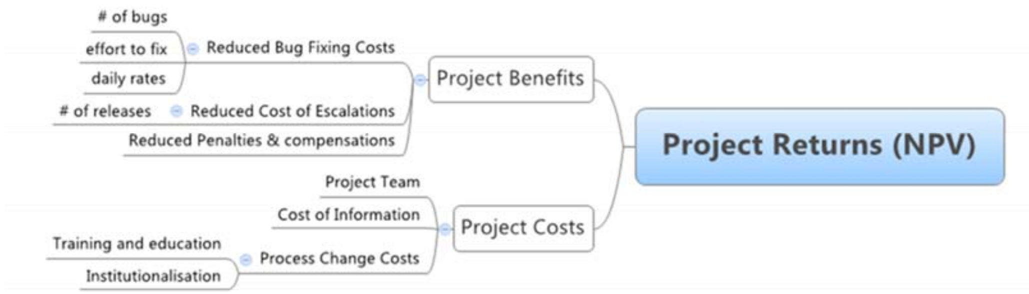

Figure 2.2: Structural View of the Business Case

Next, estimates for the expected NPV are developed using probabilistic approaches, such as assigning a probability distribution to every variable in the business case, based on the current knowledge about the problem. The use of probabilistic models is essential in order to increase the "visibility" of our current and desired levels of confidence.

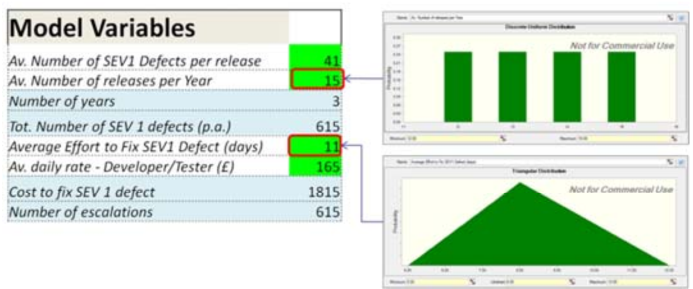

Figure 2.3: Model Parameter Distributions

Probability distributions reflect the real state of knowledge about the process and its key attributes: the larger the range of the estimate, the smaller the knowledge or confidence. A simple Monte Carlo simulation of the business case model will provide a measure of the level of confidence associated with the entire SPI initiative, and is shown in Figure 2.4

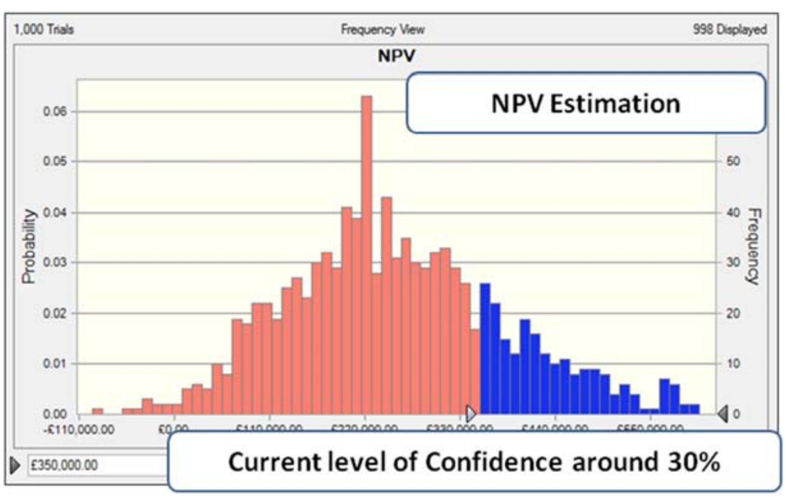

Figure 2.4: Monte Carlo Simulation

Once a target is set, the level of confidence with the outcome of the entire SPI initiative can be measured. In our example, an NPV of £350,000 or higher is associated with a level of confidence lower than 30%. Sensitivity analysis can be used to reveal the factors that contribute the most to this intolerable level of uncertainty and guide the decisions related to further investment in knowledge acquisition.

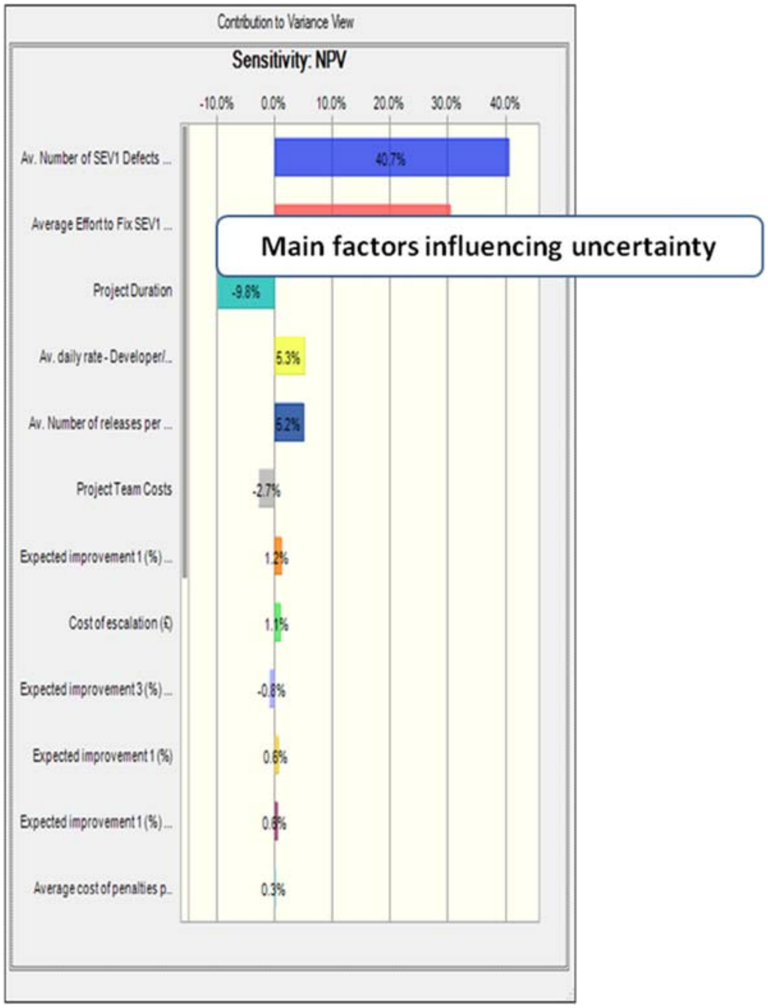

Figure 2.5: Sensitivity Analysis

The SPI team realized that the fragmented information about defects and their costs (ranging from £400 to £1800) and the expected improvement associated with code inspections (defect removal effectiveness increase and cost reduction) have the highest influence on the final range of results. With no empirical evidence to support them, the initial estimates are broad and range from a 5% to a 25% of increase in DRE across all defect types and a lower bug fixing cost (between £25 to £315 per functional defect).

When presented with an estimate at a 30% confidence level, managers are not entirely thrilled and ask the SPI team to review and improve its estimates. From previous studies of statistics, decision makers remember that 99% or 95% should be the target levels of confidence to make any informed decision. The SPI team quickly estimates the activities required to reduce uncertainty on these key factors, including more detailed analysis and re-classification of defects, more granular estimates of bug fixing costs, and performing and measuring the results of code inspections on a sample of projects in the form of controlled experiments.

The knowledge generated from these activities will definitely help to improve the quality of estimates. An additional investment of around £12,000 would, in the estimates of the SPI team, generate sufficient knowledge to make decisions at around 80-85% confidence (based on the simple economic model previously described).

However, to achieve a 95% confidence level, the investment would probably be in the region of £150,000, which would overshadow the potential returns from the entire initiative. Results are presented to management, and, after negotiation, it is agreed that the team will use a target of an NPV of £350,000 or more with a level of confidence around 80%. The business case is revisited with the additional information gathered, including, on the cost side, the investment in additional knowledge acquisition.

The teams have performed around 18 code reviews (a sample sufficient to perform some level of statistical analysis, at least on continuous variables). The defect databases have been consolidated and analyzed in detail. The range for the cost of repairing a defect has shrunk, through further analysis and experiments, to a more realistic £150 to £280. A new model is simulated, and the results are shared with managers.

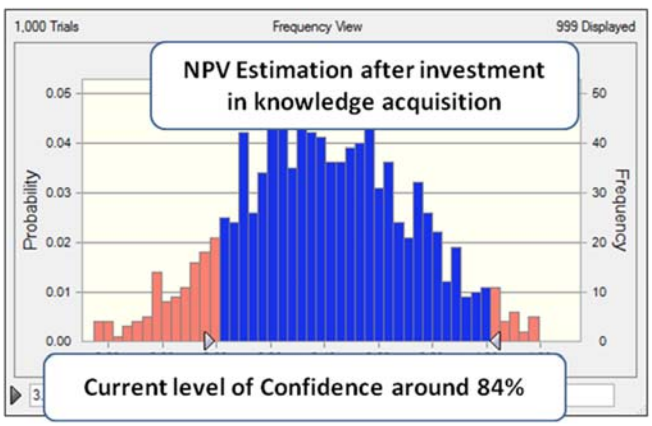

Figure 2.6: Simulation Results After Investment

The additional investment of £12,000 has generated an increase of 54% in the confidence level on expected returns higher than £350,000.

The organization has probably achieved a significant ROI in knowledge acquisition as well, which shows that the cost of information was justified by the increase in confidence.

2.5 Implications for the Software Process Improvement Community

Decision-making processes are subject to a universal rule stating that "the quality of decisions cannot consistently rise above the quality of information upon which those decisions are made." As a result, any decision-making process relies on information as the key input.

With modern levels of complexity in the systems we build and the sophistication of the processes we use to govern our engineering activities, complete and timely information is rarely available. Many decisions are based on fragmented evidence and the opinions and beliefs of decision makers. Furthermore, the degrees of freedom of our decisions (both related to the product and to the process) are often limited by technical and organizational constraints and the availability of resources.

In other words, decisions are often made in conditions of uncertainty, under constraints, and with very limited understanding of the risks (and consequences) of being wrong.

People tend to behave irrationally when faced with uncertainties. Savage provides an example of such irrational behavior [Savage 2009].

> *A man in a restaurant is trying to decide between the fried chicken and the roast beef. He is inclined toward the chicken, but asks the waiter one more question: "Do you also have duck today?" "Yes we do." responds the waiter. "Oh, in that case," says the man, "I'll have beef."*

Such irrational behavior is not only observable in restaurants, but is also common to many SPI decisions. The sophistication of decision making in software becomes, ultimately, the main determinant of the ability to deliver value. The introduction of sound economic reasoning in software project and risk management and SPI could drive a radical transformation in the industry and help define the foundations of a true software engineering discipline.

We have previously qualified process improvement and change management as decision-making processes, where available information is processed in the hope of making the right choice. While we can partially rely on cost models such as COCOMO or COCOMO II mainly for project management and cost estimation, we aim to develop basic, predictive models that can be used specifically for SPI.

Perhaps the single most important contribution of the economic thinking applied to SPI is in the application of risk management techniques to business cases and to articulate the notion that we have called the cost of ignorance or value of information. Keeping in mind the axiom that states that "the quality of decisions cannot consistently rise above the quality of the information upon which that decision is made," we can attribute almost the entire cost of wrong decisions to the poor quality of the information (or the ignorance) used to make such decisions. An unrealistic business case is often the principal contributor to many SPI initiatives failing to deliver the expected increase in value or failing to get the necessary stakeholder support.

If we were to develop a simplified "maturity model" for the SPI business case, we could define five levels of maturity, based on the sophistication of the estimation techniques used and the appreciation of the risks associated with each decision. Figure 2.7 shows a proposed business case maturity model.

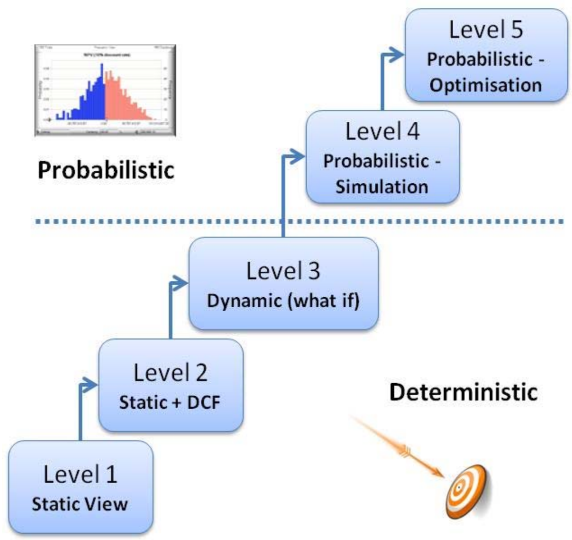

Figure 2.7: Proposed Business Case Maturity Model

Level 1: The business case is developed using point estimation techniques for costs and benefits. The corresponding indicator would be a simple ROI percentage or payback period.

Level 2: The notion of the time value of money is incorporated and the analysis is based on discounted values (DCF). The project benefits can then be measured in terms of NPV and internal rates of return (IRR).

Level 3: The notion of “risk” is (hesitantly) introduced in the model by recognizing the possibility of multiple outcomes of the initiative and developing what-if scenarios (to which multiple financial indicators can be associated through a multi-point estimation exercise). These scenarios, however, remain deterministic in nature and do not provide useful measures of risk.

Level 4: The notions of variability and uncertainty are placed at the core of the decision-making process. Probabilistic modeling and estimation techniques are used to provide a quantification of the risk and therefore the potential cost of making the wrong decision (the cost of ignorance). This cost is then compared to the expected levels of return and are assessed against the organization’s appetite for risk to make a decision in favor or against the change.

Level 5: The notion of scarce resources is recognized as an essential element of decision making, injecting pragmatism in the business case model; different alternatives and combinations are compared in order to define the optimal solution under constraints (“what’s best” analysis).

The industry as a whole seems to struggle to move towards more mature approaches to developing an SPI business case model and the role of the software economist in process improvement efforts becomes ever more relevant. Depending on the level of maturity of current estimation practices, he or she is not only able to quantify the risk levels associated with any decision and the corresponding benefits, but also to express clearly when a decision cannot be made at an acceptable level of risk. This will often be a consequence of excessively fragmented or unreliable information used to make this decision and an unacceptably high potential cost of ignorance.

We believe the majority of software organizations fall somewhere between level 2 and 3 in our business case maturity model and we do not expect to see major changes to this distribution, until the contribution of economists to SPI becomes a generalized practice in the industry.

2.6 The Role of the SPI Economist

The field of software economics is situated at the intersection of information economics and software engineering [Oudrhiri 1999]. Its basic concern is to improve the value created by investments in software engineering. It seeks to better understand relationships between economic objectives, constraints, and conditions on one hand and practices or technologies on the other in order to improve value creation at all levels [Boehm 2000].

As software engineering strives to evolve into a mature engineering discipline, the importance of scientific approaches (demonstrating, through empirical and experimental studies, the causal links between changes and the results they produce) is progressively recognized as a necessary component for successful process improvement [Seaman 2007]. Empirical studies that use quantitative data increase confidence in our decisions help us to measure the impacts of change and allow us to avoid following trends and fashions by imposing a rigorous approach to justifying change. The same thinking is applicable and should be applied to the business case.

The challenges for software process improvement initiatives today are mainly related to the ability to address the deficit in terms of "economic thinking" and develop meaningful economic models for software engineering and SPI and, most importantly, to be able to drive cultural change within the organization by supporting the argument for improvement with a solid business case.

Organizations will be more likely to invest in SPI (especially in the current business climate) if a credible ROI can be demonstrated and supported with reliable data. This data not only provides figures for the expected costs and benefits, but also provides the level of confidence in our estimates and the cost of information, such as the cost-opportunity of investing additional resources in knowledge acquisition. The corresponding process for developing a useful SPI business case would necessarily follow a set of invariant phases.

The software economist uses advanced methods in statistical analysis, mathematics, and programming to asses this information and to develop and apply theories and concepts from economics. The software economist is the "ear" and "eye" of management, which is always more receptive to arguments for process improvement that are built on detailed cost information and a convincing business case rather than generic quality improvement objectives.

A software economist is able to articulate the business case for SPI by using a deep understanding of the software organization and its processes, the cost of poor quality, the impact of defects on development schedules, the effects of latency of software defects across the development life-cycle, the modeling of risks and uncertainty factors, and the estimation of cost and benefits for any improvement project. He or she can elaborate on the value of information and cost of ignorance by introducing new perspectives in the analysis (e.g., real options analysis).

These are not easy tasks. Costs may be hidden and "intangible" in the software development process; software processes may not be directly observable or easily measurable; and these processes deal with a final product (software) that is invisible and can only be communicated through modeling and abstraction.

Costs also need to be understood, modeled, measured, or estimated by studying the system (either process or product) that has generated them and investigated to the level of their root causes. A software economist is able to perform causal analysis and isolate key factors contributing to a particular cost item (be it effort, waste, missed revenues, or revenue leak) and pinpoint areas where significant efficiency gains can be attained.

Economics and experimental methods are also intimately intertwined; they provide (together with improvement best practices) the cornerstones of a mature software engineering and process improvement discipline. When confronted with process issues, a software economist will be able to characterize and solve problems by understanding the relevant economic models, identifying the element of cost, and providing a credible cost benefit analysis on the improvement initiative.

2.7 Conclusions

We believe a real opportunity exists to create a new discipline and area of study for a software process improvement economist in academe, research institutions, and the software industry as whole. It is more than just a job title—this represents the dawn of a new era for software. Software is everywhere and is the key ingredient of our economy. Marc Andreessen, co-founder of Netscape and currently business angel and venture capitalist helping technology start-ups and subject matter experts, in his interview and in the *Wall Street Journal*, said "Software is eating the world" [Andreessen 2011], and declares:

> *"My own theory is that we are in the middle of a dramatic and broad technological and economic shift in which software companies are poised to take over large swathes of the economy."*

In addition, he claims that all the fastest growing companies are software companies and most often we do not think of them as software companies, which is the case when considering Amazon, Pixar, Disney, Netflix, and other companies of this type.

Our experience working with IT and software organizations worldwide confirms the general perception that professional software process improvement economists and their skill sets are in short supply, most notably in small and medium sized organizations. The life span of many software organizations is often less than a few years, mainly and not surprisingly, for reasons of economic underperformance.

The industry as a whole would benefit tremendously from recognizing that the increasing pervasiveness of software and its growing criticality to modern society make it urgent and necessary to move from perceiving software as an art to a real engineering discipline. The introduction of economics thinking to address software process improvement issues will play a key role in facilitating this transition.

2.8 References

URLs are valid as of the publication date of this document.

[Andreessen 2011]
Andreessen, M. *Software Is Eating the World*. http://online.wsj.com/article/SB10001424053111903480904576512250915629460.html (2011).

[Baker 2002]
Baker, D., Bridges, D., Hunter, R., Johnson, G., Krupa, J., Murphy, J. & Sorenson, K. *Guidebook to Decision-Making Methods, WSRC-IM-2002-00002*. http://emi-web.inel.gov/Nissmg/Guidebook_2002.pdf (2002).

[Bateson 1972]
Bateson, G. *Steps to an Ecology of Mind: Collected Essays in Anthropology, Psychiatry, Evolution, and Epistemology*. University Of Chicago Press, 1972.

[Bateson 1979]
Bateson, G. *Mind and Nature: A Necessary Unity (Advances in Systems Theory, Complexity, and the Human Sciences)*. Hampton Press, 1979.

[Boehm 2000]
Boehm, B. & Sullivan, K. *Software Economics: A Roadmap.* ICSE, 2000.

[Howard 1966]
Howard, R.A. "Information Value Theory." *IEEE Transactions on Systems Science and Cybernetics SSC-2* (December 1966): 22-26.

[Oudrhiri 1999]
Oudrhiri, R. *Une approche de l'évolution des systèmes: application aux systèmes d'information.* Vuibert, Dunod, 1999.

[Oudrhiri 2011]
Oudrhiri, R.. & Pellizzetti, F. "Six Sigma: Towards an Empirical and Experimental Approach to Software Engineering." *SEPG Europe*, Dublin, 2011.

[Pellizzetti 2011]
Pellizzetti, F. & Oudrhiri, R. "Probabilistic Approaches to Estimate and Communicate the R.O.I. in Software Process Improvement." *SEPG Europe*, Dublin, 2011.

[Savage 2009]
Savage, S. *The Flaw of Averages: Why We Underestimate Risk in the Face of Uncertainty*. Wiley, 2009.

[Seaman 2007]
Seaman, C. B. "Empirical Paradigm: Position Paper," 23. *Empirical Software Engineering Issues, Critical Assessment and Future Directions, Internal Workshop*. Dagstuhl Castle, Germany, June 2006. Revisited Papers, Springer-Verlag, 2007.

2.9 About the Authors

Radouane Oudrhiri, CTO, Systonomy

Radouane has more than 20 years of teaching and consulting experience in software quality engineering, Lean Six Sigma, and DFSS for Software and maturity-based models. He implemented SPI initiatives within large and small organizations. He is a Six Sigma MBB trained by GE Medical Systems. Radouane is conducting research works on Information Systems Evolution and integration of practices such as: CMMI®[1], SPC, Lean, Six Sigma, Structural Design, TRIZ, Axiomatic Design, and Reliability Engineering. Radouane has an M.S. in Operations Research, an MBA, and a PhD in Information Systems. He is a lecturer at La Sorbonne Paris, Telecom Paris and an evaluator at the European Commission.

Fabrizio Pellizzetti, Senior Consultant, Systonomy

Fabrizio is a Senior Consultant and Six Sigma for Software Master Black Belt with Systonomy Ltd, a London-based practice specialized in Software Quality Engineering. Fabrizio has over 12 years of experience working with a large number of software organizations on the implementation of Lean Six Sigma programs; he regularly delivers training, coaching, and consulting on Lean Six Sigma for Software and ICT.

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

3 Influencing Change Management Decisions in Large Enterprise Transformation Programs

Prasad M. Shastri, Tata Consultancy Services, prasad.shastri@tcs.com

3.1 Abstract

Many large information technology (IT) programs fail to achieve results despite clear vision and good intentions, due to the inadequate management of organizational change. This paper identifies ten dynamic areas of organizational change and demonstrates how their correlation with solution enablers derives a strong qualitative relationship matrix for decision making. Senior executives and program management practitioners learn good examples to apply to their business situations through illustrated success stories of applying this framework in work-streams of enterprise resource planning and enterprise data management, along with use of optimum scale IT off-shoring for cost optimization. Use of quality function deployment and control-impact matrix tools effectively demonstrate the correlation between change barriers and solution enablers.

3.2 Introduction

In this volatile global environment, the world is witnessing a period of economic shift, and customers are cautious in spending their budgets. Business leaders may be keen to invest in large programs to realize huge cost savings, but it may take a long time for these programs to reach the final desired end state of implementation. In many scenarios, the programs fail to reach an end state of implementation or are abruptly shelved for financial reasons. Senior management expects tightly and cost-efficiently managed large transformation programs to accelerate the realization of desired business benefits. All changes have a disruptive potential for the business, and as a result, it is critical to control the release of changes.

This paper gives practical insight into the decision drivers that influence change management and change acceleration decisions in large transformation programs. Large programs are characterized by groups of related projects that are managed using similar techniques in a coordinated fashion. Such large programs with their complex, dynamic, and transformational nature are likely to face numerous stages of organizational change.

3.3 Barriers for Transformation

The causes and effects of change and related evolving dynamics for large programs can be categorized into one or more than one of the following ten segments. Each segment needs special focus and adequate treatment so that it does not hamper the program implementation progress.

1. strong resistance to change, disturbing the “comfort zone” of the affected people (people side)
2. fear of failure at multiple tactical and operational layers of organization structure resulting from implementation of the change (the “trauma” of change)
3. conflicting vested interests of different stakeholder groups (people politics)
4. dynamic nature of program financial budget; periodically changing quarter on quarter (this is frequently observed in the financial industry segment)

5. unplanned attrition of key resources from a business unit or program execution team
6. reorganization of management team members at strategic level
7. subject matter experts (SMEs) from business units having conflicting priorities on their day-job versus their ability and willingness to focus and contribute towards program needs
8. challenges associated with outsourced vendors—their ability to ramp up right skilled resources in a timely manner
9. external factors, like economic, political, and societal shifts
10. management's desire of minimal or no disruption to the business resulting from change

3.4 Important Solution Enablers [Jones 2004, Russell 2010]

No single methodology fits every organization, but there is a set of practices, tools, and techniques that can be adapted to a variety of situations. What follows is a "Top 16" list of guiding principles for change management. By using these as a systematic and comprehensive framework, executives can understand what to expect and how to manage and engage the entire organization in the change process. These 16 guiding principles play an important role in addressing the program situations:

1. Hold proactive program benefit awareness sessions for stakeholders of business units. These sessions involve explaining program objectives and core benefits, how the proposed program will address their pain areas, what involvement and contributions are expected from stakeholders, and answer any other questions that stakeholders might have.
2. Provide the best possible clarity towards return on investment, enabling decisions like "why to spend money."
3. Identify the correct composition of the program team and key risks, and formulate mitigations very early in the program.
4. Perform due diligence: For large programs, applying certain standards of care when examining the situation of an organization in relation to process is an important step to determine the 'as-is' situation of the problem statement. It also determines the gaps for the 'to-be' state.
5. Clarify roles and responsibilities for business and IT stakeholders and program teams, as well as ensure management oversight from the initial stages of the program.
6. Intelligently avoid or engage the correct business units and other groups of stakeholders from relevant organizational hierarchies. This is essential for managing conflicting vested interests of different stakeholder groups.
7. Provide strong resource retention policies, rewards and recognition, and create a harmonious and empowered work culture.
8. Pursue a strategic relationship with outsourcing and off-shoring vendors. A proven past track record and strong global delivery model capabilities will help to address their contributions in delivering value.
9. Obtain thoughtful, visionary, and iterative financial budget planning and buy-in from all strategic stakeholders. The program financial budgets should include adequate contingency planning.
10. Create detailed capacity planning and work breakdown structures on distribution of tasks, and prioritization of duties amongst involved business unit subject matter experts.

