# Establishing a Software-Subcontracting Management Model to Improve the Software-Subcontracting Process in Small-size Enterprises

Garcia, I. and Pacheco, C.

Postgraduate Department
Technological University of the Mixtec Region. Mexico (*www.utm.mx*)
{ivan@mixteco.utm.mx,dago@mixteco.utm.mx}

**Abstract.** Software subcontracting using external suppliers could be the best option in reducing the investment and implementation costs of new solutions, and would allow an organization to use their resources more efficiently. Two centuries ago John Ruskin said: "the quality never is an accident; always is the result of an intelligence effort" Subcontracting is an intelligent decision, and particularly now when small-size enterprises are facing the major economic power of large corporations. The subcontracting process offers a 'nearby colleague' to increase the benefits and quality of service of the enterprises, at the same time reducing the investment. This paper illustrates the design of an alternative model to improve the software-subcontracting management process. The model is orientated towards to small-size enterprises, as the high implementation costs of the methods reviewed in this study are well known. As additional work, this paper depicts the current situation of software-subcontracting processes, reviewing the principal models applied in industry, and establishes a base comparative of the developed model. The results obtained by the model's implementation are presented to expose its applicability.

**Keywords:** Software-subcontracting process, service model, effective practices, small-size enterprises.

## 1 Introduction

The software-subcontracting process is defined as the process of acquiring, either partially or totally, Information Technology (IT) from an external services supplier [17]. It means delegating, via a contract, all or part of the software work to an external company that joins in the client organizational strategy and seeks to design a solution to existing problems within this. In recent years, software subcontracting has been of increasing interest to researchers and a considerable number of organizations. In spite of its origins in the 60's and 70's [24], the subcontracting process was only applied to financial areas and support operations; nowadays the organizations are focused on saving costs; paying more attention to their essential business processes; accomplishing economic, strategic and/or technological advances; or, basically, determining if their IT functions are unnecessary, ineffective or incompetent. It is important to say that not until the Eastman Kodak study [25] was presented in 1989, was the subcon-

*Received 30/04/10*
*Final version 19/05/10*
*Accepted 14/05/10*

tracting process considered as an efficient strategy for organizations to select, and emerged as one of the ten fundamental issues for their survival. However, instead of the successful experiences that had been published [5] [25] [14] [4] [28], the projects subcontracting failures still remain within organizations because the organizations that subcontract loose control of the acquired software products/services, as well as the usual problems of delayed delivery, costs exceeded, the poor quality of subcontracted products, and more [13].

The remainder of the paper is organized as follows. Section 2 presents the background to software-subcontracting process that established the motivations for proposing our alternative model. The principal models and standards used to manage the software-subcontracting process are analyzed in Section 3. Section 4 presents an empirical comparative between models to emphasize the significant disadvantages for applying them in small enterprises. Section 5 presents a detailed description of the proposed model. To demonstrate the feasibility of our model, the experimental results are explained in Section 6. Finally, conclusions and future work are drawn in Section 7.

## 2 Motivations

Some small companies have had good experiences subcontracting for example the payroll and accounting management departments, leaving these services in professional hands and obtaining important cost reductions. But, if these companies have good results subcontracting those processes, why do they not also delegate software services and processes to an external supplier? Nowadays, the principal disadvantages to subcontracting software services are fundamentally the lack of guidance for its implementation, the need to commit to dedicating important long term resources, and the Return of Investment (ROI) which is just too much to expect for many companies [12] [20]. The fast rate of technological innovation in IT impedes many companies from being up-to-date, but they do not want to 'miss the technological train' and the definition of alternative methods is necessary to avoid the system's obsolescence [8]. Nevertheless, the software-subcontracting process is complex because it is managed externally in order to acquire products, systems and services, rather than internally, where they could manage their own processes. As a result, the identification of effective practices in the delegation of responsibilities of the software services and processes are focused internally, to ensure that the subcontracting process will be effective, and externally, in order that the company can manage the subcontracting process and take control of their suppliers. These effective practices provide the fundamental component of a subcontracting discipline and the rigor that enables the rapid development of products/services with a high level of success [3].

This approach requires a renewed dedication to ensure that a defined, implemented, measured and maintained software-subcontracting process exists. The implementation of an effective software-subcontracting process not only makes the technological difference in an organization with regard to its competitors, it also facilitates the concentration of their own resources in the 'core business', focusing on the activity for which the organization was created, generating higher benefits without losing the capacity of maintaining service quality for clients and identifying new business opportunities [9].