11. Reduce dependency on a single vendor and dividing the work logically among multiple vendors. This helps to keep competitive spirit and good control on financial outflow. Coordination between vendors is an important aspect to look at.
12. Provide early visibility into external uncertainties in the economic and socio-political environments.
13. Implement detailed review mechanisms for program deliverables.
14. Control the expectations of different stakeholder groups towards alignment to program objectives.
15. Form a program steering committee as an escalation point to take critical strategic decisions.
16. Deliver bad news in an appropriate manner along with solution options for problem resolution.

3.5 Use of Quality Function Deployment and Control-Impact Matrix

The concept of quality function deployment (QFD) is used to bring out strong correlation between change barriers and the solution enablers. QFD helps to derive a relative weightage for each of the solution enabler with respect to other. The change barriers as customer requirements (the "what" portion) and solution enablers as solution design characteristics (the "how" portion) form the QFD matrix.

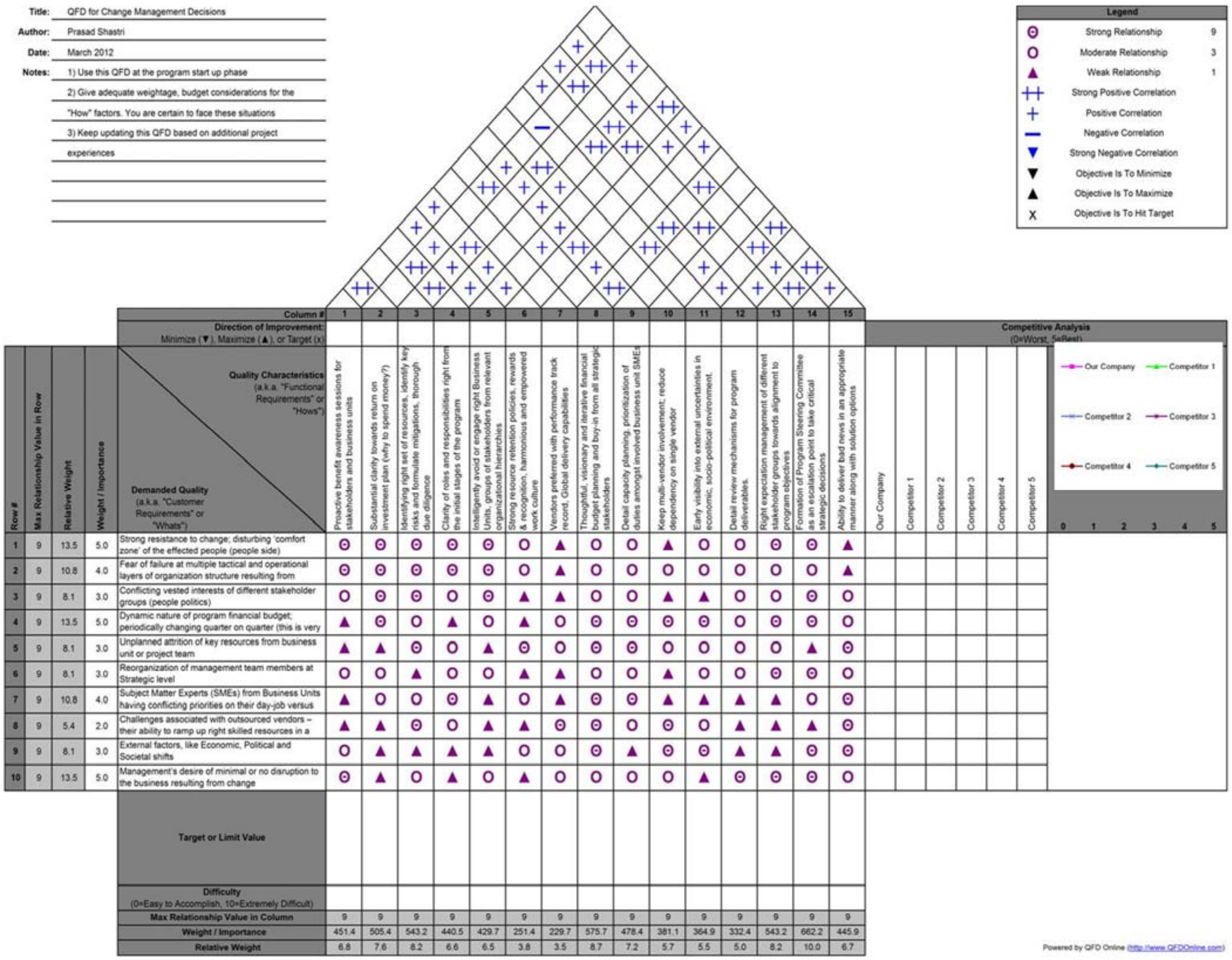

Figure 3.1: Use of QFD [QFD 2007]

3.5.1 Results From the QFD Analysis

Using this QFD tool for the purpose of change management gave positive and interesting results. While all factors are important, the 'relative weight' in this particular matrix helps practitioners to prioritize the critical solution enablers that they have to proactively consider while working on multi-generation program plans. With the correct focus on high-weightage items, the chances of cost and schedule slippage would be substantially reduced.

From the QFD analysis outcome, it is evident that the most critical factors for large programs are the formation of a steering committee, thoughtful and visionary financial budget planning. Closely following are resourcing, risk and mitigation tracking, and controlled expectation management of key stakeholder groups. Many large programs face schedule slippage and cost overrun issues due to inadequate capacity planning. An underestimated capacity for requirements and design reviews and user testing from business users are the primary reasons for deliverables going out of schedule and consequentially out of budget.

3.5.2 Use of the Control-Impact Matrix

The next step in evaluating QFD results is to do an assessment on which factors are within control and which ones are out of control. The control-impact matrix tool [Tata 2012] is used to categorize the solution enablers further into those that are under control and those that are not under control vis-à-vis their ability to create an impact on the key program components (e.g., cost, schedule).

IMPACT →	**High**	**Medium**	**Low**
In Control	• Formation of steering committee • Thoughtful, visionary financial budget planning • Identifying right set of resources, identify key risks and formulate mitigations, thorough due diligence • Right expectation management of different stakeholder groups for alignment to program objectives	• Substantial clarity towards return on investment plan • Detail capacity planning, prioritization of duties amongst involved business unit SMEs • Proactive benefit awareness sessions for stakeholders and business units • Ability to deliver bad news in an appropriate manner along with solution options • Clarity of roles and responsibilities right from the initial stages of the program • Intelligently avoid or engage correct business units, groups of stakeholders	• Multi-vendor involvement • Detail review mechanisms, contingency planning • Strong resource retention policies, rewards & recognition, good project atmosphere, right expectation setting • Vendors preferred with performance track record, global delivery capabilities
Out of Control			• Early visibility into external uncertainties

Figure 3.2: Use of Control Impact Matrix [Tata 2012]

The good news was that majority of the high weightage levers with a high or medium impact are "within control" of the program team. This means the program managers, along with stakeholder groups, can address these situations.

3.6 Case Studies

The author's experience through several programs [Tata 2012] indicate that the program management team has to invest a significant amount of time during the start-up phase to foresee these potential obstacles and lay out a plan for smooth functioning of the program execution. Examples from the following large program disciplines are used to support this hypothesis:

1. Enterprise resource planning (ERP)
2. Enterprise data management (EDM)
3. Use of optimum scale IT off-shoring for cost optimization, in the following case studies

3.6.1 Case Study #1: Enterprise Transformation Through ERP Implementation (Discrete Manufacturing)

Customer profile	The customer is the world's largest designer and manufacturer of turbochargers for the medium-heavy duty diesel engine market and has a reputation for producing innovative and dependable solutions for specific product requirements of this key market sector. Its brand is the best known in the industry, having developed an enviable reputation in pursuit of improved engine efficiencies and emissions reduction in on- and off-highway, marine, and power generation applications worldwide. Headquartered in the UK, its other technical, sales. and manufacturing facilities are located in Brazil, India, UK, USA, and China. Its parent company is the largest independent maker of diesel engines and related products in the world with 40,000 employees in operations in more than 190 countries.
Customer challenges	Meet the growing demand for turbochargers in Europe, the Middle East, and Asia (EMEA) by sourcing components and finished products from production plants in Asia, the Americas, and the United Kingdom. Source items from lower-cost countries to boost profitability. Keep order cycle times for globally sourced items within levels in customer service level agreements (SLAs) to avoid penalties. Gain a real-time view of sales, stock levels, and financial performance at each plant to facilitate timely, informed decision making and compliant statutory reporting.
Program solution	• Replaced the company's disparate, standalone regional order management systems with a single global solution based on Oracle Order Management, Oracle Inventory Management, and Oracle Financials • Integrated the Oracle applications with specialist invoicing tools to build a system capable of automating the fulfillment of customer orders that include multiple part numbers sourced from customer's globally dispersed manufacturing facilities • Used Oracle's inventory management functionality to determine stock levels of parts required at each plant and at the vendor-managed inventory warehouse local to the customer • Leveraged Oracle's order management capabilities to fulfill all parts of each order in the fastest and most cost-effective way • Provided secure system access to a third-party logistics provider that is contracted to handle transport, warehousing, and delivery. • Used Oracle Financials to build a Sarbanes-Oxley and Intrastat-compliant system for statutory reports and value added tax (VAT) declarations • Migrated orders for a major truck manufacturer to the global solution and is now rolling out to all clients in Europe and U.S. • Diversified sourcing strategy from geography centric to global sourcing
Program size	12 person-years (60% IT off-shoring effort)
Author's role in program	Delivery director
Key change management	• Strong resistance to change from manufacturing operational layer • Lack of global data visibility, common measurements, standardized practices and

challenges	processes • Unplanned attrition of key IT resources during important phases of the program • Lack of synergy between business and IT • Specific critical program phases, like chart of accounts finalization and conference room pilots, took substantially longer time than budgeted.
Proactive actions and decisions	• Proactive program benefit awareness sessions for stakeholders of different business departments (manufacturing, sourcing, business finance, quality, inventory/stores) • Right from program startup phase, tremendous focus was given to return on investment, risk management, and mitigations. • Strong emphasis on expectation management of various stakeholder groups • Close collaboration with product vendor and keeping a close vigil on module errors • Regular program steering committee meetings, stringent action tracking • Team bonding events at off-shore locations to keep motivation high

Table 3.1: Enterprise Transformation Through ERP Implementation: Discrete Manufacturing

3.6.2 Case Study #2: Enterprise Transformation Through ERP Implementation (Process Manufacturing)

Customer profile	The customer is a manufacturer of inorganic, organic, and fine and specialty chemicals and is one of the largest producers of sodium nitrite and sodium nitrate with a diverse product portfolio and presence in over 20 countries, including the USA, European Union & East European nations, Japan, ASEAN countries, South Korea, and South America.
Customer challenges	• Improve the productivity of process and personnel • Lower the cost of products and services purchased • Inventory reduction • Reduce lead time • Reduce stock obsolescence • Faster product / service look-up and ordering saving time • Automated ordering and payment, lowering payment processing and paper costs
Program solution	The program involved an enterprise-wide business process re-engineering followed by implementation of 18 modules of Oracle applications (ERP) in a big-bang approach to completely transform and manage the customer's day-to-day business operations and integrate business processes, enabling intelligent data analysis and optimal resource utilization. This complex ERP system implementation involved multiple business modules of financials, distribution, human resources, customer relationship management, process manufacturing, enterprise asset management, business intelligence, and balanced scorecard, all of which were implemented in a record one-year timeframe. A project of this size requires significant commitments of money, time, and human resources from departments across the organization. Apart from addressing most of the customer challenges, the program also gave the following intangible benefits: • increase organizational transparency and responsibility • accurate and faster access to data for timely decisions • reach more vendors, producing more competitive bids • improved customer response • save enormous time and effort in data entry • more controls thereby lowering the risk of miss-utilization of resources • facilitate strategic planning • uniform reporting according to global standards
Program size	20 person-years
Author's role in program	Program director

Key change management challenges	• Strong resistance to change from various departments • Unplanned attrition of key business, IT resources during important phases of the program • Change in the strategic leadership in the middle of the program • Some earlier accepted deliverables by earlier leadership team were questioned for acceptance by the new leadership team, which resulted in loss of time and cost • Very difficult to convince the customer for any change request approval • Challenges on implementation team to establish why a particular functional practice is a best practice in the industry • The implementation team was required to invest long hours.
Proactive actions and decisions	Each functional module involved providing a detailed orientation session on the module for stakeholders with a lot of similar industry examples on how this was implemented in other customer situations. Key risks were monitored and escalated, while mitigations and their effectiveness were evaluated. Close collaboration occurred with product vendor and module errors were closely tracked. Regular program steering committee meetings were held, with stringent action tracking. Detailed review mechanisms were conducted by TCS functional and technical experts.

Table 3.2: Enterprise Transformation Through ERP Implementation: Process Manufacturing

3.6.3 Case Study #3: Enterprise Transformation Through Enterprise Data Management (Investment and Asset Management)

Customer profile	The customer is a leading investment management firm in the USA. It serves more than 2,300 clients in over 35 countries with more than $137.6 billion in assets under management. More than 8,000 investment products are scrutinized annually. The customer has around 1,750 associates in 18 offices worldwide, providing strategic advice and implementation to institutional investors. Creation of state-of-the-art performance benchmarks, in the form of Indexes is their primary expertise. The customer helps individuals prepare for retirement through a range of objectively researched and institutional-quality products.
Customer challenges	• The customer's data management approach had traditionally been highly fragmented in terms of ownership, processes, systems and providers—a complex "spaghetti bowl." • Until 2009, the customer spent an estimated US $55M per year (+/- 10%) in data management (including data providers, feeds, related labor, systems, support). • Large numbers of full-time equivalents (95+) support data cleaning, vendor relationships, and related operational processes in an uncoordinated manner. • Business divisions had taken individual action to improve data handling in specific areas; however, a collective, centralized data management theme was lacking. • Customer was unable to quickly integrate new products and platforms as customer expands and diversifies service offerings. • There was an extended interval between reporting cycles due to time and effort required by business operations to collect required information. • Significant and costly manual data collection and scrubbing was needed to support the business operations. • The fragmented infrastructure affected access to quality, accurate and timely data.
Program solution	A key infrastructure effort, an enterprise data management (EDM) program, was institutionalized to address key business drivers and challenges around accessibility and quality of enterprise data in a cost-effective manner and position customer for strategic revenue expansion. This large multi-year enterprise-wide engagement had the following objectives to attain in phases until the end state: • centralized and ease of access to critical business information • timely and transparent delivery of data to respective business units across the firm • scalability to support expansion in offerings and services • substantially increase speed to market: for introduction of new products/services/reporting • reduction in risk of error in information provided to clients, management, and operations • significant reduction in cost and effort across the enterprise to collect and report information

<table>
<tr>
<td></td>
<td>

single authoritative source for critical business information
automated delivery channel to integrate with applications and enable end user queries
insulate the business unit application layer from future EDM platform changes and enhancements
multi-dimensional analysis enabled via enterprise data warehouse
substantial reduction of manual data collection (possible elimination)
more frequent reporting windows (going from monthly to daily)
formation of data steering group with the following responsibilities:

validate and approve new data to be acquired in partnership with business, IT, and vendor management services
approve sourcing strategy for all new data or additional uses for existing data
all projects implementing solutions that require new data, new access to enterprise data, or changes/additions to enterprise data classes must engage EDM staff to ensure compliance with procedures for data sourcing, naming, keying, security, and others
setting up strong data governance policy to leverage existing strategies, policies, and infrastructure where possible (e.g., vendor oversight, IT security, business continuity)
set up data management infrastructure of standards, processes, and tools that ensure sustainability of EDM services
focus on setting standards for authoritative data sources that will become the foundation of the data factory
address data quality and timeliness standards, inclusive of error logging and metrics reporting
address specific roles and responsibilities

Savings from EDM program
<table>
<tr><th>Year</th><th>Savings (US$ M)</th></tr>
<tr><td>2011</td><td>0.46</td></tr>
<tr><td>2012*</td><td>0.92</td></tr>
<tr><td>2013*</td><td>3.27</td></tr>
<tr><td>2014*</td><td>4.02</td></tr>
<tr><td>2015*</td><td>4.42</td></tr>
</table>
* Projected
</td>
</tr>
<tr>
<td>Program size</td>
<td>150 person-years (70% IT off-shoring effort)</td>
</tr>
<tr>
<td>Author’s role in program</td>
<td>Senior program manager</td>
</tr>
<tr>
<td>Key change management challenges</td>
<td>

This was the first time that a program of this vision and scale was running in the organization.
Significant resource constraints from customer business units
Concurrent run of 16 to 18 projects under aggressive financial budget constraints
Unplanned attrition of key business and IT resources during important phases of the program
Consensus decision making on 80% of the program challenges consumed a significant effort, rather than management hierarchical decision making
Multiple financial budget iterations

</td>
</tr>
<tr>
<td>Proactive actions and decisions</td>
<td>

Workshop sessions with senior management for strategy planning with a high emphasis on strategy planning and execution
Strong program governance structure, monthly steering committee meetings, program status meetings with executive sponsors
Clarity of roles and responsibilities for stakeholders, program team, management oversight from initial stages of the program.
Detailed budget workout iterations and buy-in from all strategic stakeholders. Adequate contingency planning was part of the financial budgeting exercise.
Detail capacity planning, work break-down structure on distribution of tasks, prioritization of duties amongst involved business unit subject matter experts.
IT vendors with global presence engaged for off-shoring work. Vendor agnostic teams

</td>
</tr>
</table>

	were formed for development and testing. There was a strong team spirit between different vendor teams. • Right expectation management of different stakeholder groups • Teams were motivated to bring up innovative ideas. An innovative 'testing factory' approach was institutionalized to address significant resource constraints from business units.

Table 3.3: Enterprise Transformation Through ERP Implementation: Investment and Asset Management

3.7 Conclusions and Recommendations

The previous examples illustrate the following points:

1. Understanding how a proactive, periodic, and thorough assessment and review of change management risks in the start-up phase of a program goes a long way in successfully mitigating these types of risks.
2. The systematic and comprehensive framework developed by the author can be used to understand what to expect, how to manage change, and how to engage the entire organization in this enterprise transformation process for large programs.
3. Understand that complex people issues can also be addressed using scientific tools and they help in the change acceleration process for optimizing the benefits of the change. Tools can greatly help to remove subjectivity.
4. Program management practitioners are encouraged to use the QFD as a type of "jump start" material when articulating their risk mitigation strategies for large, complex programs. This QFD can be augmented with specific program situations and can also be generally used for various industry verticals and segments.
5. This paper is particularly useful for the project managers of smaller engagements who are aiming at managing larger programs in their careers.

3.8 References

URLs are valid as of the publication date of this document.

[Jones 2004]

Jones, John, Aguirre, DeAnne, & Calderone, Matthew. *10 Principles of Change Management*. http://www.strategy-business.com/article/rr00006?gko=643d0 (2004).

[QFD 2007]

QFD Online 2007. *Free QFD Templates, version 2.0.346.0.* http://www.qfdonline.com/templates/ (2007)

[Russell 2010]

Russell, George F., Jr. *Success by Ten*. John Wiley & Sons, 2010.

[Tata 2012]

Tata Consultancy Services. *Internal Reference Material*. 2012.

3.9 About the Author

Prasad Shastri is a Senior Consultant at Tata Consultancy Services (TCS) presently based in Seattle, US. He has more than 16 years of industry experience and specializes in process improvements, large program management, with core competency in enterprise transformation engagements and managing off-shore delivery centers. He has worked with several Fortune 50 Global customers in North America, Europe, and Asia-Pacific regions in large enterprise transformation programs to help them realize benefits of implementing these programs.

Prior to Joining TCS, Prasad was an Executive with VIP Industries, the world's third largest manufacturer of molded and soft luggage, managing process planning and process automation. During his four years with VIP, he led initiatives of design, development, and implementation of robotics' solutions for injection molding machines and conveyorized assembly systems for luggage assembly.

Prasad holds an MBA in Finance and Marketing from Nagpur University, India. Prior to receiving his MBA, he finished his Post Graduate Diploma in Marketing and Sales Management (Gold Medalist) and Bachelor of Engineering in Mechanical Engineering from Nagpur University, India.

4 Using CMMI for Software Improvement in Small Organizations: A Case Study

Dr. Joaquín Lasheras, CENTIC (Technological Centre of ICT), Spain, joaquin.lasheras@centic.es

Dr. Javier Garzás, Kybele Research and Kybele Consulting, Spain, javier.garzas@urjc.es

José María García, SQA (Software Quality Assurance), Spain, jmgarcia@sqa.es

4.1 Abstract

Small and medium software organizations (SMEs) are very important to the economic growth of many Spanish regions. As a consequence, improving their competitiveness in the information and communication technology (ICT) sector must be considered both an aim and a challenge. In this paper we will discuss the experience of a particular pioneering project in the region of Murcia, called IMPULSE CMMI, which has implemented CMMI in several companies within the region. We will present the lessons learned and the benefits obtained using a protocol template for case studies, including the efforts needed regarding time and cost, of the SME. We will also explore the following question: Is CMMI useful and practical for carrying out software process improvement efforts within small software organizations in regions such as Murcia, which is considered a disadvantaged region of the European Union?

4.2 Learner Outcomes

- Listen to a case study of evaluation of CMMI level 2 in version 1.2
- Hear lessons learned from the implementation of CMMI in SME
- Discuss whether CMMI is useful and practical for carrying out software process improvement efforts in SMEs inside a region considered to be disadvantaged by the EU

4.3 Introduction

Small and medium software organizations are important to the economic growth of many Spanish regions and, as a consequence, their improvement in competitiveness in the ICT sector must be considered a goal and a challenge. On the other hand, important models from organizations like the Software Engineering Institute (SEI) or the International Organization for Standardization (ISO) are widely used in Spain to ensure that small software organizations increase their productivity and quality. In particular, Spain has the fourth highest number of Capability Maturity Model Integration—Development (CMMI-DEV) appraisals carried out per country, making CMMI the most extended and representative model for the improvement of software processes.

Therefore, in this paper we will present the experience of a pioneering project in the region of Murcia, Spain called *IMPULSE CMMI*, which implements CMMI in several companies in the region. This project was financed by the Autonomous Community of Murcia through the Institute for Industrial Development of Murcia and the Technological Centre of Information Technologies and Communications of Murcia (CENTIC), a non-profit private institution created to contribute to the excellence and sustainable development of its members by means of cooperation, rendering technological services, and promoting values related to constant innovation. Currently CENTIC has about 40 members, including the most important ICT companies of the Murcia region. The project has also involved the enterprise organization

Kybele Consulting, a spin-off of Rey Juan Carlos University, in a consulting role, along with the support of CENTIC and the four SMEs evaluated. Kybele Consulting employs more than 100 professionals from Murcia, Spain. The project focused on improving the management and development processes of these companies under the CMMI-DEV quality model at maturity level 2, with the goal of increasing the competitiveness of the region through the improvement and certification of its software companies. This implied the adoption of a complete model for improving the processes of software development, adapting it to the characteristics of the Murcia software industry. Because of its interest to the region, the project has been supported by the European Regional Development Fund (ERDF).

In this paper, we will present a detailed case study that addresses the process improvement in the company SQA, one of the SMEs involved in the *IMPULSE CMMI* project. SQA is a company that focuses on the development of tailored software for state-owned companies and state entities. It specializes in large corporate websites with standard attributes of accessibility. With an average workforce of 25 employees and a business volume of about one million euros, its main aim is the achievement of efficiency of the enterprise by applying software engineering techniques and good quality assurance practices.

A case study explores a phenomenon within its real context [Yin 2008]. Runeson and Host [Runeson 2009] identified the case study as often being a good method for research in software engineering. We conducted this case study following the protocol template for case study planning given by Brereton et al. [Brereton 2008]. The following subsections describe the case study.

The paper has been structured in five main sections: first an introduction about the case study is presented, and section 4.4 shows other related work. Section 4.5 describes the case study, identifying the research design, the subject and analysis unit, the field procedure and data collection, viability plan, limitations and, finally, the results generated. The lessons learned are discussed in Section 4.12, and conclusions and further work are presented in Section 4.14.

4.4 Related Work

The software process community has long expressed a special interest in software process improvement. In this sense, there are many papers that deal with this topic, as can be seen in the study of the trends in publications presented in Hall, Rainer, and Baddoo [Hall 2002].

Moreover, for some time now, there have been several initiatives specifically oriented at studying processes on real environments. Studies such as Pino, Pardo, García, and Piattini [Pino 2010]; Saiedian and Carr [Saiedian 1997]; Johnson and Brodman [Johnson 1997]; Hareton and Terence [Hareton 2001]; Staples et al. [Staples 2007]; and Brodman and Johnson [Brodman 1994] show how process models and CMMI are applied in companies. Furthermore, there have been systematic reviews specifically oriented at studying processes, such as those presented in Pino, García, & Piattini [Pino 2008].

However none of the above deal specifically with process improvement in small and medium enterprises in order to improve the competitiveness of businesses present in small regions.

4.5 Case Study

We considered it to be fundamental to have a case study protocol from the outset in order to define and record matters like design, case selection, case study procedures and roles, data collection, collecting evidence, analysis, plan validity, and study limitations data, among other factors. Studies such as those described in Brereton et al. [Brereton 2008], Yin [Yin 2008], Host and Runeson [Host 2007], and Runeson and Host [Runeson 2009] should be taken into account in the creation of the protocol, which should be reviewed by other researchers with more experience in the empirical research field, initially to validate it and subsequently to track the case study execution in order to ensure that it was performed properly. In any event, the checklist for case studies presented in Host and Runeson [Host 2007] is a suitable guide for determining whether all the elements that must be taken into account when performing a case study have been considered. At the end of each of the main activities involved (design, preparation for data collection, collecting evidence, analysis of collected data and reporting), each group of questions on the checklist was checked. In addition, feedback on the data collected from those involved in a case study could be useful for reviewing and confirming findings. This lets us validate and improve each activity planned in the protocol, thereby guaranteeing the rigor of the case study and the greater reliability of the results obtained.

4.6 Research Design

This section describes the research goals, while the approach of this study and case study contexts will also be introduced shortly.

The main research question addressed by this study is: Is CMMI useful and practical for carrying out software process improvement efforts in small software enterprises in regions like Murcia? In this context, “useful and practical” means that CMMI is helpful and can be used by the companies to achieve a competitive advantage. Additional research questions addressed by this case study are as follows:

- Is the effort involved suitable for small companies in Spanish regions? (A “suitable” effort is defined as one that a company can undertake by following the proposed methodology.)
- Are the benefits (internal and external) enough for this effort? (internal benefits might include improvement in the development process, while external benefits might include those that enhance the company’s reputation)

By asking these questions, we sought to discover whether CMMI has a useful function and a practical use, while respecting the reality of small companies, that is, is it suitable for them? Moreover, the object of this study is to validate a methodology for improving software processes in small software organizations in disadvantaged regions, such as Murcia. Furthermore, we must adapt CMMI to the particular characteristics of the Murcia software industry, which until now has been focusing on tailored software, mainly to state-owned companies and state entities, but now, in a time of crisis that have affected mainly these state-owned companies and state entities, the industry must evolve to improve the development of software factories and near shore capabilities, showing off-shore benefits with competitive prices.

We conducted this case study following the guidelines set out in Kitchenham, Pickard, and Pfleeger [Kitchenham 1995], Runeson and Host [Runeson 2009], Yin [Yin 2008], and Host and Runeson [Host 2007]. In addition, this case study, based on the approach presented by Yin [Yin 2008], has been designed according to guidelines for case studies presented by Brereton et al. [Brereton 2008], where there are eight main activities: case study design, case selection, case study procedures and roles, data collection, analysis, plan validity, study limitations, and reporting.

The measures used in the research were, first, the effort used in carrying out the tasks associated with CMMI. We measured the total effort, the effort for each CMMI process phase (improvement phases), and the final evaluation effort (readiness review and final evaluation—appraisal phase). We also took into account the benefits described by the company involved, both internal benefits (process improvement) and external benefits (client opinions).

4.7 Subject and Analysis Unit (Case Selection)

SQA is the organization presented in this case study. SQA is part of the group of companies involved in the "*IMPULSE CMMI project*," a pioneering project in the region of Murcia, Spain. The project has involved Kybele Consulting (a spin-off of Rey Juan Carlos University) as a consultant, with the support of CENTIC as well as four companies of the region, which employ more than 100 professionals from Murcia in Spain.

The project focused on improving the management and development processes of these companies under the CMMI quality model at maturity level 2 with the aim of increasing the competitiveness of the region through the improvement and certification of their software companies. This implied the adoption of a complete model for improving the processes of software development, adapting it to the particular characteristics of the Murcia software industry, which is considered a disadvantaged region by the European Union (EU). Consequently, the subject and analysis unit in question will be to start a cycle of improvement with the support of CENTIC and Kybele Consulting in SQA (a small company in Murcia, Spain). Table 4.1 describes the profile of SQA.

Organization	**SQA**
Country and Region	Murcia (Spain)
Employees	25
Market	state-owned companies and state entities
Professional activity	tailored software
Specialization	large corporate websites with standard attributes of accessibility
Business volume	900,000 euros
Business objective	achievement of efficiency of the enterprise by applying software engineering techniques and good quality assurance practices

Table 4.1: Profile of SQA

The project scope of SQA has allowed them to achieve success in all their projects related to tailored software. SQA considers project tailored software for those clients that have specific requirements, a deadline, and a specific budget. These projects are characterized by a strong management component by the customer's technician and the application of its norms, methodologies, standards, and tools. However, customers usually have a number of listed requirements to meet goals, which must be considered when implementing improvement efforts.

4.8 Field Procedures and Data Collection

In the following sections, we will present an overview of the work carried out on the improvement and diagnoses of the processes in SQA that follows the proposed methodology below.

4.8.1 Software Process Improvement Projects

Before the final diagnoses of the processes in the organization, a preliminary project of process improvement was necessary, consisting of initiating the cycle of improvement through the following phases.

Improvement phase:
1. Initial assessment (SCAMPI-C): previous diagnosis and improvement plans definition
2. Goal 1 assessment: Quality management system definition
3. Goal 2 assessment: Process implementation in pilot projects
4. Final SCAMPI-B appraisal
Appraisal phase:
5. Readiness review
6. Final SCAMPI-A appraisal

Table 4.2: Phases in the Software Process Improvement Project, Including the Appraisal Phase

For each company, the preliminary improvement plan was defined, and the number of iterations making up the improvement cycle presented, together with the order of their execution and the overall schedule. Proactive administration of the major risks involved in the improvement cycle was established and the corresponding management strategy was registered. Training was planned for those involved and it was established that basic process measures would be performed on two things: first, the processes to be improved in an organization (based on the extent to which the organization achieved its capability level), and second, the improvement process to be used (by measuring the effort made in carrying out this process). The person responsible for process improvement created the general improvement plan, along with the assessment report and the preliminary improvement plan.

In the improvement of processes, the first instruction was provided by the Kybele Consulting adviser, who worked in close relationship with the person responsible for the process improvement in each company. The person not only received training in process assessment, but also gained the experience needed to do their job properly. Process diagnosis was done on an ongoing basis in the organization, supported by its own staff and with no need for an external adviser. The process followed these steps:

- Launch the cycle of improvement and collect information about the companies involved.
- Socialize software process improvement in order to disseminate the software improvement initiatives, whereby CENTIC shared the work to be done with other companies and received feedback and expressed expectations.

4.8.2 Assessment Execution

The SCAMPI assessment process was used to drive the activity of diagnosing the processes in each organization. This can be broken down in the phases that appear in Table 4.2.

In the initial assessment (SCAMPI-C) a preliminary diagnosis for the company with respect to CMMI level 2 was developed. Consequently, an improvement plan was defined in order to put this into practice in the organization, which is used in the next phase, namely, the quality management system definition (goal 1 assessment). Finally we had to prove that the improvement plan had been implanted through a pilot project first of all (process implantation in pilot projects, goal 2 assessment) and finally in the entire organization (SCAMPI-B).

During each of these phases, the consultancy company visited the evaluated company in order to carry out the different assessments of the chosen processes, for which evidence-gathering techniques (interviews and surveys) were performed using specially created information-collecting instruments. For example, the extraction of evidence was performance with practice implementation indicators database (PIIDB) adapted by the consultancy company.

Finally, the appraisal phase corresponded to the phases for a formal appraisal of CMMI: the readiness review where the PIIDBs approved for the SEI are prepared and, finally, with the final appraisal (SCAMPI-A), where a SCAMPI Lead Appraiser evaluated the company during a week through the use of interviews with the stakeholders.

4.9 Plan Validity

To address the threats to the validity of the case studies, the following issues were considered:

- The design of the case study and the data collection plan were checked against the checklists for case studies on software engineering proposed in Host and Runeson [Host 2007], with a high percentage of positive results.
- Regarding the construct validity, we collected data from participants with different roles and from multiple sources, including document archives, surveys, interviews, and participant observation.
- Furthermore, the use of templates related to each activity of the field procedure allowed us to maintain a chain of evidence with traceability between research questions, recorded data, evidence, and analysis.

4.10 Limitations

The case studies set out in this paper have certain limits.

- The observations and conclusions presented are based on only one case study, which could limit the power of generalization, although the company is representative of the software industry in Murcia, Spain.
- Because of the current global economic crisis, particularly in Murcia with its state-owned companies and state entities, new related projects have been restricted, so some qualitative data collected cannot be compared to data from other related projects.
- The bias of the case studies, as employees' performance of daily activities, may be affected by being observed. Bias could also arise from the particular kind of handling of events and data by the advisers.

4.11 Results Generation: Analysis and Reporting

The information generated from the data achieved its aim to provide a view of the state of the organization's processes.

4.11.1 Strengths

After process improvement, the strengths of SQA are mainly as follows:

- processes clearly described in the company's documentation
- staff stability in software development
- reduced maintenance costs
- improved on time delivery of software projects

- increased productivity
- decreased software bugs
- increased customer satisfaction (mainly with state-owned companies and state entities)

4.11.2 Process Maturity

The information on the maturity of the processes was obtained by the application of an extended version of SCAMPI C using an Excel worksheet (using the PIIDB as the baseline). Values thus assigned were “never” if there was no evidence of practice implementation or existence of a product, “always” if there was direct evidence, and “sometimes” if there was indirect evidence or comments.

The evaluator analyzed the information collected during the process assessment of the SQA organization in order to identify potential improvement opportunities.

4.11.3 Process Prioritization

SQA decided to improve the process in this order (according to the process areas in CMMI maturity level 2):

1. Configuration Management (CM)
2. Project Monitoring and Control (PMC) and Project Planning (PP)
3. Requirements Management (REQM)
4. Measurement and Analysis (MA)
5. Process and Product Quality Assurance (PPQA)

SQA decided to proceed in this order because they expected to have a centralized and flexible work environment which would allow them to manage the lifecycle of any project and ensure the compliance of the procedures defined for each of these CMMI process areas. They wanted to have them integrated into their working tool called SQAdra (ADotprojectTool). For this, the process was guided for new modules added to SQAdra, which gives support to some specific area of CMMI, including this training (see Section 4.11.5 below). The area of Supplier Agreement Management (SAM) was not tackled since SQA does not subcontract any software development process in projects.

4.11.4 Efforts

In this section, we will present the efforts per person/hours, in accordance with the implied effort per person to carry out the event or phases described in Table 4.2. Table 4.3, Table 4.4, and Table 4.5 show the effort in terms of hours per person, grouped by consultants and developers, and the date when the phases were performed. The developer’s effort is specifically related to improvements to SQAdra and for the training sessions for institutionalizing processes and procedures, while the consultant´s effort is related to analysis and design.

Table 4.3: Improvement Phases Effort in Hours per Person

	Effort (Hours)	Profile Hours per Development	Profile Hours per Consultant
IMPROVEMENT PHASES			
1 (July 2009)	19	0	19
2 (December 2009)	311	11	300
3 (April 2010)	1374	748	626
4 (December 2010)	750	156	594
Total	**2454**	**915**	**1539**

Table 4.4: Appraisal Phase Effort in Hours per Person

	Effort (Hours)	Profile Hours per Development	Profile Hours per Consultant
APPRAISAL PHASE			
5 (April 2011)	337	0	337
6 (May 2011)	250	44	206
Total	**587**	**44**	**543**

Table 4.5: Total Effort in Hours per Person

	Effort (Hours)	Profile Hours per Development	Profile Hours per Consultant
IMPROVEMENT	2454	915	1539
APPRAISAL	587	44	543
Total	**3041**	**959**	**2082**

4.11.5 Tools Infrastructure

SQA decided to implement a continuous semi-integration system where the tools to give support to the software configuration system, in order to achieve the improvement goals, are:

- Configuration elements repository: Subversion
  - The free software "VisualSVN Server" was decided on as a base.
  - The open-source software "Sventon" was used as the web browser of the SVN repositories.
- Project and task management and control: SQAdra (a proprietary tool)
  - The open-source software "DotProject" is used as a base, adapting to the necessities of the company, in order to
    a. support project management and monitoring
    b. enable artifact verification and validation (requirements, meeting reports, training actions, task implementation, etc.)
    c. enable a dashboard of indicators of Measurement and Analysis (MA)

d. enable bidirectional traceability between requirements by using cases/tasks and source code

e. enable non-conformance management and monitoring

- Management and Traceability of tasks and repository: Redmine
  - The open-source software "Redmine" was used to enable the direct traceability of tasks with Apache Subversioning (SVN) repositories.
  - The Module "news" is used as a blog where the comments about the evolution of the project were collected to make a log of the project.
- Change traceability (repository to task)
  - Implantation of TortoiseSVN through the integration of Subversion-Redmine, and finally Redmine-DotProject

These tools were integrated in the main tool SQAdra as modules, for example: SQATest (verification, validation and agreements), SQATrace (traceability query) or SQAhistory (indicator query).

4.12 Lessons Learned

In this section, we will highlight the most relevant aspects of the application of the CMMI and IMPULSE project in SQA. First, we will examine the lessons learned of the application of the methodology and then examine the lessons learned by the company's stakeholders, its problems, and suggested improvements.

4.12.1 Experience of the Case Study

The research method was the case study method carried out in a systematic manner, which led to increasingly more reliable results. It is a well-defined structure for application in small enterprises. In this section, we will describe the most interesting lesson learned from the case study.

4.12.1.1 Effort

In this section, we will try to evaluate if the effort of applying CMMI was reasonable or not for the characteristics of the organizations involved in the improvement. We must also consider if the employees were able to take on this effort without or with any negative effect on their normal activities.

Little information is currently available in the literature on the effort associated with conducting a process improvement in small companies. Based on the research by Kelly and Culleton [Kelly 1999], Humphrey, Snyder, and Willis [Humphrey 1991], and F. J. Pino et al. [Pino 2010], we set out the various phases in the software process improvement project in Table 4.2, we can see that the effort has been exceeded in this company, because of their full involvement in the improvement process. What is more, we have to highlight the greater effort made during the improvement phase (see Table 4.3) than for the appraisal phase (see Table 4.4As a result, it proves that the company had been properly prepared, regardless of the fact that the effort in these phases had been expensive, but it assured them a good evaluation in the appraisal phase.

The effort made in improving the software development has been achieved by the internal and external benefits, which are described in the following sections.

4.12.2 Internal Benefits

Both the management and the employees of the organizations saw the benefits of the results and, more importantly, they realized the need for ongoing assessment and improvement that follows this same approach for future cycles. Process improvement that is based on the results of the assessment carried out allows firms to move from a chaotic and unpredictable software process to a tangible one, which is currently being used on development projects.

The companies now keep a register of the work products related to the processes improved, together with the instancing in the projects applied. This has allowed them to begin to generate a knowledge base that makes historical data available when decisions are made.

The project visibility has been clearly improved in the sense that all the stakeholders are totally integrated in the project, including the clients. Other than that, the change and configuration management has been improved through the definition of baselines and metrics in the following specific ways.

- Requirements management has greatly improved thanks to the agreements reached with clients, which are now based on formal documents and not solely on ideas understood by the analysts. Misunderstandings in the features expected from the application do not now exist. Formal validation of requirements based on the IEEE Standard and the use of prototypes by the client by validation tests in SQAdra permits the company to detect problems and difficulties in early phases, instead of having them appear in later ones. In addition, the side effects that may exist in changing requirements and ongoing maintenance of their applications are avoided by the use of traceability controls.
- Planning and project control, together with measurement and analysis, have increased the visibility of projects for the management of the company and for the project manager, enabling advances in decision-making and managing unexpected situations more successfully (risk management). Previously, the state and performance of the projects were unknown objectively until their closure, but now program managers have to renegotiate contract conditions and minimize the impact of some incidents that, unchecked, could have resulted in losses for the company. For example, the automatic measurement (deadlines, predicted final, percentage of cost effectiveness or last monitoring meeting with the client) give support to the decision taken in the weekly planning meeting, together with the assignment of resources.
- Configuration management, although it was practiced in the company, was upgraded to provide a robust system that has increased the reliability of work and control over products, avoiding problems and allowing the company to know in-depth the state of products and their differences.

4.12.3 External Benefits

The main benefits with respect to the client and the working environment that can be emphasized at the end of the improvement project are that this project has allowed them to be more competitive in the market and take on expansion challenges. This improvement effort represents an improvement in the company image. In addressing the implementation, SQA has demonstrated its ability and interest in the improvement of its processes. Their customers are very satisfied with the dependability and quality of their work, and the technology community in our region welcomes their achievements and ability to demonstrate the maturity of their development methodology.

SQA has collected reviews from customers that reflect their compliance with these new techniques. Members of the company have also been invited to give conferences about their new solution and some possibilities of work have been achieved from these opportunities.

4.13 Experience of the Company's Stakeholders Involved

The main problems with this implementation were related to the cultural changes that represented the institutionalization of formal processes in the company and for some customers who were used to the previous informal situation.

In the area of requirements management, most of the project managers resisted the change. The formalization of requirements, validation, agreement of the team on the requirements specification, and subsequent final validation of the customer about the product were the most difficult processes to implement and maintain.

The areas of planning and project management processes were based on the previously existing processes, so that adaptation was easier. In these areas, it is important to note the difficulty of keeping an objective, up-to-date, and continuous monitoring of indicators of the projects. The operational tasks often prevented the development of planning and monitoring tasks by project managers, but in the end we could correct it when the ability to maintain control and visibility in the projects was important.

The areas of configuration management and measurement and analysis were designed from the beginning to be automated and were the first to be successfully institutionalized in their projects, almost seamlessly. In this sense, the implementation of the process and product quality assurance area was not a great effort either, since these implementations mainly affect the quality of the SQA group and not the entire staff.

On the other hand, the existence of formal and detailed process implies an essential guide and contributes to the best performance of software professionals, who know what to do and can drive their partners in order to get the best performance in the development of software project and its closure. This was achieved because of the increase in organizational capacity and collaboration, thanks to the sharing of ideas during each of the phases of the improvement process described in this paper.

4.14 Conclusions and Further Work

On the basis of the analysis described in this paper, we consider that the evidence from the data collected at the end of the improvement shows that the IMPULSE CMMI methodology generated reliable information, which was used to formulate and execute process improvement in small organizations with the aim of increasing the capability of their processes. The method used to evaluate this has been useful, describing the data to collect, the phases, and the iteration of the project phases.

We have described this as being a difficult task, implying the need for an important effort made in the company, but thanks to the complete involvement of all the stakeholders, we have obtained enough benefits. Besides, it is important to distinguish between the effort involved in the improvement process and the effort involved in the appraisal process, prioritizing the former.

On the other hand, the company, represented by its chief executive officer emphasized that the company has a more specific vision of itself, which has helped and motivated the company to set out on the road to quality certification. Reflecting on their processes and the sharing of ideas for improvement has led to an increase in the company's organizational capacity and collaboration. Moreover, it is worth noting, that the customer involvement represents a success factor, and that the focus on quality has gone on to become an ongoing objective and now guides every step of the development process.

In any case, it has been shown that its profits are better and now the company is prepared to confront the current crisis, because it now has a strong and privileged position and a quality company image, so it can be more competitive in the market and take on expansion challenges. To answer the original question (Is CMMI useful and practical for carrying out software process improvement efforts in small software enterprises at regions like Murcia?), we have to answer yes, but we should take into consideration that the main objective of the company must be focused on preparing the development of software factories and improving near-shore capabilities, showing off-shore benefits, with competitive prices, and avoiding tailored software for state-owned companies and state entities.

Finally, another important item to note is that the work does not finish with the appraisal. Moreover, the process of measurement and analysis must continue (indeed improving even more), if they are considering reaching other maturity levels in CMMI, such as level 3 or higher.

4.15 Acknowledgments

We acknowledge the assistance of the following projects in carrying out our case study: the IMPULSE CMMI 2009–2010 program financed by the Regional Ministry of Industrial Development, which ran for 16 months and involved CENTIC, Aquiline Computer, SQA Murcia, Grupo SIS, and Visualtis, as well as the MASAI project of Rey Juan Carlos University (TIN-2011-22617), which was financed by the Spanish Ministry of Education and Science.

4.16 References

URLs are valid as of the publication date of this document.

[Brereton 2008]
Brereton, P., Kitchenham, B., Budgen, D., & Li, Z. "Using a Protocol Template for Case Study Planning," 1–8. *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. University of Bari, Italy, 2008. BCS, 2008.

[Brodman 1994]
Brodman, J. G. & Johnson, D. L. "What Small Businesses and Small Organizations Say About the CMM," 331. *Proceedings of the 16th International Conference on Software Engineering (ICSE-16).* Sorrento, Italy, May 1994. IEEE Conference Publications, 1994.

[Hall 2002]
Hall, T., Rainer, A., & Baddoo, N. "Implementing Software Process Improvement: An Empirical Study." *Software Process: Improvement and Practice, 7,* 1 (March 2002): 3–15.

[Hareton 2001]
Hareton, L. & Terence, Y. "A Process Framework for Small Projects." *Software Process Improvement and Practice 6*, 2 (June 2001): 67–83.

[Host 2007]
Host, M. & Runeson, P. "Checklists for Software Engineering Case Study Research," 479–481. *First International Symposium on Empirical Software Engineering and Measurement*. Madrid, Spain, September 2007. IEEE Conference Publications, 2007.

[Humphrey 1991]
Humphrey, W. S., Snyder, T. R., & Willis, R. R. (1991). "Software Process Improvement at Hughes Aircraft." *IEEE Software 8*, 4 (July 1991): 11-23.

[Johnson 1997]
Johnson, D. L. & Brodman, J. G. *Tailoring the CMM for Small Businesses, Small Organizations, and Small Projects*. Edited by K. E. Emam, & N. H. Madhavji. Computer Society Press, 1997.

[Kelly 1999]
Kelly, D. P. & Culleton, B. "Process Improvement for Small Organizations." *Computer 32*, 10 (October 1999): 41–47.

[Kitchenham 1995]
Kitchenham, B., Pickard, L., & Pfleeger, S. L. "Case Studies for Method and Tool Evaluation." *IEEE Software, 12*, 4 (July 1995): 52–62.

[Pino 2008]
Pino, F., García, F., & Piattini, M. "Software Process Improvement in Small and Medium Software Enterprises: A Systematic Review." *Software Quality Journal 16*, 2 (June 2008): 237–261.

[Pino 2010]
Pino, F. J., Pardo, C., García, F., & Piattini, M. "Assessment Methodology for Software Process Improvement in Small Organizations." *Information and Software Technology 52*, 10 (October 2010): 1044–1061.

[Runeson 2009]
Runeson, P., & Host, M. "Guidelines for Conducting and Reporting Case Study Research in Software Engineering." *Empirical Software Engineering 14*, 2 (April 2009): 131–164.

[Saiedian 1997]
Saiedian, H. & Carr, N. "Characterizing a Software Process Maturity Model for Small Organizations." *ACM SIGICE Bulletin 23*, 1 (July 1997): 2–11.

[Staples 2007]
Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P., & Murphy, R. "An exploratory study of why organizations do not adopt CMMI." *Journal of Systems and Software 80*, 6 (June 2007): 883–895.

[Yin 2008]
Yin, R. K. *Case Study Research: Design and Methods (Applied Social Research Methods).* Sage Publications, 2008.

5 Analysis of Coverage of CMMI Practices in Software Engineering Curricula

Ana M. Moreno, School of Computing, Universidad Politécnica de Madrid, Spain

Maria-Isabel Sanchez-Segura and Fuensanta Medina-Dominguez, School of Computing, Universidad Politécnica de Madrid, Spain

5.1 Introduction

Educating software practitioners to confront current and future challenges of the software industry is a problem that we have been grappling with more or less ever since the outset of software engineering. Lethbridge et al. [Lethbridge 2007] suggest that filling this gap is one of the most critical tasks to be addressed in software engineering education. They argue that the task is complicated by several open questions, such as: What industrial practices are currently not being taught? How effective are these practices? Which practices should be taught to undergraduates?

Several studies have addressed this topic and have tried to identify differences between key knowledge for a software engineer from the viewpoint of industry and academia. One of the first surveys was conducted by Lethbridge et al. [Lethbridge 2000], where a representative set of U.S. and Canadian software professionals valued the relevance and depth of the knowledge received as part of their graduate education. Those professionals highlighted a significant mismatch between software education and their actual industry practice. Other similar studies, like the surveys by Kitchenham et al. [Kitchenham 2005] or Surakka [Surakka 2007], again identified such a fissure.

The authors published another study that discusses this gap between software practice and software education [Moreno 2012] comparing knowledge provided by SE2004 [IEEE 2004] and GSwE2009 [Graduate Software Engineering 2009] to cover software development-related professional profiles listed in the Career Space report [Career Space 2001].

Koong et al. [Koong 2002], Prabhakar et al. [Prabhakar 2005], or Kovacs and Davis [Kovacs 2008] undertook other initiatives that tried to identify relevant skills for the software industry, studying the keywords in online job postings as indicators of the critical skill requirements of information technology (IT) professionals. The Gartner Group [Morello 2005] also provided some insights to this debate and agreed with Davey [Davey 2008] about the need to complement technical software engineering education with other business-oriented knowledge.

As this is an important issue for both industry and academia, we provide a complementary analysis adopting CMMI and specifically CMMI-DEV as a source of software practices towards which the software industry is now moving. The choice of CMMI as a baseline for the knowledge required by the software industry is not frivolous. It is well known that the benefits of CMMI lead to it being used by large and small enterprises across different domains, like banking, health, insurance, government, and others. Users include Boeing, General Motors, JP Morgan, Bosch, and many others. CMMI certification has come to be synonymous with a competitive edge for an organization, giving it better market options. CMMI DEV was chosen, among the different CMMI models, as it covers the best practices that address development activities applied to products and services, during all development lifecycle. Such development activities perfectly cover the software development activities to be performed by a software engineering organization.

In this context, our goal is to analyze how well qualified graduates of programs based on the international software engineering education standards, SE2004 and GSw2009, are for implementing practices covered by the different CMMI-DEV process areas as thoroughly as required by this model.

The results of this study can be useful for different purposes. The software industry can use them to identify possible skill gaps among graduates; these are specific gaps to which training can be targeted depending on either the organization's current or targeted maturity level. From the academic viewpoint, the results of this study can be used to improve software engineering programs by filling the identified gaps in either the program cores or electives. Finally, this information can help graduates that intend to join CMMI-appraised organizations and actively contribute to their improvement in conformity with this model to identify gaps in their training.