## 3 Related Work

This Section aims to show a brief review of the most significant commercial models and standards used to manage the software-subcontracting process in big software organizations. Unfortunately, most models could not be implemented in small enterprises because they were created for big organizations. Besides, small-size enterprises usually confuse two terms; “acquiring” and “subcontracting” and try to adopt a model without a clear understanding of both terms. Currently, there are two approaches to guide small enterprises in adopting an efficient software-subcontracting process: to manage and monitor the subcontracting process effectively according to a defined standard; to establish deliverable products and staff roles and responsibilities, etc.; and to adapt commercial models within an improvement initiative. Almost all small enterprises fail when implementing these models, but they still use them; so, it is convenient to analyze each one and try to identify a set of effective practices.

One of the process areas included in CMMI-DEV 1.2 is the Supplier Agreement Management (SAM) process area [6]. The purpose of Supplier Agreement Management is to manage the acquisition of products from suppliers through a formal process. According to its description, the model provides seven practices to perform the SAM process. This ‘managed’ process begins with the identification of the acquisition type to perform and closes with the transition of the subcontracted products to the project. The research by [26] presents a business workflow process model for SAM process area of CMMI: Capability Level 2. It consists of three layers: contextual layer, elaboration layer, and definition layer. A software tool called Supplier Agreement Management Tool was also developed to help integrate the details of this approach. However, the fundamental disadvantage of CMMI models is the high cost of implementation, unaffordable for small enterprises. Small enterprises must obtain a “certification” of CMMI’ Capability Levels for each process that they do not consider a strategic decision; so, it is very difficult to address the software subcontracting problems with this excessively formal model.

The ISO/IEC 15504:2004 standard goes beyond quality audits to help an organization assess how well its processes perform. One standard’s objective is to guide organizations in software-subcontracting process through the determination of the potential supplier’s capacity [15]. This evaluation enables risk identification, related to each supplier, when an organization subcontracts software products of services. The purpose of ISO/IEC 15504:2004 subcontracting and supply processes is to obtain the product and/or service that satisfies the need expressed by the customer (the standard assumes this is a software oriented product or service –this is apparent in the supply process). According to the standard, the software-subcontracting process begins with the identification of a customer requirement and ends with the acceptance of the product and/or service. However, ISO/IEC 15504:2004 is more known as a Software Process Improvement model, but there is no strong evidence that it had been useful in implementing software-subcontracting processes in small enterprises. Besides, process dimension should be wider and covers all possible lifecycles applicable in small enterprises; it is impossible that all process attributes be universal and can be used by all processes and base practices without an expensive cost.

In SA-CMM an individual subcontracting process begins with the definition of a customer need and ends with the contract closure. SA-CMM is designed to be sufficiently generic for use by any government or industry organization, regardless of size, for subcontracting products. According to the SEI [21], effective subcontracting processes are critical to the success of process improvement, but the output quality is only determined within the context of organizational business-needs. There is relevant research using SA-CMM in industrial environments. Wong [29], for example, summarized software-subcontracting management lessons learned from a complex multi-disciplinary and contract environment. He also identified a number of measures for improvement in a project. Later, these measures were analyzed against SA-CMM Process Areas for their applicability and comprehensiveness. However, the "problem" for demonstrating the small enterprises' applicability endures.

COBIT provides good practices for the management of IT processes in a manageable and logical structure, meeting the multiple needs of enterprise management by bridging the gaps between business risks, technical issues, control needs and performance measurement requirements. Information systems subcontracting, development and maintenance should be considered in the context of the organization's IT long and short-term plans. The organization's system development life cycle methodology should provide for a software-subcontracting strategy plan, by defining whether the software will be acquired off-the-shelf, developed internally, through contract or by enhancing existing software - or a combination of all of the above [7]. But more than a guideline of implementation, COBIT is designed to help in understanding and managing the risks and benefits associated to information and IT related. It is not a model that focuses exclusively in the software-subcontracting process and it has a noticeable tendency towards the supervision of the IT generic processes.

ITIL's Availability Management provides reliable access to IT services. Availability means that the client will always receive the expected services as necessary. The main benefits are: supplier performance improvement and detailed information availability for negotiations at service level [19]. In spite of the use of ITIL in small enterprises, there are only three processes that are not completed covered by the model: Supplier Management (related to software-subcontracting process), Business Relation Management, and Service Report. The small enterprise should be very carefully if tries to implement some process from ITIL, because it requires a lot of time and money without warranty of success.