This paper is structured as follows. Sections 5.2 and 5.3 briefly describe the SE2004 and GSwE2009 standards. Section 5.4 contains an overview of CMMI-DEV. Section 5.5 introduces the analysis of the core contents recommended by SE2004 and GSwE2009 in terms of their adequacy for addressing different CMMI-DEV process areas. Finally, Section 5.6 contains a discussion and conclusions with the main findings derived from this study.

5.2 Software Engineering Undergraduate Curriculum Guidelines

The 2004 Software Engineering Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering were instituted with the aim of "providing guidance to academic institutions and accreditation agencies about what should constitute an undergraduate software engineering education" [IEEE 2004]. One of the outcomes of SE2004 was what was termed software engineering education knowledge (SEEK). SEEK represents a body of core knowledge or essential material that professionals teaching software engineering agree is necessary for anyone to earn an undergraduate degree in this field. So, even though SEEK does not represent a comprehensive curriculum, it does provide the groundwork for the design, implementation, and delivery of core educational units that make up a software engineering curriculum.

The body of SEEK is organized hierarchically into three levels. Knowledge areas (KAs) represent particular subdisciplines of software engineering that are generally recognized as significant parts of the body of software engineering knowledge that an undergraduate should learn. Each area is broken down into smaller divisions called units. Units represent individual thematic modules within an area. Finally, each unit is further subdivided into a set of topics. The contents can be packaged into different course names, generating specific curricula that cover the same core of software engineering knowledge.

Table 5.1 sums up the SEEK knowledge areas. The specific topics to be covered in each unit are outlined in SE2004. Table 5.1 also shows the percentage of in-class time a student should spend on each KA, defined in terms of Bloom's taxonomy (Knowledge -K-, Comprehension -C- and Application -AP-) [Bloom 1969]. Bloom's taxonomy is a classification of learning objectives within education proposed in 1956 by a committee of educators chaired by Benjamin Bloom. It refers to a classification of the different objectives that educators set for students (learning objectives). SE2004 works with what are termed contact hours (the minimum number of hours of in-class time needed to present the material to students).

For the purpose of comparison with GSwE, we have translated contact hours to percentage of in-class time, as shown in Table 5.1.

Table 5.1: SEEK Knowledge Areas and Knowledge Units

SEEK Knowledge Area	SEEK Units	Bloom's Taxonomy	Percentage of In-Class Time Per Unit	Percentage of In-Class Time Per Area
Computing Essentials	Computer science foundations	C/AP	28%	35%
	Construction technologies	C/AP	4%	
	Tools	C/AP	1%	
	Formal construction methods	K/C	2%	
Mathematical and Engineering Fundamentals	Mathematical foundations	C/AP	11%	18%
	Engineering foundations for software	K/C	5%	
	Engineering economics for software	K/C	2%	
Professional Practice	Group dynamics/psychology	K/C	1%	7%
	Communication skills	AP	2%	
	Professionalism	K/C	4%	
Software Modeling and Analysis	Modeling foundations	K/C	4%	11%
	Types of models	C/AP	2%	
	Analysis fundamentals	C/AP	1%	
	Requirements fundamentals	K/C	1%	
	Eliciting requirements	C	1%	
	Requirements specification & documentation	K/AP	1%	
	Requirement validation	C/AP	1%	
Software Design	Design concepts	C/AP	1%	9%
	Design strategies	C/AP	1%	
	Architectural design	K/AP	2%	
	Human-computer interface design methods	C/AP	2%	
	Detailed design	C/AP	2%	
	Design support tools and evaluation	K/AP	1%	
Software Verification and Validation	Verification and Validation terminology and foundations	K/AP	1%	9%
	Reviews	AP	1%	
	Testing	AP/C	4%	
	Human-computer UI testing and evaluation	C/AP	1%	
	Problem analysis and reporting	C	1%	
Software Evolution	Evolution process	K	1%	2%
	Evolution activities	K	1%	
Software Process	Process concepts	K/C	1%	3%
	Process implementation		2%	
Software Quality	Software quality concepts and culture	K	0.5%	3%

SEEK Knowledge Area	SEEK Units	Bloom's Taxono my	Percentage of In-Class Time Per Unit	Percentage of In-Class Time Per Area
	Software quality standards	K	1%	
	Software quality process	K/C	1%	
	Process assurance	K/C	0.5%	
Software Management	Management concepts	K	0.5%	4%
	Project planning	C/AP	1%	
	Project personnel and organization	K	1%	
	Project control	K/C	0.5%	
	Software configuration	K/C	1%	

5.3 Graduate Software Engineering Curriculum

The Curriculum Guidelines for Graduate Degree Programs in Software Engineering ([Graduate Software Engineering 2009] were developed to help create new and improve existing professional master programs in software engineering. The project was started in 2007 by the Stevens Institute of Technology and sponsored by the U.S. Department of Defense. Both the IEEE Computer Society and ACM supported the initiative, and about 100 experts from industry, government, and academia have participated in the project to date. One of the main outputs of GSwE2009 is the core body of knowledge (CBOK) to be covered by a professional SE master program. See Table 5.2.

Table 5.2: CBOK Knowledge Areas and Knowledge Units

CBOK Knowledge Area	CBOK Units	Bloom's Taxonomy	Percentage of Contact Hours
A. Ethics and Professional Conduct	1. Social, legal, and historical issues	C	3%
	2. Codes of ethics and professional conduct	C/AP	
	3. The nature and role of SE standards	C	
B. Systems Engineering	1. Systems Engineering Concepts	C	5%
	2. Systems Engineering Lifecycle Mgmt.	C	
	3. Requirements	C/AP	
	4. System Design	C/AP	
	5. Integration and Verification	C	
	6. Transition and Validation	C	
	7. Operation, Maintenance and Support	C	
C. Requirements Engineering	1. Fundamentals of Requirements Engineering	C/AP	14%
	2. Requirements Engineering Process	C	
	3. Initiation and Scope Definition	AP	
	4. Requirements Elicitation	AP	
	5. Requirements Analysis	AN	
	6. Requirements Specification	AP	
	7. Requirements Validation	AP	

CBOK Knowledge Area	CBOK Units	Bloom's Taxonomy	Percentage of Contact Hours
	8. Practical Considerations	C/AP	
D. Software Design	1. Software Design Fundamentals	C/AP	21%
	2. Key Issues in Software Design	AP	
	3. Software Structure and Architecture	AP/AN	
	4. Sw. Design Quality Analysis and Evaluation	AP	
	5. Software Design Notations	AP	
	6. Software Design Strategies and Methods	AP/AN	
E. Software Construction	1. Software Construction Fundamentals	AP	4%
	2. Managing Construction	AP	
	3. Practical Considerations	AP	
F. Testing	1. Testing Fundamentals	AP	10%
	2. Test Levels	AP	
	3. Testing Techniques	AP	
	4. Test-Related Measures	AP/AN	
	5. Test Process	C/AP	
G. Software Maintenance	1. Software Maintenance Fundamentals	C	7%
	2. Key Issues in Software Maintenance	AP	
	3. Maintenance Process	AP	
	4. Techniques for Maintenance	AP	
H. Configuration Management	1. Management of the CM Process	C/AP	5%
	2. Configuration Identification	AP	
	3. Configuration Control	AP	
	4. Configuration Status Accounting	AP	
	5. Software Release Management and Delivery	AP	
I. SE Management	1. Software Project Planning	AP	16%
	2. Risk Management	AP	
	3. Sw. Project Organization and Enactment	AP	
	4. Review and Evaluation	C	

CBOK Knowledge Area	CBOK Units	Bloom's Taxonomy	Percentage of Contact Hours
	5. Closure	C	
	6. Software Engineering Measurement	AP	
	7. Engineering Economics	C	
J. SE Process	1. Process Implementation and Change	C/AP	7%
	2. Process Definition	C	
	3. Process Assessment	AP	
	4. Product and Process Measurement	AP	
K. Software Quality	1. Software Quality Fundamentals	AP	8%
	2. Software Quality Management Processes	AP	
	3. Verification and Validation (V&V)	AP	

CBOK is designed to cover about 50% of master program contents. The reason behind this decision is to provide a flexible framework for the development of graduate SE programs. The other 50% should be covered by detailing the CBOK contents or focusing on a chosen application domain.

Like SE2004, CBOK is organized around knowledge areas, which are decomposed into units and further into topics. Table 5.2 shows knowledge areas and units, along with the in-class time for each KA as a percentage of the entire program and the recommended level to which a student should learn each KA according to Bloom's taxonomy. GSwE2009 uses the Comprehension (C), Application (AP) and Analysis (AN) levels. GSwE2009 also differs from SE2004 as to the recommended contact hours for each area and unit. GSwE2009 works with percentages of hours for each area, which apply only to the core of the program. The program core represents approximately 50% of the curriculum. In this context, the percentages are set to be considered as general high-level guidance, not as a precise curriculum specification.

5.4 CMMI for Development

CMMI for Development (CMMI-DEV) consists of best practices that address development activities applied to products and services. It addresses practices that cover the product's lifecycle from conception through delivery and maintenance [CMMI 2010]. CMMI-DEV proposes 22 process areas grouped into four categories, as shown in Table 5.3.

Table 5.3: Categories, Process Areas, and Maturity Levels in CMMI-DEV

Category	Process Area	Maturity Level
Engineering	Requirements Development	3
	Technical Solution	3
	Validation	3
	Verification	3

Category	Process Area	Maturity Level
	Product Integration	3
Project Management	Integrated Project Management	3
	Project Monitoring and Control	2
	Project Planning	2
	Quantitative Project Management	4
	Requirements Management	2
	Risk Management	3
	Supplier Agreement Management	2
Process Management	Organizational Process Definition	3
	Organizational Process Focus	3
	Organizational Performance Management	5
	Organizational Process Performance	4
	Organizational Training	3
Support	Causal Analysis and Resolution	5
	Configuration Management	2
	Decision Analysis and Resolution	3
	Measurement and Analysis	2
	Process and Product Quality Assurance	2

Figure 5.1 illustrates the information provided by each process area. This paper looks at what knowledge standard educational programs provide to enable graduates to help their software organizations to develop in conformity with CMMI-DEV. We have focused on the specific goals and specific practices for each process area. Specific goals are required model components that describe unique characteristics that must be present to satisfy a particular process area. Specific practices describe activities expected to lead to the achievement of the specific goals of a process area.

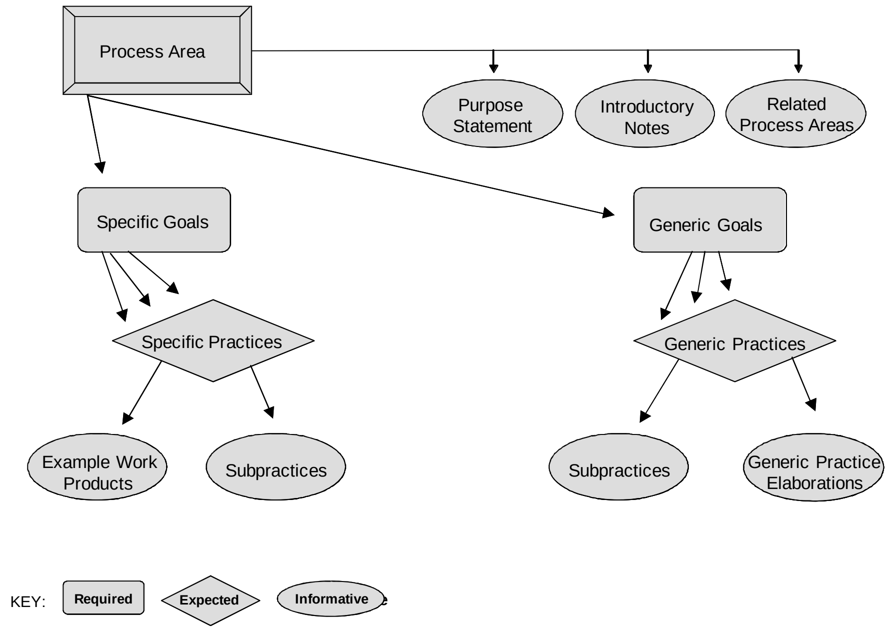

Figure 5.1: Model Components [CMMI 2010]

5.5 SE2004 and GSwE2009 Provision for CMMI-DEV Practices

In the following section, we analyze the extent to which the core knowledge provided by SE2004 and GSwE2009 qualifies graduates of programs based on these standards to enact practices for each of the 22 process areas proposed by CMMI-DEV. In other words, we will examine whether the SE2004 and GSwE2009 cores provide knowledge to satisfy the goals considered important for making improvements in each process area.

This analysis is based on the KA and unit descriptors proposed by SE2004 and GSwE2009, as our focus is to provide a global analysis of what knowledge these standards provide to achieve key practices in CMMI-DEV. We are aware that only professionals with years of experiences can reliably perform a specific practice at the maximum capability level (level 3, defined). By examining the coverage provided by the SE2004 or GSwE2009 cores to perform a practice, on the other hand, we are approaching the issue from the angle of the academic knowledge required to perform that practice. As already mentioned, this study sets out to examine how much of this knowledge the SE2004 and GSwE2009 cores cover. We do not intend to determine whether a graduate of an SE2004- or GSwE2009-based program is able to perform a particular practice. This would be a far from easy undertaking, as other factors influence this finding.

Below, we present the analysis performed by each process area category. To do this, we describe the relationship of the specific practices in each process area within each category to the KA in the respective educational standard. This relationship exists if the corresponding educational standard contains the core knowledge needed to perform the respective practice. If a standard does not provide any of the knowledge required to perform a particular practice, it will be labeled with a non-coverage icon (). If the knowledge is clearly insufficient, it will be marked with a partial coverage icon ().Finally, if the knowledge is likely to be sufficient, we use the coverage icon (). When applicable, we have also indicated the source KAs and units of the respective knowledge for each practice.

5.5.1 Engineering

According to Table 5.3, the engineering category covers five process areas: Requirements Development, Technical Solution, Verification, Validation, and Product Integration. As we will discuss in this section, neither SE2004 nor GSwE2009 provide full coverage for any of the process areas. However, we have found that graduates of GSwE2009-based programs are better qualified to perform the practices in these process areas, save product integration, where training is clearly deficient in both cases. On the other hand, graduates are better qualified for the Requirements Development, Technical Solution, and Verification process areas.

5.5.1.1 Requirements Development

Table 5.4 shows the analysis for the Requirements Development process area. The purpose of requirements development is to elicit, analyze, and establish customer, product, and product component requirements. Generally, both SE2004 and GSwE2009 deliver knowledge required to perform the different practices involved in this process area within the software modeling and analysis KA and requirements engineering KA, respectively. Both standards set aside a similar core percentage in-class time to this type of knowledge, namely, 11 percent in SE2004 and 14 percent in GSwE2009. However, the knowledge is learned in more depth in GSwE2009. In SE2004, save for specific practice3.3, knowledge is generally learned at the knowledge and comprehension levels, whereas almost all the contents of this process area are delivered at the application level in GSwE2009. Graduate students then are able to apply what they have learned in a real setting and at the level required by CMMI much more proficiently than SE2004 undergraduate students. GSwE2009 recommends some knowledge for this process area within the system engineering KA.

There are some practices that are not covered by either standard. This is the case in SP 3.1, where neither standard explicitly addresses the knowledge necessary for defining scenarios. According to CMMI, this practice involves subpractices like developing operational concepts and scenarios that include operations, installation, development, maintenance, support, and disposal as appropriate, or developing a detailed operational concept, as products and product components are selected, that defines the interaction of the product, the end user, and the environment, and which satisfies the product’s operational, maintenance, support, and disposal needs. As CMMI is applicable for developing any product, this practice can be especially worthwhile in the case of specialized software domains, such as embedded systems. It is perhaps less applicable to traditional software and is therefore not part of the standard cores.

Specific practice SP 3.4 is another area to which educational standards should pay more attention, as both SE2004 and GSwE2009 deliver only partial knowledge. This practice involves analyzing requirements to balance stakeholder needs and constraints like cost, schedule, product or project performance, functionality, priorities, reusable components, maintainability, or risk. The cores of both standards include these issues only secondarily. SE2004 includes some concepts on risk analysis within the Requirements Fundamentals unit, but at the comprehension level, and GSwE2009 includes some issues related to constraints within the Fundamentals of Requirements Engineering unit, but again at the comprehension level.

Therefore, we can conclude that neither educational standard’s core provides enough coverage for performing all the Requirements Engineering process area practices with the maturity level required by CMMI. However, GSwE2009 graduate students have more applied knowledge for enacting such practices, which is delivered by the core.

Table 5.4: Requirements Development Process Area Analysis

Requirements Development (RD)	SE2004 KA	GSwE2009 KA
SP 1.1 Elicit Needs	Software Modeling and Analysis/Eliciting	Requirements Engineering/Requirements Elicitation
SP 1.2 Transform Stakeholder Needs into Customer Requirements	Software Modeling and Analysis/Requirements Fundamental /Analysis Fundamental	Requirements Engineering / Fundamentals of Requirements Engineering
SP 2.1 Establish Product and Product Component Requirements	Software Modeling and Analysis/Requirements Fundamental /Analysis Fundamentals	System Engineering /Requirements Specification C/AP Requirements Engineering/ Fundamentals of Requirements Engineering
SP 2.2 Allocate Product Component Requirements	Software Modeling and Analysis/Requirements Fundamental	System Engineering /Requirements Specification Requirements Engineering Fundamentals of Requirements Engineering
SP 2.3 Identify Interface Requirements	Software Modeling and Analysis/Requirements Fundamental	System Engineering /Requirements Specification C/AP Requirements Engineering Fundamentals of Requirements Engineering
SP 3.1 Establish Operational Concepts and Scenarios		
SP 3.2 Establish a Definition of Required Functionality and Quality Attributes	Software Modeling and Analysis/Modeling Fundamentals /Types of Models	Requirements Engineering Fundamentals of Requirements Engineering
SP 3.3 Analyze Requirements	Software Modeling and Analysis/Modeling Fundamentals /Types of Models	Requirements Analysis / Requirements Attributes
SP 3.4 Analyze Requirements to Achieve Balance	Software Modeling and Analysis/Analysis Fundamentals	Requirements Engineering/Initiation and Scope Definition
SP 3.5 Validate Requirements	Software Modeling and Analysis/Requirements Validation	Requirements Engineering / Requirements Validation

5.5.1.2 Technical Solution

The purpose of the Technical Solution process area is to select, design, and implement solutions to requirements. Table 5.5 shows the analysis for this process area. Both SE2004 and GSwE2009 provide fairly satisfactory knowledge for performing the practices to be enacted within this process area. There are some differences with respect to design-related and implementation-related practices.

Design-related practices occupy a greater proportion of GSwE2009 core: 21 percent, as opposed to 9 percent of SE2004. On the other hand, the two curricula also differ not only regarding the quantity but also the depth of knowledge related to these practices that they recommend. Accordingly, most of the knowledge related to design practices in GSwE2009 is taught at the application, or even analysis, level, SE2004-based programs deliver much of the knowledge at the knowledge, comprehension and application levels. This applies particularly to the knowledge required to perform practices SP 1.1 and SP 1.2.

Note that neither curriculum explicitly provides knowledge for performing practices SP 2.2, SP 2.4, and SP 3.2. This does not mean that graduates and postgraduates are not able to perform these practices, but it does mean that they will use any implicit knowledge from their training in the software development field for this purpose. Both curricula should formally set out the knowledge for performing these practices if graduates are to enact them as thoroughly as required by CMMI-DEV.

Implementation-related practices, that is, SP 3.1, cover coding and unit testing. In this case, the trend mentioned with respect to design practices is somewhat reversed. Thus, programming-related knowledge is included under the Computing Essentials KA in SE2004 and accounts for a significant percentage of the core, 23 percent delivered at the comprehension and application levels. GSwE2009 includes a Software Construction KA, which, although it has a very low percentage, 4 percent, delivers this knowledge at the application level. As both concepts are delivered at application level, they are labeled as covered in Table 5.5, as knowledge for unit testing is provided at the application level in both cases. The differences in the impact of this type of knowledge on both cores is consistent with the two programs' goals, that is, train undergraduate and postgraduate students, respectively, that play different roles in the development process.

Generally, we find that neither of the educational standards explicitly provides knowledge for thoroughly performing all the practices in the Technical Solution process area. Even so, GSwE2009-based program graduates will have more in-depth knowledge (more applied and with more in-class time) for enacting this process area than SE2004-based program graduates.

Table 5.5: Technical Solution Process Area Analysis

Technical Solution (TS)	**SE2004 KA**	**GSwE2009 KA**
SP 1.1 Develop Alternative Solutions and Selection Criteria	Software Design/Design Concepts Software Design/Architectural Design ◕	Software Design/Software Design Fundamentals Software Design/Software Structure and Architecture Software Design/Software Design Quality Analysis and Evaluation ●
SP 1.2 Select Product Component Solutions	Software Design/Design Concepts Software Design/Architectural Design ◕	Software Design/Software Design Fundamentals Software Design/Software Structure and Architecture Software Design/Software Design Quality Analysis and Evaluation ●
SP 2.1 Design the Product or Product Component	Software Design ●	Software Design/ Software Design Notations Software Design / Strategies and Methods Software Design / Software Structure

Technical Solution (TS)	SE2004 KA	GSwE2009 KA
		and Architecture ●
SP 2.2 Establish a Technical Data Package	○	○
SP 2.3 Design Interfaces Using Criteria	Software Design/Detailed Design AP ●	Software Design/Key Issues in Software Design AP ●
SP 2.4 Perform Make, Buy, or Reuse Analyses	○	○
SP 3.1 Implement the Design	Computing Essentials Software Verification & Validation/Terminology/Unit Testing ●	Software Construction /Software Construction Fundamentals/Managing Construction/Practical Considerations Testing/ Fundamentals, Test Levels Testing Techniques ●
SP 3.2 Develop Product Support Documentation	○	○

5.5.1.3 Verification

According to CMMI, the purpose of verification is to ensure that selected work products meet their specified requirements.

Table 5.6 shows how verification is covered by the educational standards. In general, this process is covered by both SE2004 and GSwE2009. They both recommend knowledge to be delivered at the application level, although GSwE2009 sets aside a greater percentage of in-class time for this type of knowledge than SE2004. So, the Verification & Validation KA in SE2004 includes the knowledge related to this process area with an in-class time of 9 percent of the core. However, there is a special-purpose Testing KA in GSwE2009 with an in-class time of 10 percent, plus a Verification & Validation unit within the Software Quality KA with an in-class time of 8 percent, including content related to this practice.

Note that practices SP 1.3 and SP 3.2 are labeled as partially covered in

Table 5.6for both programs. SP 1.3 involves defining verification criteria for comparing the results of the verification methods selected in SP 1.2. CMMI mention standards, organizational policies, proposals and agreement, and test types as sources for defining these criteria. Neither SE2004 nor GSwE2009 explicitly mention this type of knowledge. However, as their cores both provide knowledge about verification methods, students are likely to learn the criteria applied in these methods, on which ground coverage has been labeled as partial. For SP 3.2, real verification results should be compared to establish verification criteria to determine acceptability. SE2004 contains knowledge unrelated to this practice, although it is delivered at the comprehension level. While GSwE2009 does not explicitly provide knowledge for enacting this practice, knowledge related to verification techniques and criteria should qualify graduates to perform the practice at least partially. However, both curricula should include explicit knowledge for performing these practices more systematically.

Table 5.6: Verification Process Area Analysis

Verification (VE)	SE2004 KA	GSwE2009 KA
SP 1.1 Select Work Products for Verification	Software Verification and Validation/ Testing ●	Testing/Fundamentals/Levels/Testing Techniques ●
SP 1.2 Establish the Verification Environment	Software Verification and Validation/ Testing ●	Testing/Fundamentals/Levels/Testing Techniques ●
SP 1.3 Establish Verification Procedures and Criteria	Software Verification and Validation/ Testing ◕	Testing/Fundamentals/Levels/Testing Techniques ◕
SP 2.1 Prepare for Peer Reviews	Software Verification and Validation/ Reviews ●	Software Quality / Verification &Validation ●
SP 2.2 Conduct Peer Reviews	Software Verification and Validation/ Reviews ●	Software Quality / Verification &Validation ●
SP 2.3 Analyze Peer Review Data	Software Verification and Validation/ Reviews ●	Software Quality / Verification &Validation ●
SP 3.1 Perform Verification	Software Verification and Validation/ Testing ●	Testing/Fundamentals/Levels/Testing Techniques ●
SP 3.2 Analyze Verification Results	Software Verification and Validation/Problem Analysis ◕	Testing/Fundamentals, Levels, Testing Techniques ◕

5.5.1.4 Validation

The purpose of validation is to demonstrate that a product or product component fulfills its intended use when placed in its intended environment. As shown in Table 5.7, GSwE2009 provides knowledge related to this process area at the application level within the Verification & Validation Unit of the Software Quality KA (with a total in-class time of 8 percent). This knowledge focuses on the description of general-purpose Verification & Validation concepts and some techniques, such as demonstrations, audits or analysis. The practices proposed by CMMI are labeled as partially covered, as the in-class time spent on them will be fairly low, and this standard makes no explicit reference to related knowledge.

SE2004 provides quite a lot less coverage for this process area than GSwE2009. It addresses only a few concepts within the Terminology and Foundations unit of the Verification & Validation KA with an in-class time of 1 percent. The Requirements Validation unit in the Software Modeling and Analysis KA does deliver some knowledge on requirements validation techniques, again with an in-class time of 1 percent. This knowledge provides very limited coverage for SP 2.1.

Note that SP 1.2 and SP 1.3 have been labeled as not covered by SE2004. This standard does not recommend any knowledge for performing these practices as systematically as proposed by CMMI, which includes subpractices, like identify requirements for the validation environment, identify customer-supplied products, identify test equipment and tools, identify validation resources that are available for reuse and modification, and plan the availability of resources in detail. As already mentioned, this does not mean that graduates will not be able to perform these practices, but that they will do so relying on any implicit knowledge they have and not as thoroughly as required by CMMI.

Table 5.7: Validation Process Area Analysis

Validation	SE2004 KA	GSwE2009 KA
SP 1.1 Select Products for Validation	Verification & Validation/V&V terminology and foundations ◕	Software Quality / Verification & Validation ◕
SP 1.2 Establish the Validation Environment	○	Software Quality / Verification & Validation ◕
SP 1.3 Establish Validation Procedures and Criteria	○	Software Quality / Verification & Validation ◕
SP 2.1 Perform Validation	Software Modeling and Analysis/Requirements Validation ◕	Software Quality / Verification & Validation ◕
SP 2.2 Analyze Validation Results	Software Verification and Validation/Problem Analysis ◕	Software Quality / Verification & Validation ◕

5.5.1.5 Product Integration

According to CMMI, the Product Integration process area "addresses the integration of product components into more complex product components or into complete products. The scope of this process area is to achieve complete product integration through progressive assembly of product components, in one stage or in incremental stages, according to a defined integration strategy and procedures." Generally, as Table 5.8 shows, neither SE2004 nor GSwE2009 detail the explicit knowledge for thoroughly performing the different practices in this process area at the capability level that is demanded by CMMI. SE2004only mentions integration testing within the Testing unit in the Verification & Validation KA at the comprehension level. On the other hand, integration testing and system testing are mentioned within the Test Levels unit of the Testing KA in GSwE2009. On this basis, SP 3.3 has been labeled as having partial, but very limited coverage.

Table 5.8: Product Integration Process Area Analysis

Products Integration (PI)	SE2004 KA	GSwE2009 KA
SP 1.1 Establish an Integration Strategy	○	○
SP 1.2 Establish the Product Integration Environment	○	○
SP 1.3 Establish Product Integration Procedures and Criteria	○	○

Products Integration (PI)	SE2004 KA	GSwE2009 KA
SP 2.1 Review Interface Descriptions for Completeness	○	○
SP 2.2 Manage Interfaces	○	○
SP 3.1 Confirm Readiness of Product Components for Integration	○	○
SP 3.2 Assemble Product Components	○	○
SP 3.3 Evaluate Assembled Product Components	Verification & Validation / Testing ◕	Testing/Testing Levels ◕
SP 3.4 Package and Deliver the Product or Product Component	○	○

5.5.2 Project Management

Project Management process areas cover the project management activities related to planning, monitoring, and controlling the project. The seven project management process areas in CMMI-DEV are Integrated Project Management, Project Monitoring and Control, Project Planning, Quantitative Project Management, Requirements Management, Risk Management, and Supplier Agreement Management.

As we will discuss in this section, Risk Management and Supplier Agreement Management are the only process areas fully covered by the GSwE2009 educational standard. The other process areas are not fully covered by either educational standard. However, we have found that graduates of programs based on both educational standards have moderate training in all of them. Note, however, that GSwE2009 graduates are better qualified to carry out the practices involved in these process areas. The process area in which graduates of programs based on both educational standards are least qualified is Supplier Agreement Management, where graduates of a program based on SE2004 receive no training at all in this area and graduates of a program based on GSwE2009 receive deficient training.

5.5.2.1 Project Planning

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

Table 5.9 shows, both SE2004 and GSwE2009 provide knowledge and applications to carry out the different specific project management process practices.

The percentage of in-class time spent on project planning in the SE2004 core is 4 percent, which is quite a lot lower than the 16 percent for GSwE2009. According to

Table 5.9, however, both educational standards, SE2004 and GSwE2009, detail explicit knowledge for thoroughly enacting different practices related to software project management, particularly practices regarding software project risk estimation, organization, planning, and identification. In this respect, both educational standards cover the training necessary to perform the practices recommended by CMMI for SP 1.1, SP 1.2, SP 1.3, SP 1.4, SP 2.1, SP 2.2, SP 2.4, and SP 2.7, at both the knowledge and application level, and coverage is complete at both the comprehension and application level.

Table 5.9: Project Planning Process Area Analysis

Project Planning (PP)	**SE2004 KA**	**GSwE2009 KA**
SP 1.1 Estimate the Scope of the Project	Software Management/Project Planning ●	Software Engineering Management / Software Project Organization and Enactment ●
SP 1.2 Establish Estimates of Work Product and Task	Software Management/Project Planning ●	Software Engineering Management / Software Project Planning ●
SP 1.3 Define Project Lifecycle Phases	Software Modeling and Analysis / Modeling Foundations Software Modeling and Analysis / Types of models Software Modeling and Analysis / Analysis Fundamentals ●	System Engineering Process / Process Definition / Lifecycle Models Software Engineering Management/ Software Project Planning ●
SP 1.4 Estimate Effort and Cost	Software Management/Project Planning ●	Software Engineering Management / Software Project Planning ●
SP 2.1 Establish the Budget and Schedule	Software Management/Project Planning ●	Software Engineering Management / Software Project Planning ●
SP 2.2 Identify Project Risks	Software Management/Project Planning ●	Software Engineering Management / Risk Management ●
SP 2.3 Plan Data Management	○	○

Project Planning (PP)	SE2004 KA	GSwE2009 KA
SP 2.4 Plan the Project’s Resources	Software Management/Project Planning Project Personnel and Organization ●	Software Engineering Management / Software Project Planning Software Engineering Management / Software Project Organization and Enactment ●
SP 2.5 Plan Needed Knowledge and Skills	Software Management/Project Personnel and Organization ◕	Software Engineering Management / Software Project Planning ●
SP 2.6 Plan Stakeholder Involvement	Software Management / Project Personnel and Organization ◕	Software Engineering Management / Software Project Planning ◕
SP 2.7 Establish the Project Plan	Software Management/Project Planning ●	Software Engineering Management / Software Project Planning ●
SP 3.1 Review Plans That Affect the Project	○	Software Construction / Practical Consideration ◕
SP 3.2 Reconcile Work and Resource Levels	○	Software Engineering Management ◕
SP 3.3 Obtain Plan Commitment	○	○

However, there are two specific practices within the project planning process area, SP 3.1 and SP 3.2, which are not covered by the SE2004 educational standard at all. SE2004 does not specify any knowledge units qualifying graduates to perform as formally as proposed by CMMI. However, the educational standard GSwE2009 does propose some KAs that deliver training at the knowledge level that is useful for performing some of the practices recommended by the CMMI, like integrating and review plans that affect the project and renegotiate budgets and review schedules. GSwE2009 KAs do not cover all the practices recommended by CMMI in SP 3.2, like renegotiate stakeholder agreements, which they do not cover in as much detail as recommended by CMMI, for which reason both specific practices have been labeled as partially covered.

Note, on the other hand, that neither curriculum explicitly provides knowledge for performing practices SP 2.3 and SP 3.3. This does not mean that graduates of programs based on these curricula will not be able to perform the practices but that, to do so, they will make use of any implicit knowledge acquired as a result of their training in the software development field. Both curricula should thus specify knowledge for performing these practices as thoroughly as required by CMMI-DEV.

In short, the graduates of programs based on either educational standard are not qualified to perform all the specific practices in the project planning process area.

5.5.2.2 Integrated Project Management

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

Table 5.10 summarizes the analysis of this process area. The percentage in-class time assigned to this process area in SE2004 core is 3 percent, lower than GSwE2009's 7 percent. In the case of this process area, we can conclude that there is a big gap between the knowledge provided by the two educational standards. The knowledge provided by SE2004 educational standard covers SP 1.1, SP 1.3, SP 1.6, and SP 1.7 at the level of theoretical comprehension but not application, where SP 1.2 is the only specific practice that SE2004 covers fully. There are no specific knowledge units in SE2004 to cover the other specific practices in the integrated project management process area, SP 1.3, SP 1.4, SP 2.1, SP 2.2, and SP 2.3. Some noteworthy practices that are not part of SE2004 and are recommended by CMMI are: identify how conflicts will be resolved that arise among relevant stakeholders; address the causes of selected issues that can affect project objectives; and review and get agreement on commitments to address each critical dependency with those who are responsible for providing or receiving the work product.

On the other hand, we can conclude that students graduating from a program based on GSwE2009, are qualified at the knowledge and application level for all the specific practices, except SP 1.1, SP 2.1, SP 2.2, and SP 2.3, where the knowledge provided by the Software Engineering Process, Process Definition and Software Engineering Management, and Software Project Organization and Enactment KAs partially cover the knowledge required by CMMI to perform these practices.

From this analysis, we find that the core of neither educational standard provides full coverage of the knowledge to perform all the integrated Project Management process area practices at the maturity level demanded by CMMI. However, GSwE2009 graduate students have broader and more applied knowledge that covers the specific practices of the Integrated Project Management process area.

Table 5.10: Integrated Project Management Process Area Analysis

Integrated Project Management (IPM)	SE2004 KA	GSwE2009 KA
SP 1.1 Establish the Project's Defined Process	Software Engineering Process/ Process Definition	Software Engineering Process/ Process Definition
SP 1.2 Organizational Process Assets for Planning Project Activities	Software Management/Project Planning	Software Engineering Management / Software Project Planning
SP 1.3 Establish the Project's Work Environment	Software Process/ Process Concepts	Software Engineering Process/ Process Implementation and Change
SP 1.4 Integrate Plans		System Engineering / Integration and Verification Software Construction / Practical Consideration

Integrated Project Management (IPM)	SE2004 KA	GSwE2009 KA
SP 1.5 Manage the Project Using Integrated Plans	○	System Engineering / Integration and Verification Software Construction / Practical Consideration ●
SP 1.6 Establish Teams	Software Management/ Project Personnel and Organization ◕	Software Engineering Management /Software Project Organization and Enactment ●
SP 1.7 Contribute to Organizational Process Assets	Software process/ Process Implementation ◕	Software Engineering Process /Process Definition ●
SP 2.1 Manage Stakeholder Involvement	○	Software Engineering Management/ Software Project Organization and Enactment ◕
SP 2.2 Manage Dependencies	○	Software Engineering Management/ Software Project Organization and Enactment ◕
SP 2.3 Resolve Coordination Issues	○	Software Engineering Management/ Software Project Organization and Enactment ◕

5.5.2.3 Quantitative Project Management

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

After analyzing SE2004 and GSwE2009 (see Table 5.11), we find that the educational standard GSwE2009 fully covers all the specific practices of the Quantitative Project Management process area at the knowledge and application levels. The only exception is SP 2.3, which is partially covered by GSwE2009, as there is no KA that covers all the practices, including identifying and analyzing potential actions and assessing the impact of the actions on subprocess performance in as much detail as recommended by CMMI.

On the other hand, the educational standard SE2004 provides less coverage than GSwE2009 of the knowledge that should be acquired to perform the specific practices in this process area. SE2004 fully covers just two specific practices, SP 1.1 and SP 1.2, at both the knowledge and application levels. However, it partially covers SP 1.3, SP 1.7, and SP 2.2 at the knowledge level only. It does not cover SP 2.1 and SP 2.3 at all. As far as the specific practices that it does not cover, note that none of the knowledge units in a SE2004-based program teach the knowledge necessary to perform these practices.

In short, we can conclude that the core of educational standard GSwE2009 fully covers the Quantitative Project Management process area, with the exception of one specific practice (SP 2.3), where deeper knowledge of statistical analysis would be required for full coverage. However, SE2004 partially covers the knowledge required to prepare for quantitative management but does not provide the knowledge required to cover the specific practices required to quantitatively manage the project.

Table 5.11: Quantitative Project Management Process Area Analysis

Quantitative Project Management (QPM)	SE2004 KA	GSwE2009 KA
SP 1.1 Establish the Project's Objectives	Software Quality/ Software Quality Processes Software Engineering Management / Software Project Planning ●	Software Quality/ Software Quality Management Processes ●
SP 1.2 Compose the Defined Process	Software Quality/ Software Quality Processes Software Engineering Management / Software Project Planning ●	Software Quality/ Software Quality Management Processes ●
SP 1.3 Select Subprocesses and Attributes	Software Quality/ Software Quality Processes ◕	Software Quality/ Software Quality Fundamentals Software Quality Management Processes ●
SP 1.4 Select Measures and Analytic Techniques	Software Quality/ Software Quality Processes ◕	Software Quality/ Software Quality Management Processes ●
SP 2.1 Monitor the Performance of Selected Subprocesses	○	Software Quality/ Software Quality Management Processes ●
SP 2.2 Manage Project Performance	Software Management/Project Planning ◕	Software Quality/ Software Quality Management Processes Software Engineering Management /Risk Management ●
SP 2.3 Perform Root Cause Analysis	○	Software Quality/ Software Quality Management Processes ◕

5.5.2.4 Requirements Management

The purpose of Requirements Management is to manage requirements of the project's products and product components and to ensure alignment between those requirements and the project's plans and work products.

Looking at the analysis (see Table 5.12), we find that both standards cover specific practices related to requirements management. We should point out two specific practices, SP 1.3 and SP 1.4, of which students graduating from an SE2004-based program have partial or no knowledge, unlike GSwE2009, where they have full or partial knowledge of these specific practices, respectively.

To summarize this analysis, we can say that the core of neither educational standard provides full coverage of knowledge for performing all the specific practices in the Requirements Management process area with the maturity level required by CMMI. However, GSwE2009 graduate students have broader and more detailed knowledge of specific practices like managing requirements changes and maintaining bidirectional traceability of requirements.

Table 5.12: Requirements Management Process Area Analysis

Requirements Management (REQM)	SE2004 KA	GSwE2009 KA
SP 1.1 Understand Requirements	Software Modeling and Analysis/Requirements Fundamentals ●	Requirements Engineering/ Requirements Specification ●
SP 1.2 Obtain Commitment to Requirements	Software Modeling and Analysis/ Requirements Fundamentals ◕	Requirements Engineering/ Requirements Specification ◕
SP 1.3 Manage Requirements Changes	Software Modeling and Analysis/Requirements Fundamentals ◕	Requirements Engineering/Practical Considerations ●
SP 1.4 Maintain Bidirectional Traceability of Requirements	○	Requirements Engineering/Practical Considerations ◕
SP 1.5 Ensure Alignment Between Project Work and Requirements	Software Modeling and Analysis / Requirements Validation ●	Requirements Engineering /Requirements Validation ●

5.5.2.5 Risk Management

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

The analysis of this process area (see Table 5.13) shows that GSwE2009 fully covers all the knowledge required to perform all the specific practices.

Table 5.13: Risk Management Process Area Analysis

Risk Management (RSKM)	SE2004 KA	GSwE2009 KA
SP 1.1 Determine Risk Sources and Categories	Software Modeling and Analysis/ Analysis fundamentals ◕	Software Engineering Management/ Risk Management ●
SP 1.2 Define Risk Parameters	Software Modeling and Analysis/ Analysis fundamentals ◕	Software Engineering Management / Risk Management ●
SP 1.3 Establish a Risk Management Strategy	○	Software Engineering Management/ Risk Management ●
SP 2.1 Identify Risks	Software Modeling and Analysis/ Analysis fundamentals ◕	Software Engineering Management/ Risk Management ●
SP 2.2 Evaluate, Categorize, and Prioritize Risks	Software Modeling and Analysis/ Analysis fundamentals ◕	Software Engineering Management/ Risk Management ●
SP 3.1 Develop Risk Mitigation Plans	○	Software Engineering Management/ Risk Management ●
SP 3.2 Implement Risk Mitigation Plans	○	Software Engineering Management/ Risk Management Software Engineering Management/ Software Project Organization and Enactment ●

The same does not apply, however, to the knowledge covered by the SE2004 standard. Although this standard partially covers SP 1.1, SP 1.2, SP 2.1, and SP 2.2, it omits knowledge of specific practices related to the establishment of risk management strategies (SP 1.3) and the development and implementation of risk mitigation plans SP 3.1 and SP 3.2. Therefore, we can conclude that students graduating from a SE2004-based educational program do not know how to develop and implement risk management and mitigation plans and strategies, although they do have theoretical knowledge for analyzing, defining, identifying, evaluating, categorizing, and prioritizing risks.

We find that for this process area, the core of the GSwE2009 educational standard provides full coverage and application for performing all the Risk Management process area practices with the maturity level demanded by CMMI. However, the SE2004 educational program only partially covers this process area and does not cover knowledge on the development and implementation of risk mitigation plans at all.

5.5.2.6 Project Monitoring and Control

The purpose of Project Monitoring and Control is to provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.

After analyzing the Project Monitoring and Control process area (see Table 5.14), we can conclude that both standards stipulate similar knowledge to be learned with respect to the specific practices of this process area, except for two specific practices, SP 1.1 and SP 1.7, where the knowledge provided by GSwE2009 is broader than that of SE2004. Both standards provide identical coverage of knowledge in the other specific practices, which does not totally cover all the specific practices in this process area.

Note that students graduating from programs based on both standards have all the knowledge that they require to perform SP 1.3 and SP 1.6. However, neither standard provides coverage for the knowledge necessary for performing the practices recommended by CMMI in SP 1.4 and SP 1.5, such as, documenting the results of data management activity reviews, documenting the results of stakeholder involvement in status reviews, and others.

Another noteworthy point is that both standards partially cover SP 2.1 (identify corrective actions), but there are no units or KAs in SE2004 and GSwE2009 that specifically cover their management (SP 2.2 and SP 2.3). Therefore, graduates of programs based on both educational standards should receive further training to be able to perform these practices in the detail recommended by CMMI.

After this analysis, we can conclude that the core of both educational standards does not fully cover the knowledge for performing all the specific practices of the Project Monitoring and Control process area with the maturity level required by CMMI and should additionally specify that both educational standards provide almost the same knowledge coverage for performing these specific practices, except that GSwE2009 graduate students have broader knowledge with respect to the practices related to SP 1.1 (monitor project planning parameters) and are provided with knowledge to partially cover SP 1.7 (conduct milestone reviews), a practice that is not covered at all by SE2004.

Table 5.14: Project Monitoring and Control Process Area Analysis

Project Monitoring and Control (PMC)	SE2004 KA	GSwE2009 KA
SP 1.1 Monitor Project Planning Parameters	Software Management / Project Control ◕	Software Engineering Management /Software Project Organization and Enactment ●
SP 1.2 Monitor Commitments	Software Management / Project Control ◕	Software Engineering Management / Review and Evaluation ◕
SP 1.3 Monitor Project Risks	Software Management/ Project Planning ●	Software Engineering Management/Risk Management ●
SP 1.4 Monitor Data Management	○	○
SP 1.5 Monitor Stakeholder Involvement	○	○
SP 1.6 Conduct Progress Reviews	Software Management/ Project Planning ●	Software Engineering Management /Software Project Organization and Enactment ●
SP 1.7 Conduct Milestone Reviews	○	Software Engineering Management/Closure ◕
SP 2.1 Analyze Issues	Software Management/ Project Control ◕	Software Engineering Management /Software Project Organization and Enactment ◕
SP 2.2 Take Corrective Action	○	○
SP 2.3 Manage Corrective Actions	○	○

5.5.2.7 Supplier Agreement Management

The purpose of Supplier Agreement Management is to manage the acquisition of products and services from suppliers. From the analysis shown in Table 5.15, we find that graduates of an educational program based on SE2004 have little knowledge of the specific practices covered by the Supplier Agreement Management process area. Similarly, graduates of GSwE2009-based educational programs also have little knowledge in performing specific practices in this process area, except for two specific practices, SP 1.3 and SP 2.1, where graduates are qualified to perform these practices, although not as thoroughly as recommended by CMMI. On this ground, both specific practices have been labeled as partially covered.

In short, we can conclude that the core of both educational standards provide negligible coverage of the knowledge required to perform all the practices in the Supplier Agreement Management process area with the maturity level demanded by CMMI. Note that GSwE2009 graduate students have knowledge for performing specific practices SP 1.3 and SP 2.1, although not as thoroughly as recommended by CMMI.

Table 5.15: Supplier Agreement Management Process Area Analysis

Supplier Agreement Management (SAM)	SE2004 KA	GSwE2009 KA
SP 1.1 Determine Acquisition Type	○	○
SP 1.2 Select Suppliers	○	○
SP 1.3 Establish Supplier Agreements	○	Software Engineering Management /Software Project Organization and Enactment ◕
SP 2.1 Execute the Supplier Agreement	○	Software Engineering Management / Software Project Organization and Enactment ◕
SP 2.2 Accept the Acquired Product	○	○
SP 2.3 Ensure Transition of Products	○	○

5.5.3 Process Management

Process management process areas contain the cross-project activities related to defining, planning, deploying, implementing, monitoring, controlling, appraising, measuring, and improving processes. The five process management process areas in CMMI-DEV are as follows: Organizational Process Definition, Organizational Process Focus, Organizational Performance Management, Organizational Process Performance, and Organizational Training. As discussed in the following, entry-level professionals that have graduated from educational programs based on either SE2004 or GSwE2009 are not qualified to properly perform any of the process areas in this category. All these process areas focus on organizational processes, on which the KAs of neither SE2004 nor GSwE2009 focus.

5.5.3.1 Organizational Process Definition

The purpose of Organizational Process Definition is to establish and maintain a usable set of organizational process assets, work environment standards, and rules and guidelines for teams. Table 5.16 summarizes the analysis of this process area.

Educational programs based on SE2004 and GSwE2009 do include specific practices within this process area:

- Programs based on both SE2004 and GSwE2009 include subjects, within the Software Process KA in SE2004, and in the Software Engineering Process KA inGSwE2009, that are helpful for learning SP 1.1 and SP 1.2. These specific practices are not fully covered, because, although students have theoretical knowledge, or even practical knowledge in the case of GSwE2009, this does not qualify them completely to establish which is the method or process best adapted to an organization in each case. We do not believe that an entry-level graduate could not perform these specific practices 100 percent reliably.
- SE2004 provides partial coverage for SP 1.7 because the Process Implementation—Team Process unit in the Software Process KA accounts for the knowledge required to identify work team types and how they work at the theoretical level. Again, GSwE2009 offers partial coverage, because, even though this knowledge is learned at the practical level, a

negligible amount of in-class time is spent on this subject in the Project Directing, Motivation, Conflict Resolution, and Team Building units of the Software Project Organization and Enactment KA.

On the other hand, neither SE2004 nor GSwE2009 include the following specific practices:

- Entry-level graduates of both programs will not have studied SP 1.4 specifically, but could, in both cases, use database knowledge that will be helpful for performing this specific practice.
- A similar thing applies to SP 1.5, but the knowledge at both the theoretical and practical level has to be much greater in this case, as much more thorough knowledge of both process assets and the actual organization is needed in order to establish and maintain an organization's process assets.
- SP 1.6 is not explicitly covered in either undergraduate and postgraduate programs, but professionals have to have good technological skills and be abreast of what process management technologies there are on the market and have sufficient knowledge of the company to be able to determine when a technological solution meets the company's process needs to perform this specific practice.

Table 5.16: Organizational Process Definition Process Area Analysis

Organizational Process Definition (OPD)	**SE2004 KA**	**GSwE2009 KA**
SP 1.1 Establish Standard Processes	Software process ◕	Software Engineering Process ◕
SP 1.2 Establish Lifecycle Model Descriptions	Software Process /Process Implementation ◕	Software Engineering Process /Process Definition ◕
SP 1.3 Establish Tailoring Criteria and Guidelines	Software Process / Process Implementation /Process Tailoring ◕	Software Engineering Process / Process Definition /Process Adaptation ◕
SP 1.4 Establish the Organization's Measurement Repository	○	○
SP 1.5 Establish the Organization's Process Asset Library	○	○
SP 1.6 Establish Work Environment Standards	○	○
SP 1.7 Establish Rules and Guidelines for Teams	Software Process /Process Implementation / Team Process ◕	Software Project Organization and Enactment /Project Directing / Motivation, Conflict Resolution /Team Building ◕

5.5.3.2 Organizational Process Focus

According to CMMI, the purpose of Organizational Process Focus is to plan, implement, and deploy organizational process improvements based on a thorough understanding of current strengths and weaknesses of the organization's processes and process assets. Let us discuss the analysis shown in Table 5.17.

In the case of the Organizational Process Focus process area, SE2004 does not specifically take into account the knowledge required to perform the respective specific practices. Although the Software Process KA does include a Software Engineering Process Improvement unit taught at the conceptual level, which would serve as a basis for entry-level graduates to perform the specific practices in this process area, they are not considered sufficient, as CMMI addresses these specific practices from the viewpoint of the organization as a whole and SE2004 as specific to software.

The findings for GSwE2009 with respect to the organizational process focus process area are as follows:

- Students can acquire knowledge that can be useful for performing SP 1.1, SP 1.2, and SP 1.3 in the Process Assessment unit of the Software Engineering Process KA at the application level, taking into account that they apply specifically to software development organizations in GSwE2009.
- The knowledge required to perform SP 2.1 and SP 2.2 can be learned in the Project Organization and Enactment KA at the application level, where improvement project planning is construed as just another project to be planned within the organization. However, there is no guarantee that an entry-level graduate of this type of program will be successful at deploying these two specific practices, because their knowledge will be confined to only software projects.
- No specific knowledge for performing SP 3.1, SP 3.2, and SP 3.3 is included, but knowledge of process and change implementation is learned at the application level in the Process and Change Implementation unit of the Software Engineering Process KA, which can be helpful for an entry-level graduate to perform these specific practices. However, there is no guarantee that graduates will have the knowledge to perform them successfully, as they have learned them for software processes, whereas they are addressed in CMMI at the organizational level.

Table 5.17: Organizational Process Focus Process Area Analysis

Organizational Process Focus (OPF)	**SE2004 KA**	**GSwE2009 KA**
SP 1.1 Establish Organizational Process Needs	○	Software Engineering Process /Process Assessment ◕
SP 1.2 Appraise the Organization's Processes	○	Software Engineering Process /Process Assessment ◕
SP 1.3 Identify the Organization's Process Improvements	○	Software Engineering Process /Process Assessment ◕
SP 2.1 Establish Process Action Plans	○	○
SP 2.2 Implement Process Action Plans	○	○
SP 3.1 Deploy Organizational Process Assets	○	○
SP 3.2 Deploy Standard Processes	○	○

SP 3.3 Monitor the Implementation	○	○
SP 3.4 Incorporate Experiences into Organizational Process Assets	○	○

5.5.3.3 Organizational Performance Management

The purpose of Organizational Performance Management is to proactively manage the organization's performance to meet its business objectives. The analysis of this process area is summarized in

Table 5.18.

SP 1.1, SP 1.2, and SP 1.3 are related to the business part of the company, and neither SE2004 nor GSwE2009 include training content related to these specific practices. For SP 2.1, 2.2, 2.3, and 2.4, SE2004 includes the Software Process KA, where the Software Engineering Process Improvement unit, studied at the conceptual level, would serve as a basis for entry-level graduates to perform the specific practice in this process area. But this is not considered sufficient, as CMMI addresses these practices for the organization, and SE2004 is specific for software. GSwE2009 does not consider any of these specific practices.

Given the relationship that SP 3.1, SP 3.2, and SP 3.3 have with planning and control, an entry-level graduate of a program based on SE2004 could apply the knowledge acquired in the Software Management KA to a deployment project, and an entry-level graduate of a program based on GSwE2009 could apply the knowledge learned in the Software Engineering Management KA. In neither case, however, would they have any guarantee of success, as, in both cases, the knowledge would be specific to software, whereas CMMI is general purpose.

Table 5.18: Organizational Performance Management Process Area Analysis

Organizational Performance Management (OPM)	SE2004 KA	GSwE2009 KA
SP 1.1 Maintain Business Objectives	○	○
SP 1.2 Analyze Process Performance Data	○	○
SP 1.3 Identify Potential Areas for Improvement	○	○
SP 2.1 Elicit Suggested Improvements	○	○
SP 2.2 Analyze Suggested Improvements	○	○
SP 2.3 Validate Improvements	○	○
SP 2.4 Select and Implement Improvements for Deployment	○	○
SP 3.1 Plan the Deployment	○	○

SP 3.2 Manage the Deployment	O	O
SP 3.3 Evaluate Improvement Effects	O	O

5.5.3.4 Organizational Process Performance

The purpose of Organizational Process Performance is to establish and maintain a quantitative understanding of the performance of selected processes in the organization’s set of standard processes in support of achieving quality and process performance objectives, and to provide process performance data, baselines, and models to quantitatively manage the organization’s projects. Table 5.19 summarizes the analysis of this process area.

Table 5.19: Organizational Process Performance Process Area Analysis

Organizational Process Performance (OPP)	SE2004 KA	GSwE2009 KA
SP 1.1 Establish Quality and Process Performance Objectives	O	O
SP 1.2 Select Processes	O	O
SP 1.3 Establish Process Performance Measures	O	O
SP 1.4 Analyze Process Performance and Establish Process Performance Baselines	O	O
SP 1.5 Establish Process Performance Models	O	O

The specific practices in this process area are very much focused on guaranteeing quality parameters and establishing organizational performance measures. Therefore, entry-level graduates of SE2004- or GSwE2009-based programs would only have some quality and measurement knowledge, learned, in the case of SE2004, in the Software Quality and Software Management KAs (at the theoretical level) and, in the case of GSwE2009, in the Product and Process Measurement unit of the Software Engineering Process KA (at the application level) and the Software Quality Management Processes unit of the Software Quality KA (at the application level). In both SE2004 and GSwE2009, knowledge would target software project and processes exclusively, so that there is no guarantee that they can apply generally in any part of the organization as CMMI suggests.

5.5.3.5 Organizational Training

The purpose of Organizational Training is to develop employees’ skills and knowledge so they can perform their roles effectively and efficiently. Table 5.20 summarizes the analysis made. Neither SE2004 nor GSwE2009 have specific requirements that call for employees to acquire the skills needed to perform the specific practices defined for this process area. Entry-level graduates of either undergraduate or postgraduate programs have not been trained to provide training or decide which practices best meet the specific needs of an organization.

Table 5.20: Organizational Training Process Area Analysis

Organizational Training (OT)	SE2004 KA	GSwE2009 KA
SP 1.1 Establish Strategic Training Needs	○	○
SP 1.2 Determine Which Training Needs Are the Responsibility of the Organization	○	○
SP 1.3 Establish an Organizational Training Tactical Plan	○	○
SP 1.4 Establish a Training Capability	○	○
SP 2.1 Deliver Training	○	○
SP 2.2 Establish Training Records	○	○
SP 2.3 Assess Training Effectiveness	○	○

5.5.4 Support

Support process areas contain the activities oriented to measuring, quality control, causal and decision analysis, and configuration management to endow the organization. The five support process areas in CMMI-DEV are Causal Analysis and Regression, Configuration Management, Decision Analysis and Resolution, Measurement and Analysis, and Process and Product Quality Assurance. As discussed below, entry-level graduates of educational programs based on either SE2004 or GSwE2009 cannot reliably perform the Causal Analysis and Regression and Decision Analysis and Resolution Process Areas. Only entry-level graduates of GSwE2009-based programs can reliably perform the Measurement and Analysis Process Area. Entry-level graduates of GSwE2009-based programs can successfully perform the Configuration Management and Process and Products Quality Assurance Process Areas, but success is not guaranteed if they are performed by entry-level graduates of SE2004-based programs who lack practical knowledge of these areas.

5.5.4.1 Causal Analysis and Resolution

The purpose of Causal Analysis and Resolution is to identify causes of selected outcomes and take action to improve process performance. After analyzing both SE2004 and GSwE2009 (see Table 5.21), we can say that neither contains specific KAs for learning knowledge necessary to perform the specific practices proposed by the CMMI for this process area. For example, CMMI proposes techniques like Pareto analysis, histograms, box and whisker plots for attributes, and others that require detailed and specific statistical knowledge that neither educational standards SE2004 or GSwE2009 recommend in their core curricula.

Table 5.21: Causal Analysis and Resolution Process Area Analysis

Causal Analysis and Resolution (CAR)	SE2004 KA	GSwE2009 KA
SP 1.1 Select Outcomes for Analysis	○	○
SP 1.2 Analyze Causes	○	○
SP 2.1 Implement Action Proposals	○	○
SP 2.2 Evaluate the Effect of Implemented Actions	○	○
SP 2.3 Record Causal Analysis Data	○	○

5.5.4.2 Configuration Management

The purpose of Configuration Management is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits. Undergraduate programs based on SE2004 do include basic knowledge on this process area, and programs based on GSwE2009 also provide practical knowledge, so an entry-level graduate of a postgraduate program will be better able to develop this process area (see Table 5.22).

Table 5.22: Configuration Management Process Area Analysis

Configuration Management (CM) (	SE2004 KA	GSwE2009 KA
SP 1.1 Identify Configuration Items	Software Management /Software Configuration ◕	Configuration Management /Configuration Identification ●
SP 1.2 Establish a Configuration Management System	Software Management / Software Configuration ◕	Configuration Management / Management of the CM Process ●
SP 1.3 Create or Release Baselines	Software Management / Software Configuration ◕	Configuration Management / Configuration Identification ●
SP 2.1 Track Change Requests	Software Management / Software Configuration ◕	Configuration Management / Configuration Control ●
SP 2.2 Control Configuration Items	Software Management / Software Configuration ◕	Configuration Management / Configuration Status Accounting ●
SP 3.1 Establish Configuration Management Records	Software Management / Software Configuration ◕	Configuration Management / Configuration Status Accounting ●
SP 3.2 Perform Configuration Audits	Software Management / Software Configuration ◕	Configuration Management / Configuration Status Accounting ●

5.5.4.3 Decision Analysis and Resolution

The Decision Analysis and Resolution process area is oriented to analyzing possible decisions using a formal evaluation process that evaluates identified alternatives against established criteria.

Although it is true that neither SE2004 nor GSwE2009 cover the formal process required to apply the specific practices in this process area (see Table 5.23), the graduates of these programs do have some related knowledge that they could apply informally to perform these practices. For example

- Although specific practices SP 1.2, SP 1.3, and SP 1.6 have been labeled as not formally covered by either SE2004 or GSwE2009, software engineers that have completed a SE2004-based program have enough practical knowledge of requirements prioritization and evaluation to be able to select which set should be developed, learned in the Analysis Fundamentals unit of the Software Modeling and Analysis KA, whose mechanism can be generalized to implement SP 1.2, SP 1.3, and SP 1.6. On the other hand, students that have taken a postgraduate program based on GSwE2009 also have practical, although fairly basic, knowledge on prioritization and establishment of evaluation criteria that they learn by applying decision-making techniques to the economics area of a project with an organization in the Engineering Economics unit of the Software Project Organization and Enactment KA.
- As regards specific practices SP 1.4 and SP 1.5, entry-level graduates of an undergraduate course based on SE2004 study Formal Experiments as an optional rather than compulsory subject in the Human-Computer User Interface Testing and Evaluation unit of the Software Verification and Validation KA. On the other hand, entry-level graduates of a postgraduate program based on GSwE2009 do not learn knowledge related to the development of case studies, experiments or surveys, which they learn by practice as part of their doctoral studies.

However, although all the practices related to this process area have been marked as not covered in Table 5.23, explicit knowledge for performing them with the capability level required by CMMI-DEV is not mentioned in the cores of either program.

Table 5.23: Decision Analysis and Resolution Process Area Analysis

Decision Analysis and Resolution (DAR)	SE2004 KA	GSwE2009 KA
SP 1.1 Establish Guidelines for Decision Analysis	○	○
SP 1.2 Establish Evaluation Criteria	○	○
SP 1.3 Identify Alternative Solutions	○	○
SP 1.4 Select Evaluation Methods	○	○
SP 1.5 Evaluate Alternative Solutions	○	○
SP 1.6 Select Solutions	○	○

5.5.4.4 Measurement and Analysis

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

From the analysis, which is summarized in Table 5.24, it is clear an entry-level graduate of an undergraduate program based on SE2004 formally only has some theoretical knowledge related to the Measurement and Analysis process area, knowledge that they learn in the Software Process KA at a very theoretical level, and software processes are only part of what should be measured in this co-curricular process area.

Table 5.24: Measurement and Analysis Process Area Analysis

Measurement and Analysis (MA)	SE2004 KA	GSwE2009 KA
SP 1.1 Establish Measurement Objectives	○	Software Engineering Management / Software Engineering Measurement ●
SP 1.2 Specify Measures	○	Software Engineering Management / Software Engineering Measurement ●
SP 1.3 Specify Data Collection and Storage Procedures	○	Software Engineering Management / Software Engineering Measurement ●
SP 1.4 Specify Analysis Procedures	○	Software Engineering Management / Software Engineering Measurement ●
SP 2.1 Obtain Measurement Data	○	Software Engineering Management / Software Engineering Measurement ●
SP 2.2 Analyze Measurement Data	○	Software Engineering Management / Software Engineering Measurement ●
SP 2.3 Store Data and Results	○	Software Engineering Management / Software Engineering Measurement ●
SP 2.4 Communicate Results	○	Software Engineering Management / Software Engineering Measurement ●

On the other hand, entry-level graduates of a postgraduate program based on GSwE2009 will have useful knowledge for performing specific practices proposed by CMMI for the Measurement and Analysis process area, learned at the application level in the Software Engineering Measurement unit of the Software Engineering Management KA.

5.5.4.5 Process and Product Quality Assurance

The purpose of Process and Product Quality Assurance is to provide staff and management with objective insight into processes and associated work products. Table 5.25 shows the results of this analysis.

Table 5.25: Process and Product Quality Assurance Process Area Analysis

Process and Product Quality Assurance (PPQA)	SE2004 KA	GSwE2009 KA
SP 1.1 Objectively Evaluate Processes	Software Quality /Process Assurance	Software Quality / Software Quality Fundamentals / Software Quality Management Processes Software Engineering Process / Process Assessment
SP 1.2 Objectively Evaluate Work Products	Software Quality / Process Assurance	Software Quality / Software Quality Fundamentals / Software Quality Management Processes y Verification and Validation
SP 2.1 Communicate and Resolve Noncompliance Issues	Software Quality / Process Assurance	Software Quality / Software Quality Fundamentals y Software Quality Management Processes
SP 2.2 Establish Records	Software Quality / Process Assurance	Software Quality / Software Quality Fundamentals y Software Quality Management Processes

The study of SE2004 and GSwE2009 shows that entry-level graduates of programs conforming to both standards can perform the specific practices in the Process and Product Quality Assurance process area, with the peculiarity that entry-level graduates of a postgraduate program based on GSwE2009 will have theoretical knowledge and practical knowledge for enacting such specific practices, whereas entry-level graduates of an undergraduate program conforming to SE2004 will not be able to reliably perform these specific practices, as their knowledge is confined to theory.

5.6 Global Overview

Table 5.26 summarizes how well the standards cover the different process areas analyzed in the above sections, grouped by maturity level. We have used the color green to indicate process areas for which the educational standards provide sufficient knowledge to carry out all their specific practices. We have used the color orange to indicate process areas for which the standards only provide partial coverage, that is, graduates of programs have some but not all knowledge that they need to enact the respective practices and, therefore, require further training. Readers should consult Section 5.5 to identify exactly which knowledge is missing. For example, readers should consult Table 5.22 to identify which knowledge is missing from SE2004 for the Configuration Management process area. Finally, we have used the color red to indicate process areas for which the standards provide very little or practically no knowledge.

Table 5.26: Coverage of the Process Areas

SE2004	GSwE2009	Process Area Name	Abbr.	Maturity Level
		Configuration Management	CM	2
		Measurement and Analysis	MA	2
		Project Monitoring and Control	PMC	2
		Project Planning	PP	2
		Process and Product Quality Assurance	PPQA	2
		Requirements Management	REQM	2

SE2004	GSwE2009	Process Area Name	Abbr.	Maturity Level
		Supplier Agreement Management	SAM	2
		Decision Analysis and Resolution	DAR	3
		Integrated Project Management	IPM	3
		Organizational Process Definition	OPD	3
		Organizational Process Focus	OPF	3
		Organizational Training	OT	3
		Product Integration	PI	3
		Requirements Development	RD	3
		Risk Management	RSKM	3
		Technical Solution	TS	3
		Validation	VAL	3
		Verification	VER	3
		Organizational Process Performance	OPP	4
		Quantitative Project Management	QPM	4
		Causal Analysis and Resolution	CAR	5
		Organizational Performance Management	OPM	5

5.7 Conclusions

In this paper, we have presented a study of the coverage of CMMI-DEV from the academic viewpoint. Specifically, we have analyzed what knowledge the cores of SE2004 and GSwE2009 provide for graduates to perform specific practices proposed by CMMI-DEV as thoroughly as established by this framework. This study does not aim to analyze whether or not graduates are able to perform a practice, as many aspects, such as previous experience, soft skills or other factors whose analysis is beyond the scope of this paper, influence this process. The aim of this paper is to study how helpful software engineering graduate and undergraduate programs based on the specified educational standards are for training graduates perform the practices involved in an improvement process based on CMMI-DEV. This analysis has found that GSwE2009 typically covers more of the CMMI-specific practices than SE2004, and it does so even on a higher level measured by Bloom’s taxonomy. This is in line with the theoretical expectations that a graduate program should go beyond the undergraduate level, but this study provides some evidences of that.

The results revealed by this study can be useful from both an academic and industrial viewpoint. From the academic viewpoint, we can identify training black spots in our graduates to which we have to pay attention when designing special-purpose programs. From the viewpoint of industry, this study identifies process areas in which their software engineers have received more or less training and can be used to focus special-purpose training programs depending on the organization’s current or targeted maturity level.

It would be worthwhile to replicate this analysis for specific programs about which real information on the in-class time spent on different contents, teaching styles, and optional subjects was available. This would be useful for fine tuning the global analysis conducted in this paper.

5.8 References

URLs are valid as of the publication date of this document.

[Bloom 1969]
Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. *Taxonomy of Educational Objectives: The Classification of Educational Goals.* Longman Group, 1969.

[Career Space 2001]
Career Space. *Curriculum Development Guidelines. New ICT Curricula for the 21st Century Designing Tomorrow's Education.*
http://www.eric.ed.gov/ERICWebPortal/detail?accno=ED459343 (2001)

[CMMI 2010]
CMMI Product Team. *CMMI for Development, Version 1.3* (CMU/SEI-2010-TR-033). Software Engineering Institute, Carnegie Mellon University, 2010. http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm

[Davey 2008]
Davey, B. & Tatnall, A. "Where Will Professional Software Engineering Education Go Next?" 185–192. *Learning to Live in the Knowledge Society: IFIP 20th World Computer Congress*. Milano, Italy, September 2008. Springer, 2008.

[Graduate Software Engineering 2009]
Graduate Software Engineering 2009. Curriculum Guidelines for Graduate Degree Programs in Software Engineering. http://www.gswe2009.org (2009).

[IEEE 2004]
IEEE Computer Society. *Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. A Volume of the Computing Curricula Series*. http://sites.computer.org/ccse/ (2004).

[Kitchenham 2005]
Kitchenham, B., Budgen, D., Brereton, P., & Woodall, P. "An Investigation of Software Engineering Curricula." *Journal of Systems and Software 74*, 3 (February 2005): 325–335.

[Koong 2002]
Koong, K. S., Liu, L. C., & Liu, X. "A Study of the Demand for Information Technology Professionals in Selected Internet Job Portals." *Journal of Information Systems Education 13*, 1 (Spring 2002): 21–28.

[Kovacs 2008]
Kovacs, P. J. & Davis, G. A. "Determining Critical Skills and Knowledge Requirements of IT Professionals by Analyzing Keywords in Job Postings." *Issues in Information Systems 9*, 1 (2008): 95–100.

[Lethbridge 2000]
Lethbridge, T. C. "What knowledge is important to a software professional?" *Computer* 5, 33 (May 2000): 44–50.

[Lethbridge 2007]
Lethbridge, T. C., Díaz-Herrera, J., LeBlanc, R. J., & Thompson, J.B. *Improving Software Practice Through Education: Challenges and Future Trends.* ICSE 2007.Future of Software Engineering Track.

[Morello 2005]
Morello, Diane. *The IT Professional Outlook: Where Will We Go From Here?* http://www.gartner.com/id=485489 (2005).

[Moreno 2012]
Moreno, A., Sanchez-Segura, M-I., Medina-Rodriguez, F. & Carvajal, L. "Balancing Software Engineering Education and Industrial Needs." *Journal of Systems and Software 85*, 7 (February 2012): 1607–1620.

[Prabhakar 2005]
Prabhakar, B., Litecky, C. R., & Arnett, K. "IT skills in a Tough Job Market." *Communications of the ACM 48,* 10 (October 2005): 91-94.

[Surakka 2007]
Surakka, S. "What Subjects and Skills are Important for Software Developers?" *Communications of the ACM 50,* 1 (January 2007): 73–78.

5.9 About the Authors

Dr. Ana M. Moreno is a full professor with the School of Computing at the Universidad Politecnica de Madrid. She has a BS (1994) and a PhD (1997) in Computing. She has been the Director of the MS in Software Engineering since 2001. Her research interests are software engineering education, software usability and agile development. Contact her at ammoreno@fi.upm.es.

Dr. Maria-Isabel Sanchez-Segura has been a faculty member in the Computer Science Department at Carlos III University of Madrid since 1998. Her research interests include usability of interactive systems, software engineering with a focus on processes, methodologies, reuse, management and software configuration management, and recently on intelligent organizations. Maria-Isabel holds a BS in Computing (1997), an MS in Software Engineering (1999), and a PhD in Computing (2001) from the Universidad Politecnica of Madrid.

Dr. Fuensanta Medina-Dominguez holds a BS and a PhD in Computer Science from the Carlos III Technical University of Madrid, Spain. She has been working in the field of software engineering since 2000 and has been a faculty member of the Computer Science Department in the Carlos III Technical University of Madrid since 2004. Her research interests include software engineering and software process improvement focusing on technology transfer in organizations through the use of new trends in computer-supported collaborative work technology.

6 Estimation Competency Development for IT Project Managers: An Infosys Experience

Amit Arun Javadekar, Aman Kumar Singhal, Infosys Technologies Limited

6.1 Abstract

This paper describes some high-performance practices implemented at Infosys Limited to significantly improve the estimation competency of various roles involved in project execution and management, sales, and quality assurance functions. These practices will cover the specifics of role-based workshops and in-house certification programs, which have triggered innovation for development of new estimation models, developing estimation ecosystems, improvement of overall service capability, and above all, improvements in large-scale change management. This was done in the context of accelerated growth, diverse talents, and need for global reach and scalability.

6.2 Context

Estimation methods in the information technology (IT) industry are not as well developed as in traditional industries, such as manufacturing or construction. Estimation practices are still evolving for various types of IT projects and services, and standardization across the industry is seen in only a few select areas. In today's challenging business scenarios, it is important to have a high degree of estimation maturity to ensure competitive proposals, manage costs, and above all execute projects to provide measureable value to clients. There has also been an increasing demand from clients to move towards standardized estimation methods.

Infosys is a large IT services company with over 140,000 employees spread over 75 cities across the globe and executes thousands of projects at a given point in time in various business and technology domains. To meet the challenges mentioned above, improve predictability, and reduce risk in client delivery, it was important to comprehensively address and enable key roles on estimation across the project management, sales/pre-sales, and quality assurance domains. Infosys has strategically invested in the focused development of estimation competency through its Estimation Center of Excellence (called ESTEEM).

ESTEEM has been the driving force behind the development of in-house training workshops and certifications intellectual property creation by developing new estimation models, the creation of estimation tools, process capability improvement, development of an estimation ecosystem, and above all, large-scale change management across the organization.

6.3 Key Experiences

Below are the key high performance practices that have helped us strengthen estimation competency for IT project managers in a large organization like Infosys.

6.3.1 Customized In-House Enabling and Certification Programs

Infosys has a well-defined competency development framework. The competency dimensions identified are technology, business domain, behavioral, and process and project management. Estimation has been identified as a key area to focus on as part of the project management competency dimension. Certain key roles and formal job descriptions were identified for estimation competency improvement. This ensured that the development of estimation enabling and certification program is in line with the organizational expectations from the target roles.

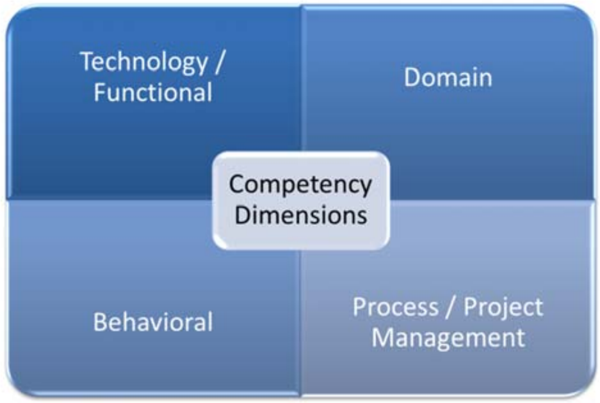

Figure 6.1: Dimensions of Competency Development

The customized training and certification programs on estimation have been developed using a structured approach, namely Strategize, Develop, Deploy, and Measure as below.

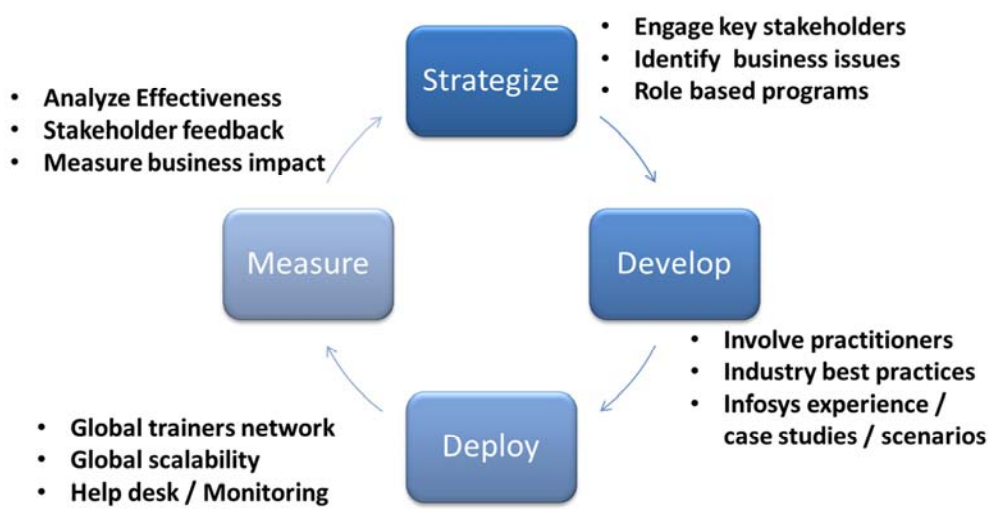

Figure 6.2: Approach for Competency Development

The Strategize phase involves engaging with key stakeholders like business units, clients, and the sales team to understand the business needs and related estimation issues in terms of predictability, financials, etc. Analysis of the current business needs is carried out to create plans to drive estimation competency improvement. The Develop phase involves developing competency improvement programs in line with the direction and the plan charted out. It includes the involvement of practitioners from various business units, leveraging industry best practices and Infosys experience to design relevant case studies and scenarios. The Deploy phase ensures that the necessary plan and resources are available to implement the program. These include people (like the trainers' pool), systems, and infrastructure required to achieve global scalability (anytime, anywhere access to the program). Regular monitoring and reporting ensures smooth implementation of the program. The Measure phase is focused on analytics and feedback mechanisms to make sure that there is measurable impact on business and a feedback loop is provided to the planning cycle.

The program development is based on a few guiding principles, such as

- customization for specific roles (estimator or reviewer)
- comprehensive coverage of various estimation aspects (size, effort, schedule, cost)
- consideration of service offerings and related estimation issues
- practice oriented (real project scenarios, case studies, practice tests)

A sample list of enabling and certification programs that form part of estimation competency development are given in Table 6.1. The certification program is closely linked with the employee performance management and is also treated as eligibility criteria for higher roles.

Table 6.1: Sample Enabling and Certification Program

Enabling / Certification Program	Remarks
Estimation Specialist	Online exam - 2 papers with case study Comprehensive enabling 40-60 hours
Estimation Foundation for Client Services	12 hours workshop with online exam
Estimation @ Proposal Stage	Focus on early life cycle estimation, comprehensive coverage of estimation methods / life cycle

Over 15,000 Infosys employees (see Figure 6.3) have gone through the role-based enabling and certification programs over the last four to five years. These trained and certified employees have played a major role in developing knowledge clusters across the globe to support business on estimation-related matters.

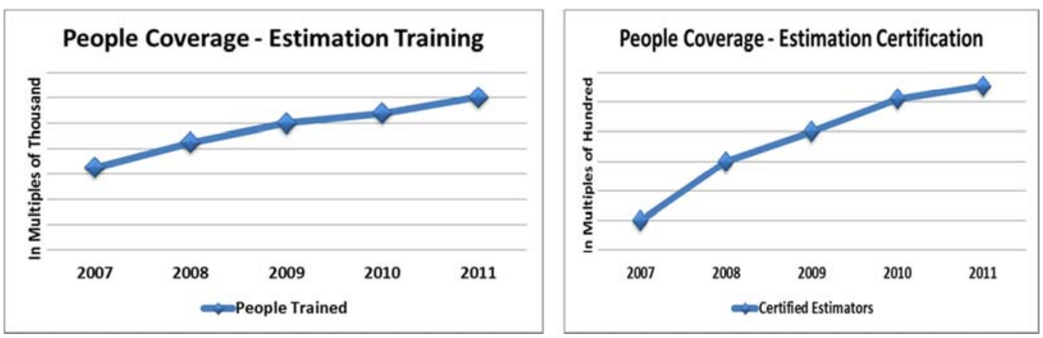

Figure 6.3: Estimation of Competency Development: People Coverage

6.3.2 Focus on Standardization through Innovation and Collaborative Research

In order to standardize estimation methods and improve process capability for various service lines (such as development, maintenance, testing, or package implementation) at Infosys, it was necessary to innovate new estimation methods where there are no standards available. This was done in collaboration with business units and also led to competency development for people involved in these research projects. These research projects, along with enabling programs, have been a breeding ground for innovative ideas to develop estimation models in select areas. This has provided a good learning opportunity for the research team, which is drawn from various units on a voluntary basis.

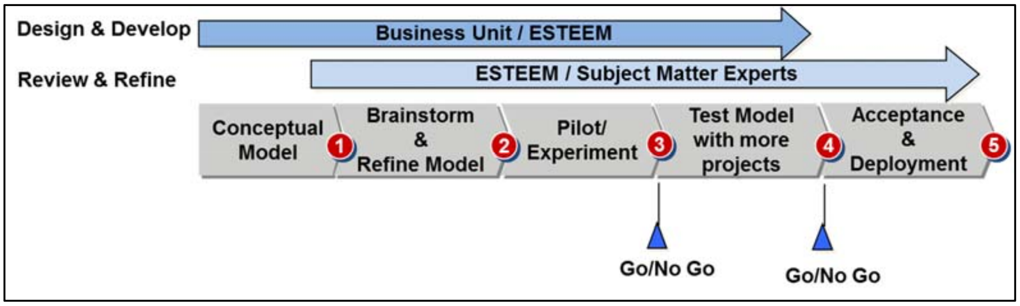

Figure 6.4: Estimation Method Development and Standardization

The estimation model development projects follow a lifecycle as shown above. Based on the analysis of various pilot results, go and no-go decisions are made with respect to acceptance and deployment. In the last few years, there has been a great deal of focused effort and investment on standardization, and various research projects were completed in areas such as testing, corrective maintenance, package implementation, data warehouse, and early lifecycle models, among others. These Infosys IPs have led to standardization of estimation models for most key service lines. These models leveraged Infosys project experience and historical data with a focus on the concept of size.

6.3.2.1 Estimation Ecosystem—Supporting Competency Development

To achieve the purpose of application of knowledge and development of in-depth estimation skills, the organization needed a robust estimation ecosystem. It helps share knowledge, provides help on the ground, and ensures estimation effectiveness to achieve desired business results.

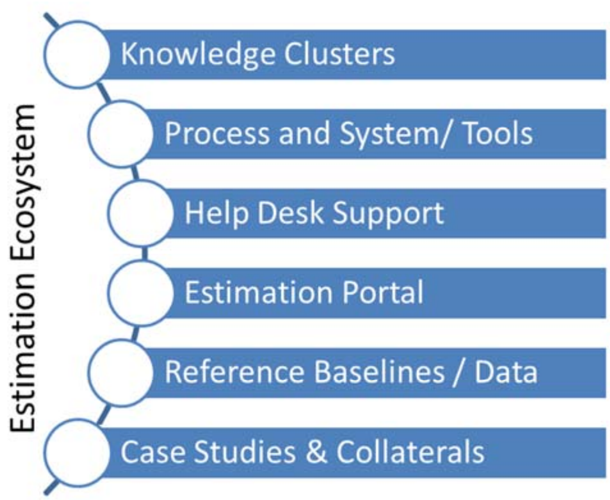

Figure 6.5: Estimation Ecosystem

The ecosystem has been focused on

- creating knowledge clusters and subject matter experts across the globe
- integrating process, systems, and tools to ensure the efficiency and effectiveness of estimation process
- creating an estimation portal, help desk, baselines, and case studies to facilitate project estimation and reviews

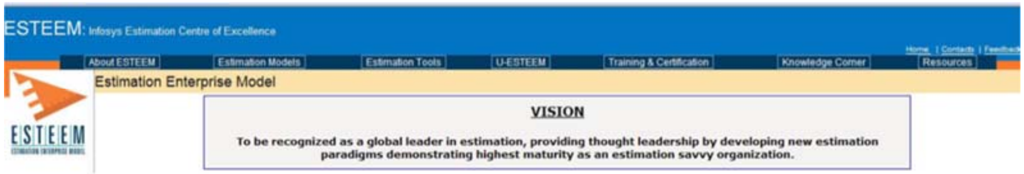

Figure 6.6: Estimation Portal

The availability of the estimation ecosystem has made it possible to achieve world-wide scalability and global deployment of the estimation process.

6.3.2.2 Managing change—estimation competency development program

Since Infosys is a large IT service organization with over 140,000 people who work across several business lines, it was a Herculean task to drive the competency development program on estimation. The critical success factors for this change program included senior management sponsorship and reviews, the formation of a unit level estimation council, linkages to goals and business impact, and integration with the Infosys competency framework.

The estimation council at unit level (U-ESTEEM) has provided essential focus and effort to accelerate change for the respective unit. U-ESTEEM received the sponsorship of unit leadership and participation of estimation champions at the unit level. They leveraged the business and estimation expertise available within the unit and also promoted the development of estimation competency at the unit level. The corporate ESTEEM group provided help to U-ESTEEM so that they could leverage corporate programs and ecosystems for estimation competency development. Infosys's senior leadership and business unit leadership periodically reviews the progress and outcomes.

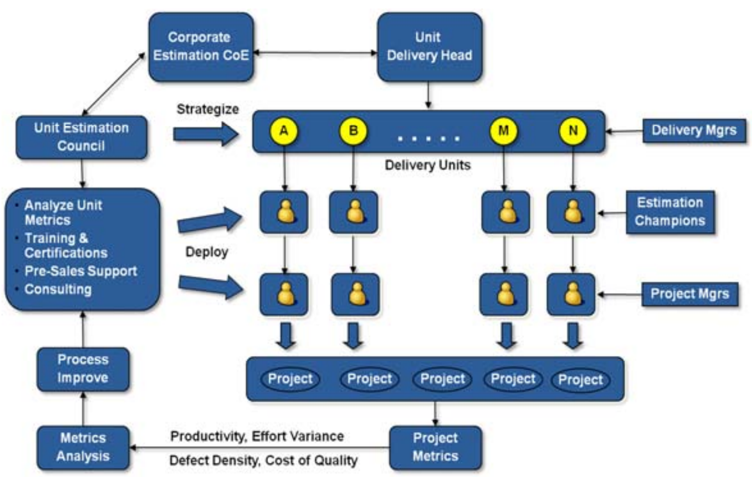

Figure 6.7: Unit Level Estimation Council

The goal-setting exercise also helped in driving the need for competency development on estimation. Goals for estimation accuracy and productivity improvement were based on the standardized sizing and estimation methods. These goals are evaluated as part of the performance management process for relevant roles.

Figure 6.8: Goal Flow Down

The goal-setting process also triggers senior management reviews to ensure that correct and accurate measurements and strategies are being adopted to achieve the goals. Also, a software quality advisor (SQA) gets associated with each project and as part of SQA service; he or she also reviews the measurements and their effectiveness.

6.4 Conclusion

Infosys differentiates itself in the market through its best-in-class execution capability that brings predictability, delivers value, and reduces risk for our client projects. The estimation competency development program has been successfully delivering the objectives in terms of improving estimation capabilities. For example, this is reflected in terms of current effort and schedule estimation accuracy of greater than 90 percent of development projects within a 10 percent deviation. There has been a significant improvement (approximately 15 percent) in estimation accuracy over last few years (See Figure 6.9). This program has been a driving force in promoting scientific estimation culture and developing an estimation-savvy team at Infosys. This has also led to higher client confidence and improved client experience.

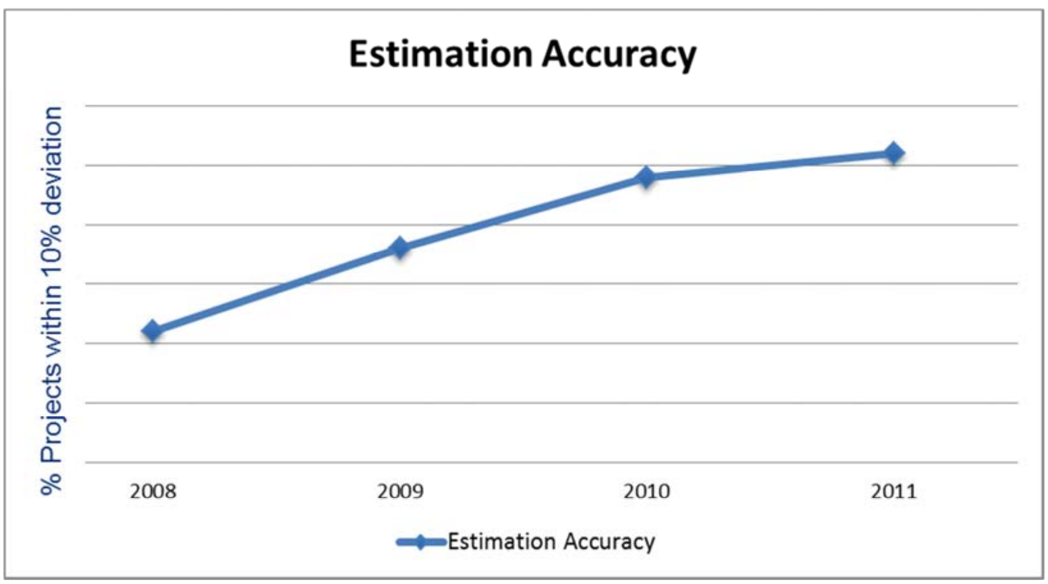

Figure 6.9: Estimation Accuracy Improvement

7 Enhancing Process Asset Assessment

Maria-Isabel Sanchez-Segura, Computer Science Department, Carlos III University of Madrid, Spain, misanche@inf.uc3m.es

Alejandro Ruiz-Robles, Information Systems Department, University of Piura, Peru, alejandro.ruiz@udep.pe

Arturo Mora-Soto, Computer Science Department, Carlos III University of Madrid, Spain, jmora@inf.uc3m.es

Javier Garcia-Guzman, Computer Science Department, Carlos III University of Madrid, Spain, jgarciag@inf.uc3m.es

7.1 Abstract

Process assets have proven to be useful to software engineering companies. However, determining the value of a process asset to an organization is still a critical open question.

The approach presented here focuses on how to perform process assets assessments and is based on two main facts. First, process assets are mainly intangible, knowledge-based assets, and their management and assessment must, therefore, be based on experience from disciplines such as knowledge management and intellectual capital. Second, process assets represent investments that are expected to add business value. From the strategic management perspective, this value must be determined by aligning process assets with business goals and assessing how these assets contribute to the achievement of these goals.

7.2 Introduction

Both practitioners and academics are aware that processes are crucial to business success, and their benefits have been widely empirically observed. Process assets are intangibles that relate to, describe, implement, and improve processes. These process assets are developed or acquired by organizations in order to meet their business goals [CMMI 2010].

Although process assets represent investments that are expected to add business value, traditional process model assessments are not endowed with enough mechanisms to highlight this value for a company [April 2009]. There are two main benefits of determining the value of process assets for a company: first, know whether the company's investments are paying off, and second, understand what each process asset contributes to the achievement of company business goals, and make decisions on how to improve the process assets that are not helping to achieve business goals.

This raises the question of how to enhance process assets assessment. In order to assess how valuable a process asset is to a software company, we first need to analyze process assets with the understanding that they are intangible assets. They then have to be assessed from the viewpoint of disciplines concerned with measuring or valuing such assets in terms of something that is significant for the company, like the achievement of the organization's business goals.

The importance of assessing process assets in order to improve company software processes has already been highlighted and demonstrated [Albuquerque 2009]. However, process assets have not yet been considered as investments that should be aligned with business goals. Likewise, the need to determine the value of company intangible assets has also been studied

[Qian 2010]. However, in view of their characteristics, this requirement should also to be extended to software process assets.

The approach presented by Basili [Basili 2010] explicitly links goals at different levels, from business objectives to project operations in software companies, which is critical to strategic measurement. Our proposal is oriented to demonstrate the link between business objectives and process assets, which is a step towards the assessment of process assets based on its valuation.

The remainder of this paper is structured as follows. Section 7.3 explains the nature of process assets and provides the groundwork for this proposal. Section 7.4 describes the proposal along with an application example. Section 7.5 discusses the implications of our work and our conclusions.

7.3 Identifying Process Asset Characteristics

A process asset is usually thought of as, for example, an electronic process guide that explains how to perform a requirement elicitation interview or a lessons learned document that summarizes experiences from the last project. We take the view that a process asset is more complex and should be construed from three different perspectives.

From the knowledge management perspective, process assets are knowledge-based assets that represent the organizational knowledge related to process description, implementation, and improvement. Depending on the type of knowledge it contains, a knowledge-based asset can be explicit, implicit, or tacit [Nonaka 1991, Davenport 2000]. Explicit knowledge is knowledge that has been articulated, codified, and communicated in symbolic form or natural language, such as an electronic process guide. Implicit knowledge is what people know from experience. Implicit knowledge can be specified: that is, it can be represented as explicit knowledge to be conveyed to other people. Finally, tacit knowledge is based on an action, experience, and involvement in a specific context, and it can only be transferred from one person to another through interaction, given that it cannot be formalized as explicit knowledge [Nonaka 1991, Alavi 2001, Nickols 2000].

If we consider process assets to be knowledge-based assets, these assets should be assessed according to the three types of knowledge; otherwise, the result of the assessment could be unsatisfactory because it would only be considering a subset of process assets.

Viewing a software company and its process assets from the intellectual capital perspective, process assets are part of the company's intellectual capital. Intellectual capital is one of a company's three main vital resources [Petty 2000, Stewart 1998, Brooking 1996, Marr 2008], and includes all non-tangible resources that contribute to the delivery of the organization's proposition value. The other two main resources are physical capital, like computers or buildings, and financial capital. It is important not to misunderstand the nature of process assets—they are non-tangible resources of the company, which means that they have no physical substance. However, they can be represented in a printed format, saved in a digital document, or take part in the balance sheets of the company. As process assets are intangible assets, we suggest that they should be viewed as a company's intellectual capital, and the experience of the intellectual capital field should be taken into account in order to assess and determine the value that they add to a company.

Considering that process assets are investments that are expected to provide business value, this value must be determined from the strategic management perspective by aligning process assets with business goals and determining how these assets contribute to the achievement of business goals.

In short, process assets are intangible and knowledge-based assets, and thus the value that a process asset adds to a company should be determined by assessing the intellectual capital that it represents and how it contributes to the achievement of business goals, taking into account the type of knowledge (explicit, implicit, or tacit) embedded in the process asset.

7.4 Assessing and Valuing Process Assets as Intangible Assets, Knowledge-Based Assets, and Investments

Although the use of process assets is considered beneficial in software companies, current process model assessments focus more on the existence of process assets than on their value for the company [Scacchi 2002, Baddoo 2003, von Wangenheim 2010, CMMI 2010]. We are convinced that knowledge maturity models aligned with intellectual capital models, which are the best tools for managing and assessing process assets, should be added to existing process improvement maturity models.

This proposal has been developed by taking the above into account and takes a step towards the view that combines intellectual capital, strategic management, and process improvement, and also accounts for the different dimensions of knowledge.

The goal of this proposal is to help software companies determine the value of their process assets by determining how process assets contribute to the achievement of their business goals. By estimating the value of process assets, a company will be able to decide whether its investments are paying off and understand the extent to which it is achieving its business goals thanks to the process assets. Consequently, the company will be able to find out where and how it can improve.

7.4.1 Outlining the Alignment of Process Assets and Business Goals

The backbone of this proposal is the alignment of process assets and business goals. However, there is no direct alignment. Process assets and business goals are linked through the software company's organizational processes (see Figure 7.1). A process asset helps to describe, implement, and improve organizational processes, and such processes are intended to meet the company's business goals.

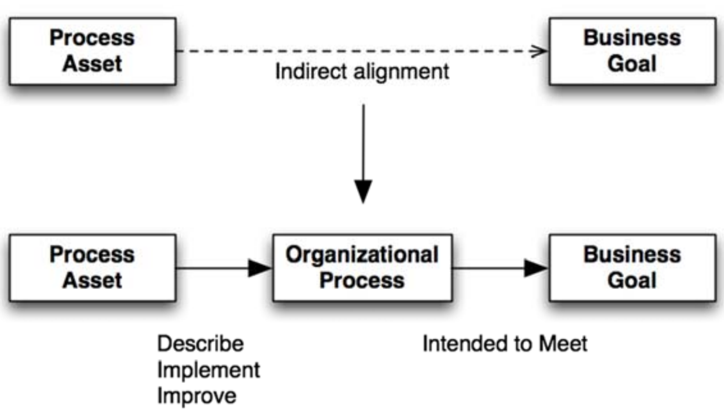

Figure 7.1: The Relationship Between Process Assets and Business Goals

This method is aimed to be used by any software company, no matter its process maturity level or whether it has a process culture instituted. Since every company has business goals to meet and process assets, even if it has no formal processes, the aim of this proposal is to allow a company to assess its process assets even when it is in its very beginnings, without requiring formal information about its processes behavior.

In order to guarantee that process assets are properly linked to business goals, we propose the use of two elements called key performance questions and performance indicators, which are shown in Figure 7.2. These elements were borrowed from research by Marr [Marr 2008]. These must be developed by software companies and linked with their process assets and business goals to align process assets with business goals. The next section introduces these elements and a method for their development.

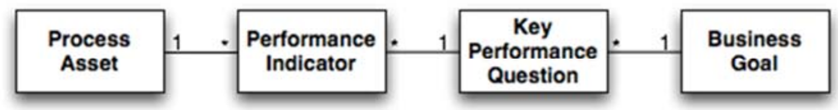

Figure 7.2: Alignment of Process Assets and Business Goals

7.4.2 Determining the Value of Process Assets

There follows a five-step method to guide software companies through the process of aligning their process assets with their business goals and determining the value of their process assets (Figure 7.3). This method involves identifying a software company's process assets and business goals, developing the elements called key performance questions and performance indicators, and, by using the previously defined key performance questions and performance indicators, aligning process assets and business goals. An applied example is presented alongside the method.

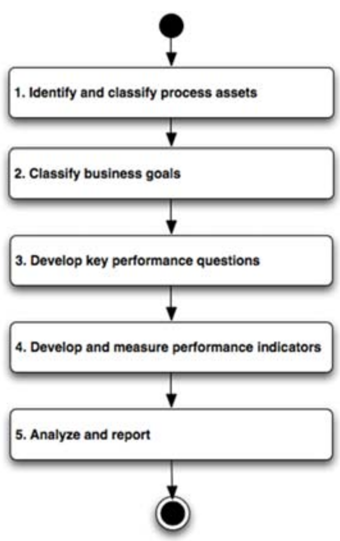

Figure 7.3: A Five-Step Method to Assess Process Assets

7.4.2.1 Step 1: Identify and Classify Process Assets

The first step is for software companies to identify and classify their process assets.

Process assets should be understood as the assets that describe, implement, and improve processes. In our opinion, as explained above, a process asset cannot be confined to the classical view that it is a document that describes how to perform a process. From a broader perspective, for instance, people's knowledge or experience should be considered as process assets because they are important for describing, implementing, or improving processes.

The process assets taxonomy presented below is the groundwork for this purpose. This taxonomy is based on different intellectual capital models [Edvinsson 1997, IADE 2003, Marr 2008] and has been developed and adapted to the reality of software companies. The kinds of intangible assets presented in the taxonomy have been considered as process assets because they can help to describe, implement, or improve processes within organizations.

The taxonomy (see Table 7.1 below) includes nine types of process assets divided into three main categories: structural process assets, human process assets, and relational process assets. These three main categories are equivalent to the main types of intellectual capital: structural, human, and relational capital. Once assets have been classed into one of the categories of the taxonomy in Table 7.1, organizations know which assets are affecting which type of intellectual capital. Note that this is not a closed taxonomy, and either its breadth or depth could be adapted to the particular needs of a company. At the same time that each process asset is catalogued, we have to identify which types of knowledge (explicit, implicit, or tacit) are embedded in each process asset.

Table 7.1: Proposed Process Assets Taxonomy

Structural process assets represent the process assets that belong to and are held by the company.	
	Knowledge documents. This category represents any kind of organizational knowledge captured in a document, either in paper or in digital format. This category includes, for instance, electronic process guides or lessons learned documents.
	Tools. This category represents any kind of technological tool used to manage any type of process asset. This category includes, for instance, a database or repository of knowledge documents or an intranet to share and consolidate experiences in order to improve processes.
	Organizational structure. This category represents how the company is organized as a whole, how the teams are configured around the projects, and how the teams are configured around the specific activities that they perform. This category includes, for instance, the organizational working policies, such as the way that two different project teams coordinate activities or consolidate experiences.
	Knowledge management culture. This category represents how the company manages its knowledge, that is, how company knowledge is developed, delivered, and used. This category includes, for instance, the processes of eliciting knowledge from experienced people or motivating people to learn and apply knowledge related to new development processes.
Human process assets represent the living and thinking part of a company's process assets; the main difference between human and structural assets is that a company loses intangible human assets when people leave.	
	Knowledge of people. This category represents people's knowledge related to the tasks they perform and any of the structural and relational process assets.
	Experience. This category represents people's experience related to task performance and the creation or use of any of the structural and relational process assets.
	Competencies and skills. This category represents the competencies and skills that people need to perform their tasks and to create or use any of the structural and relational process assets. This category includes, for instance, the self-learning capability needed to adopt a new technology or the communication skills that people need to transmit their experience.
Relational process assets represent the relationships between the organization and any outside person or organization.	
	Relationship with customers and users. This category represents the formal or informal relationships with customers and users. This category includes, for instance, the processes used to communicate with users or informal meetings held with clients.
	Relationship with suppliers. This category represents the formal or informal relationships with suppliers. This category includes the processes for ordering services from a supplier or informal channels used to improve communication with suppliers.

The company must try to identify and classify its process assets based on this taxonomy. It is important to emphasize that the use of this taxonomy facilitates discussion among the members of the organization in order to describe and understand as many process assets as possible. In practice, however, a company does not need to identify a priori all its process assets for assessment: they could be identified and classified as per the needs of the organization, as shown in the application example, or even as the assets are created and used.

7.4.2.2 Application Example of Step 1

The application example illustrated below is an extract of a real situation taken from a small Peruvian enterprise. For reasons of space, not all the information can be presented, and the company must remain anonymous due to a data privacy agreement. Some details are specified, however, into order to contextualize the example.

The company develops three different software products, called A, B, and C, that it sells and supports to private schools around Peru. Due to fiercer competition in recent years, the company has started to expand into the state schools market, and has tried to consolidate its position with its regular customers. A team with members from different company areas was set up in order to assess its process assets.

The team started by identifying and classifying the process assets using the process assets taxonomy proposed as a guide. Some of the assets identified and then assessed are shown in Table 7.2.

Table 7.2: Assets Identified and then Assessed

Type of Process Asset		Process Asset
Structural process assets		
	Knowledge documents	Life-cycle model documents. These are a set of documents used to describe how each development process activity must be performed. The three products share the same life-cycle model. Type of knowledge: explicit
	Tools	Knowledge repository. This is a wiki used to share knowledge and documents related to the processes used in the organization. Type of knowledge: explicit
	Knowledge management culture	Knowledge-transfer processes. These are a set of documents used to describe the formal processes that employees must enact in order to share their knowledge and to request knowledge that they need to perform their activities. Type of knowledge: explicit
Human process assets		
	Knowledge of people	Knowledge of new recruits. This asset is related to how much new recruits know about how to perform the type of activities required in their jobs. Type of knowledge: implicit and tacit
	Experience	Employees' experience. This asset is related to how experienced employees are at performing the activities required in their jobs. Type of knowledge: implicit and tacit
Relational process assets		
	Relationship with customers and users	Informal meetings with users. These are short meetings that are held on a regular basis when a member of the company visits a customer for a formal meeting. At informal meetings, users state requirements or highlight what they consider to have good functionality. Type of knowledge: implicit and tacit

7.4.2.3 Step 2: Classify Business Goals

The second step is to classify the organization's business goals.

Business goals must be classified within the categories of the taxonomy shown in Figure 7.4 below. This taxonomy has been taken from Clements and Bass [Clements 2010] and is general enough to cover any kind of business goal. If a company's business goals are not formally defined, the taxonomy could be used as a tool for defining or eliciting such goals.

Growth and continuity of the organization
Meeting financial objectives
Meeting personal objectives
Meeting responsibility to employees
Meeting responsibility to society
Meeting responsibility to country
Meeting responsibility to shareholders
Managing market position
Improving business processes
Managing quality and reputation of products

Figure 7.4: Business Goals Taxonomy

Application example of Step 2

The company elicited and classified the following two business goals according to the business goals taxonomy:

Expand the use of product B to the state schools market. This goal was classified in the proposed business goals taxonomy within the business goals categories of "Growth and continuity of the organization" and "Meeting financial objectives."

Improve support processes to strengthen market position. This goal was classified in the proposed business goals taxonomy within the business goals category of "Managing market position."

7.4.2.4 Step 3: Develop Key Performance Questions

The third step is to develop the company's key performance questions (KPQs).

Key performance questions set out what a company wants to know about its process assets with respect to its business goals and capture the indirect relationship between process assets and business goals through organizational processes.

Before a company develops its KPQs, it must decide which process assets it wants to assess in respect to which business goals. It must then state the KPQs considering the three rules defined below:

1. A KPQ can be associated with one or more process assets, but can target only one business goal. A process asset can be associated with several business goals through different KPQs.
2. A KPQ must be stated as an open question. A simple yes or no should not be sufficient to answer the KPQ.

3. A KPQ must link process assets with organizational processes that are expected to contribute to one or more business goals. This means that a process asset is linked to business goals not directly, but indirectly, through its contribution to organizational processes.

Companies may find the above three-rule structure helpful for formulating a KPQ as follows: (How well | To what extent) does a process asset (help | support | contribute to) the (description | implementation | improvement) of an organizational process? The organizational process should target a business goal, and for this purpose, the company must at least identify the processes it performs. A short comment justifying why the question was formulated, (its rationale should also be added).

Application Example of Step 3

The company defined the following two key performance questions in order to assess how some of its process assets were contributing to the achievement of the above business goals. An example of this process is shown in Table 7.3 and Table 7.4 below.

Table 7.3: First Key Performance Question

Process Asset	Key Performance Question	Business Goal	Rationale
Knowledge repository Life-cycle models documents Knowledge transfer processes Employees' experience Knowledge of the new recruits	How useful are the knowledge repository and employee experience for speeding up the adoption of development processes by the company's new recruits?	Expand the use of product B to the state schools market.	This question was formulated because product B had to be adapted for use in the new market segment, and a new team composed partly of new recruits was set up to for this purpose.

Table 7.4: Second Key Performance Question

Process Asset	Key Performance Question	Business Goal	Rationale
Lifecycle models documents Informal meetings with users	To what extent do informal meetings with users offset the shortcomings of the requirements elicitation process?	Improve support processes to strengthen market position.	This question was designed because informal meetings are held frequently. This could mean that the formal requirements elicitation process needs to be improved or that informal meetings should somehow be integrated with the formal processes.

7.4.2.5 Step 4: Develop and Measure Performance Indicators

The fourth step is to define and measure the performance indicators.

Performance indicators measure particular aspects, characteristics, or properties of process assets in order to answer key performance questions (KPQs). One or more performance indicators must be developed and measured for each pair of process asset and key performance questions. Note that a performance indicator of the process asset associated with a KPQ could be useless if it is applied to another KPQ.

A performance indicator must include the following information and must be developed by taking into account the type of knowledge (explicit, implicit or tacit) that process assets contain:

- **Name** is the description of what the performance indicator is to measure.
- **Value** is the possible values or value range for a performance indicator. It defines what values can be assigned to a performance indicator, such as low, medium, or high.
- **Mechanism** describes the mechanism that will be used to collect the information. It defines what mechanism would be used to collect the necessary data to define the value of the indicator, such as interviews, surveys, or document analysis.
- **Source** describes the source of the information. It defines the source of the data to be collected, such as company employees or the documents database.
- **Frequency** refers to the frequency with which each measurement is taken. It defines the period of time between each measurement.

By defining performance indicators, a company will have established a link between its process assets and its business goals. Measurement of the performance indicators must then start in order to answer the key performance questions and assess if or how the process assets are contributing to the achievement of business goals, as well as the value of the process assets.

If applicable, extra information apart from the value of a performance indicator could be requested about why a particular value has been assigned to the performance indicator. This extra information could lead to a further study of a particular process asset.

Application Example of Step 4

The company defined the following performance indicators to answer the key performance questions defined in step 3. The result of this process is shown in Table 7.5 below.

Table 7.5: Performance Indicators Definition

How useful is the knowledge repository and employee experience for speeding up the adoption of development processes by the company's new recruits?					
Process asset	**Performance Indicator**	**Value**	**Mechanism**	**Source**	**Frequency**
Knowledge repository (wiki)	Usability level	High Normal Poor	Online survey mediated by the knowledge repository	New recruits' opinion	Monthly
	Search engine precision	Good Poor	Online survey mediated by the knowledge repository	New recruits' opinion	Monthly
Life-cycle model documents	Documents learnability	Simple Normal Complicated	Online survey mediated by the knowledge repository	New recruits' opinion	For each accessed document
	Applicability in real activities	Applicable Non applicable	Online survey mediated by the knowledge repository	New recruits' opinion	For each accessed document
Knowledge transfer processes	Process effectiveness	High Low	Online survey	Employees' opinion	Every two weeks
	Extra workload	Acceptable	Online survey	Employees'	Every two

		Too much		opinion	weeks
	Support received by the new recruits from other employees	Useful Acceptable Poor	Anonymous surveys	Employees	Every three months during the project
Knowledge of new recruits	Knowledge related to the company development processes	High Medium Low	Personal interviews and document inspection	Employees and their resumes	Once at the beginning of the project
Employees' experience	Experience related to company development processes	<= 1 year > 1 <= 3 years > 3 years	Personal interviews and document inspection	Employees and their company project participation history	Once at the beginning of the project
To what extent do informal meetings with users offset the shortcomings of the requirements elicitation process?					
Process asset	**Performance indicator**	**Values**	**Mechanism**	**Source**	**Frequency**
Life-cycle model documents	Applicability of the formal requirements elicitation process	Very applicable Sometimes Not very applicable	Online survey	Employees' opinion	Every two months
Informal meetings with users	Relevance of information captured in informal meetings against information captured in formal processes	Much more relevant More relevant Equally relevant Less relevant	Online survey	Employees' opinion	Every two months

7.4.2.6 Step 5: Analyze and report

The fifth and last step is to analyze the information obtained after the assessment of the process assets under consideration and to report the results to the areas or people that asked for the assessment.

The analysis must conclude whether or not, and, if so, how a process asset is contributing to the achievement of the preselected business goals and, therefore, how valuable the process asset is. The company could then determine if its investments are paying off, understand to how process assets are helping it to achieve its business goals and, therefore, know where and how the company can improve further.

Finally, note that the assessment can be improved, for instance, by further specifying the key performance questions or improving the accuracy of the possible values of a performance indicator.

Application example of Step 5

The company assessed the *process assets* shown in Table 7.6 below. Using our proposed method, not only can the company determine the value of a process asset using the performance indicators, but it can also identify which process assets need to be improved in order to achieve a business goal by clarifying and specifying the link between process assets and business goals.

The company’s findings after analyzing this information are as follows:

1. The knowledge repository, lifecycle model documents, employee experience, and knowledge of new recruits were found to be valuable for speeding up the adoption of development processes by the new recruits in order to expand the use of Product B to the state schools market, and therefore contribute to the growth and survival of the organization and also help meet its financial objectives.
2. Although the knowledge repository, lifecycle model documents, employee experience, and knowledge of new recruits were good enough to speed up the adoption of processes by new recruits, the company should improve knowledge-transfer processes to avoid undermining the value of the other four assets. Support received by new recruits from other employees was merely acceptable because the knowledge transfer processes require an extra workload.
3. The formal requirements elicitation process was not formal, but informal meetings were a valuable asset for strengthening market position. Informal meetings with users could provide some insights about how to improve the formal requirements elicitation process.
4. Another of the things extracted from the results analysis is that the knowledge repository (wiki), lifecycle model documents, and knowledge transfer processes are part of the company’s structural capital; the knowledge of new recruits and employee experience are part of the company’s human capital; and the information meetings with users are part of the company’s relational capital. This is valuable information for auditing the company’s intellectual capital.

Business Goal Category	Business Goal	Key Performance Question	Process Assets Category		Process Asset	Performance Indicator	Value Obtained
Organizational growth and survival Meeting financial objectives	Expand the use of product B to the state schools market.	How useful is the knowledge repository and employee experience for speeding up the adoption of development processes by the company's new recruits?	Structural process assets	Tools	Knowledge repository (wiki)	Usability level	High
						Search engine precision	Good
			Structural process assets	Knowledge documents	Life-cycle models documents	Document learnability	Simple
						Applicability in real activities	Applicable
			Structural process assets	Knowledge management culture	Knowledge transfer processes	Process effectiveness	High
						Extra workload	Too much
						Support received by the new recruits from the other employees	Acceptable
			Human process assets	People's knowledge	Knowledge of new recruits	Knowledge related to company development processes	Medium
			Human process assets	Experience	Employees' experience	Experience related to company development processes	> 3 years
Managing market position	Improve support processes to strengthen market position.	To what extent do informal meetings with users offset the shortcomings of the requirements elicitation process?	Structural process assets	Knowledge documents	Life-cycle models documents	Applicability of formal requirements capture processes	High applicable
			Relational process assets	Relationship with clients and users	Informal meetings with users	Relevance of information captured with informal meetings compared with information captured with formal processes	More relevant

Table 7.6: Summary of the Process Asset Assessment

7.5 Conclusions and Future Work

This proposal classifies process assets according to intellectual capital types for the purpose of relating the software process improvement area to the intellectual capital field. The aim is to transfer the concept of intangible assets assessment from the intellectual capital field to the assessment of process assets viewed as organizational intangible assets.

To assess process assets, we propose a mechanism for identifying indicators whose value can specify the importance of process assets. These indicators, called performance indicators, can adapt general-purpose intellectual capital models for the software engineering context.

Performance indicators are defined for each process asset and answer questions, called key performance questions, which have been described in terms of particular business goals.

One of the major benefits of this process assets assessment proposal is its visibility, as the process assets can be traced to the company business goals through a series of indicators specifying not only the existence but also the status of such process assets. In this way, companies can make decisions about

- Which assets help to satisfy which business goals and to what extent?
- Which assets are not contributing as much as they should to the company, and are, therefore, superfluous?
- Which assets are not achieving the expected value and, therefore, need to be improved to attain the business value associated with the respective process asset?
- Which process assets add value to which type of intellectual capital, as the assets are catalogued based on the main types of intellectual capital: structure, human and relational capital?
- Which process assets, even if positioned in different branches of the proposed process assets taxonomy and apparently unrelated, share the same key performance questions?

This general-purpose approach has been designed to be applicable to any software company. If, however, a company has specific business goals not listed in this proposal, it could be customized, and we intend to detail the tailoring steps as the next step for improving this proposal.

Because of our goal of allowing any company to assess its process assets regardless of its process maturity level, this proposal cannot determine the value of process assets taking into account process or process improvement goals or metrics. The next step for improving this proposal is to take into account the process maturity level of companies, classifying performance indicators according to the process maturity levels [CMMI 2010], and complement them with process and process improvement metrics to determine the value of process assets.

Besides the improvement of performance indicators, the authors are developing a decision-making artifact. After performing the assessment, this artifact is intended to guide companies in what steps they should follow in respect to their process assets.

7.6 Acknowledgements

This work has been partially funded by the Spanish Ministry of Science and Technology through the project TIN2011-27244.

7.7 References

URLs are valid as of the publication date of this document.

[Alavi 2001]
Alavi, Maryam & Leidner, Dorothy E. "Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues." *MIS Quarterly 25*, 1 (March 2001): 107–136.

[Albuquerque 2009]
Albuquerque, Adriano Bessa & Rocha, Ana Regina. "Evaluation and Improvement of Processes Assets: A Real Collaborative Experience," 114-120. *Proceedings of the World Congress on Software Engineering*. Xiamen, China, May 2009. El Compendex, 2009.

[April 2009]
April, Alain & Laporte, Claude. "An Overview of Software Engineering Process and Its Improvement." *Encyclopedia of Information Science and Technology*. ISI Global, 2009.

[Baddoo 2003]
Baddoo, N. "De-Motivators for Software Process Improvement: An Analysis of Practitioners' Views." *Journal of Systems and Software 66*, 1 (April 2003): 23–33.

[Basili 2010]
Basili, Victor R., Lindvall, Mikael, Regardie, Myrna, Seaman, Carolyn, Heidrich, Jens, Münch, Jürgen, Rombach, Dieter, & Trendowicz, Adam. "Linking Software Development and Business Strategy through Measurement." *IEEE Computer 43*, 4 (April 2010): 57–65.

[Brooking 1996]
Brooking, Annie. *Intellectual Capital: Core Asset for the Third Millennium*. Thomson Learning, 1996.

[Clements 2010]
Clements, Paul & Bass, Len. "The Business Goals Viewpoint." *IEEE Software 59*, Supplement 1 (December 2010): S19–23.

[CMMI 2010]
CMMI Product Team. *CMMI for Development, Version 1.3* (CMU/SEI-2010-TR-033). Software Engineering Institute, Carnegie Mellon University, 2010.
http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm

[Davenport 2000]
Davenport, Thomas H. & Prusak, Lawrence. *Working Knowledge: How Organizations Manage What They Know*. Harvard Business Press, 2000.

[Edvinsson 1997]
Edvinsson, Leif. “Developing intellectual capital at Skandia.” *Long Range Planning 30*, 3 (June 1997): 366–373.

[IADE 2003]
IADE. *Modelo Intellectus: Medición y gestión del Capital Intelectual*. Instituto Universitario de Investigación / Universidad Autónoma de Madrid.

[Marr 2008]
Marr, Bernard. *Impacting Future Value: How to Manage your Intellectual Capital*. http://media.journalofaccountancy.com/JOA/Issues/2008/09/MAG%20IntCapital-Eng.pdf (2008).

[Nickols 2000]
Nickols, Fred. *The Knowledge in Knowledge Management.* http://www.nickols.us/knowledge_in_KM.pdf (2000).

[Nonaka 1991]
Nonaka, Ikujiro. “The Knowledge-Creating Company.” *Harvard Business Review* (November 1991).

[Petty 2000]
Petty, Richard & Guthrie, J. “Intellectual Capital Literature Review: Measurement, Reporting, and Management.” *Journal of Intellectual Capital 1*, 2 (2000): 155–176.

[Qian 2010]
Qian, Rui. “Research on Information Disclosure of Intangible Assets for Software Enterprises,” 1–4. In *International Conference on E-Product E-Service and E-Entertainment (ICEEE),* Henan. China, November 2010. IEEE Computer Society, 2010.

[Scacchi 2002]
Scacchi, Walt. *Process Models in Software Engineering.* Edited by John J. Marciniak. John Wiley & Sons, Inc., 2002.

[Stewart 1998]
Stewart, Thomas & Ruckdeschel Clare. “Intellectual Capital: The New Wealth of Organizations.” *Performance Improvement 37*, 7 (September 1998): 56–59.

[von Wangenheim 2010]
von Wangenheim, Christiane Gresse, Hauck, Jean C. R., Zoucas, Alessandra, Salviano, Clenio F., McCaffery, Fergal, & Shull, Forrest. “Creating Software Process Capability/Maturity Models.” *IEEE Software 27*, 4 (July 2010): 92–94.

7.8 Authors’ Biographies

Dr. Maria-Isabel Sanchez-Segura has been a faculty member in the Computer Science Department at Carlos III University of Madrid since 1998. Her research interests include software engineering with a focus on processes, methodologies, reuse, management and software configuration management, and recently on intelligent organizations. Maria-Isabel holds a BS in Computing

(1997), an MS in Software Engineering (1999), and a PhD in Computing (2001) from the Universidad Politecnica of Madrid.

Alejandro Ruiz-Robles has been a faculty member in the Information Systems Department at University of Piura since 2004. His research interests include software engineering with a focus on processes, knowledge management, and intelligent organization management. Alejandro holds a BS in Industrial and Systems Engineering (2003) and an MS in Computer Science (2011)

Dr. Arturo Mora-Soto is a faculty member in the Computer Science Department at Carlos III University of Madrid (UC3M) since 2006. Before joining UC3M, Arturo was an associate professor at the University of Celaya (Mexico); he also worked as a software engineering consultant in Mexico for private and government organizations. His research involves collaborative learning, web technologies, knowledge management, and software engineering with a focus on process improvement, collaborative software development, and methodologies. Arturo holds a BSc in Computing Engineering (2002) from Celaya Institute of Technology (Mexico), an MSc in e-Commerce (2006), an MSc in Computer Science (2007), and a PhD in Computer Science and Technology from Carlos III University of Madrid.

Dr. Javier Garcia-Guzman is an associate professor in the Computer Science Department at Carlos III University of Madrid. He is a software process improvement consultant in Progresion. He has 10 years of experience as a software engineer and consultant in public and private companies. He received his PhD in computer science from the Carlos III University of Madrid. His current research interests are related to SPI in small settings, measurement of software processes, ISO 15504 assessments, software capacity quick audits, evaluations for pre-diagnosis according to ISO 15504, and CMMI and management of knowledge related to software engineering.

8 The Economics of Process Management: Case Studies and Customer Experiences

Erich Meier, Method Park, Germany

8.1 Economical Process Management

Few would dispute the business value of enterprise and program processes. However, the cost of managing these critical assets is often viewed as excessive, given the extensive set of process management functions needing to be accomplished, including process definition, compliance management, tailoring, appraisals and assessments, and improvement and control, at both the organizational and program levels. In today's constrained budget environment, economical process management is a business imperative.

What is economical process management? Being economical means accomplishing tasks with careful, efficient, and prudent use of resources, such as cost, labor, tools, and others. Economical process management must operate in a manner that is thrifty, with little waste, or in a way that is focused on savings and efficiency gains.

Our experience in supporting organizations in implementing process management has provided insight into how this can be accomplished economically, while meeting unique business objectives. We will highlight examples from three case studies based on actual customer experiences, where economical approaches to process management were successfully deployed and yielded tangible business benefits.

8.2 Case Study: Focus on the End User

In the first case a global space and aerospace organization was able to dramatically improve the usability of engineering processes by employing an optimized meta-model to support analysis. Visual representation and objective evaluations enabled consistent and comprehensible definitions.

Engineering work is mostly driven by people, so the process should act as a supporting framework for efficiently performing the work or simply help them to "do the right things with the right people at the right time." As a result, while designing the process visualization, the focus was therefore always on the end user.

The organization chose to use process flow and swimlane diagrams to show the relationships between phases, activities, roles, and work products. This enabled the organization to analyze the work product flow and optimize it from an economic standpoint (e.g., minimize the number of work products in each process, minimize work product handovers between different work units). As a key element, the processes were solely designed by the subject matter experts; the process experts only guided them in the correct use of the process framework. By doing so, the organization successfully avoided the "process experts designing processes for process experts" pitfall.

As a result, process waste was reduced, the process acceptance was high from the beginning, and a lean approach to process deployment was achieved. The whole change project took only two months, and even with investments in external support and licenses of a process management software, the organization was able to realize a return on investment in less than nine months.

8.3 Case Study: Automate Where Appropriate

The second case shows how a leading global automotive manufacturer dramatically improved process fidelity using a process enactment solution to ensure that there were no lapses in deployment of the documented process. This eliminated the need for excessive and redundant process assurance, which resulted in cost savings and yielded the expected business benefits from following the process as documented.

The organization was faced with the challenge that product innovation and market pressure forced once-separate business units to cooperate in developing integrated solutions involving mechanics, electronics, and software. The major roadblock for moving to the new structure was that the business units were using different processes and had implemented them with a set of different tools.

The organization not only decided to choose a unified tooling platform, but also to streamline and optimize their processes. As the products of the different business units continue to have very different characteristics (such as software-intensive infotainment systems versus safety-critical, deeply embedded powertrain applications), rigidly standardized processes and tools all over the whole organization would have resulted in a suboptimal approach for all business units (either “least common denominator” or “bloated merge of everything”).

The key element in their solution was to use the process definition for configuring and driving the engineering tool platform that is used to automate the processes. For maximizing the economic value of process automation, the organization focused on process steps that are executed with a high frequency and with frequent variance. The process might vary between different business units, different projects, and even different product releases in the same project.

Whenever a change request is entered into the new system and decided upon by the change control board, the necessary work tickets are automatically created and assigned using the documented process as a rule set, like different sets of work tickets for safety-critical changes, changes requiring supplier interaction, or changes that result in design changes.

As a result, the organization completely avoided a gap between the documented and the executed process, because it “executes the documented process.” The organization is able to tailor or modify the process without requiring any manual changes in their tools platform, which drastically reduces turnaround times for modified processes from weeks to a few minutes. Using this economical approach, the organization was able to successfully roll out the processes to over 3000 engineers in less than a year.

8.4 Case Study: Focus on Process Performance

The third example centers on economic strategies for assuring ongoing fidelity to a variety of process and safety frameworks including CMMI [CMMI 2010] and CENELEC [CENELEC 2012] standards. This will illustrate lean methods for creating current snapshots and baselines, which can be audited for conformance with numerous frameworks and standards. Accurate visibility into current practices is achieved while minimizing labor spent solely to gather data in support of an audit. In this case, the guiding principle was to "optimize the process performance, instead of working for the audit."

When engineering a process that is optimized for a particular context, the clarity and visibility of the process components is greatly increased. Instead of "designing processes for CMMI or ISO," the organization—a leading supplier of rail signaling components—designed and optimized its process components and mapped them to the requirements of the respective standards. This supported transparency and a more systematic role for frameworks and standards as drivers for process design, not merely retrospective validation criteria.

Appraisal evidences and audit artifacts are automatically generated while performing the process instead of being created solely for the sake of the appraisal or audit. This specific sample organization was able to show an effort reduction of over 60 percent for appraisal preparation after the first year because of this approach.

8.5 Summary

A key step in economical process management is to prioritize use cases based on business needs, and identify non-recurring and recurring activities, since economizing on the latter typically yields a higher payoff. In all of the case studies examined, the economics were initially applied to only a subset of process management functions with plans to address economizing on other functions in the future.

In the demonstrated case studies, the most important principles were as follows:

- Let subject matter experts design the processes while strictly focusing on the end user.
- Automate recurring processes where appropriate.
- Focus on process performance while automatically generating process compliance evidences instead of creating them solely for proving compliance.

In all cases, the organizations were able to present significant facts to senior management, resulting in higher attention, support, and probability of success.

8.6 References

[CENELEC 2012]

CENELEC. *Software for railway control and protection systems.* http://www.cenelec.eu/dyn/www/f?p=104:110:2544619326723163::::FSP_PROJECT,FSP_LANG_ID:20508,25 (2012).

[CMMI 2010]

CMMI Product Team. *CMMI for Development, Version 1.3* (CMU/SEI-2010-TR-033). Software Engineering Institute, Carnegie Mellon University, 2010.
http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm

8.7 Author's Bio

Erich Meier is the CTO of Method Park, overseeing the global products business, technology vision, and roadmap. Erich is the creator and chief architect of Method Park's "Stages" process management system. Before joining Method Park, Erich designed and developed web-based information systems for universities and deployed them at organizations all over Germany.

Erich holds a PhD in Computer Science from the University of Erlangen in the field of scalable distributed systems.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. **AGENCY USE ONLY**
(Leave Blank)

2. **REPORT DATE**
September 2012

3. **REPORT TYPE AND DATES COVERED**
Final

4. **TITLE AND SUBTITLE**
SEPG Europe 2012 Conference Proceedings

5. **FUNDING NUMBERS**
FA8721-05-C-0003

6. **AUTHOR(S)**
Jose Maria Garcia, Javier Garcia-Guzman, Javier Garzas, Amit Arun Javadekar, Pat Kirwan, Joaquin Lasheras, Fuensanta Medina-Dominguez, Erich Meier, Arturo Mora-Soto, Ana M. Moreno, Radouane Oudrhiri, Fabrizio Pellizzetti, Alejandro Ruiz-Robles, Maria-Isabel Sanchez-Segura, Prasad M. Shrasti, Aman Kumar Singhal

7. **PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. **PERFORMING ORGANIZATION REPORT NUMBER**
CMU/SEI-2012-SR-005

9. **SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
SEI Administrative Agent
AFLCMC/PZE
20 Schilling Circle, Bldg. 1305, 3rd floor
Hanscom AFB, MA 01731-2125

10. **SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. **SUPPLEMENTARY NOTES**

12A **DISTRIBUTION/AVAILABILITY STATEMENT**
Unclassified/Unlimited, DTIC, NTIS

12B **DISTRIBUTION CODE**

13. **ABSTRACT (MAXIMUM 200 WORDS)**
The 17th SEPG Europe conference was organized by the Software Engineering Institute (SEI) in collaboration with *IEEE Software* magazine and took place from June 5–7, 2012 in Madrid, Spain. SEPG Europe's goal is to bring together software and systems professionals who have a common passion to improve the processes they use to create the products and services they develop. This year, collaboration between SEPG Europe 2012 and *IEEE Software* magazine gave technical session presenters the chance to have a paper selected by the SEPG Europe 2012 Technical Committee for inclusion in this SEI report, as well the opportunity to be considered for inclusion in a future issue of *IEEE Software* magazine. This report contains the seven papers selected for publication by the SEPG Europe 2012 Technical Committee.

14. **SUBJECT TERMS**
Process improvement, CMMI, case studies

15. **NUMBER OF PAGES**
114

16. **PRICE CODE**

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