The IEEE Recommended Practice for Software Acquisition 1062 recommends a set of useful quality practices that can be selected and applied during one or more steps in a software acquisition process or software-subcontracting process. According to standard's documentation, IEEE 1062 is designed to help organizations and individuals to incorporate quality considerations during the definition, evaluation, selection, and acceptance of supplier software for operational use; and to determine how supplier software should be evaluated, tested, and accepted for delivery to end users [22]. However, the most common problem with the IEEE 1062 standard is the lack of mechanism in planning the subcontracting-project and to elicit and monitor the subcontracting requirements. Using the standard is too difficult for small enterprises because it requires a lot of documentation and assumes that the organization knows the standard very well.

As we see, there is wide research in developing models and standards to establish efficient software-subcontracting process. However, almost all models and standards are not affordable for small-size enterprises but provide a set of effective practices that could be customized in a simple model.

## 4 An Empirical Comparative of Subcontracting Models

Some research and critique has already been done into the reviewed models and standards; important criteria to be considered when developing an alternative approach. Research by [16], for example, discussed the implementation of CMMI in small enterprises and showed the following disadvantages: there is not a customized guideline; continuous representation enables the selection of only those process areas where the organization feels comfortable; an exponential increase in areas and practices, time, resources and cost; an excessive standardization for small organizations that work and evolve in different ways to large ones.

Research by [27] provides some important information about the standard ISO/IEC 15504:2004. Reaching the capacity dimension is a big difficulty in small enterprises; there is overlap with the process dimension. Assessment complexity (and its cost) is significantly higher than other models. In the context of the software-subcontracting process, the ISO standard does not provide a formal guide to monitoring a product when it has been accepted.

SA-CMM has a high level of complexity; the period of adoption in small enterprises is longer than in larger enterprises. In this case, the statement of *"simplifying activities does not affect the professional level"* does not apply [2] [29].

More than an implementation guideline, COBIT is designed to help organizations to understand and manage the risks and benefits associated with information and related IT. COBIT is not a model that exclusively focuses on the software-subcontracting process and shows a clear tendency to supervise the IT processes.

With respect to ITIL, there are three processes that are not explicitly covered: Subcontracting Management, Business Relation Management, and Service Level Report. These processes are ignored and simultaneously fitted within the ITIL frame, for example the Subcontracting Management is included in the Service Level Management. On the other hand, the experts notice that ITIL has a list of minimum requirements against which an organization can be evaluated but it does not indicate how to reach a level of "conformity".

Despite their efficiency, the reviewed models make the same errors:

- The models specify what activities perform, however they do not provide guidelines about how to do it, in both engineering and project management.
- The models do not provide procedures for internal project management which should include own templates to facilitate the process, and fundamentally.
- There is no a customized guideline for small enterprises.

To summarize, in Table 1 we provide a homogenized analysis of the models; identifying important characteristics that we consider relevant in proposing an alternative model.

Some authors, for example [23], consider that standards reduce the developer's autonomy in large enterprises; developers are overloaded with extra work, assume coercive restrictions that suffocate the creativity required in innovating software development, and focus purely on the process, ignoring the people involved.

However, there is one that defends interdependence (as opposed to independent work) and takes a collaborative form [1], mature levels of process make a development process more 'socialized' where the collaborative effort increases efficiency and effectiveness (in our case when companies try to subcontract software as business strategy). We try to explore the benefits of the last approach, adapting it in small enterprises.

**Table 1.** Comparative analysis over reviewed models and standards

| **Criteria** | **Models and Standards** | | | | | |
|---|---|---|---|---|---|---|
| | **CMMI-DEV** | **ISO/IEC 15504** | **SA-CMM** | **ITIL** | **COBIT** | **IEEE 1062** |
| Use of procedures for internal supplier management | X | P | P | X | X | P |
| Use of templates for service management | X | X | X | X | X | P |
| Establish a Service Level Agreement | X | X | X | P | X | X |
| Incorporate an on going process improvement | P | P | P | P | X | X |
| Use of metrics for management | P | P | P | X | P | X |
| A contract requirement from start to finish | P | P | P | P | P | P |
| Relevance of the customer/supplier relationship | P | P | P | P | X | P |
| Customized for small enterprises | X | X | X | X | X | X |
| High cost of implementation | P | P | P | P | P | P |

## 5 Definition of a Customized Model for Small Enterprises

None of the previous models were created to address the particular small enterprises necessities. We propose an alternative model for managing the software applications/services that any company subcontracts, the Software-Subcontracting Management Model (SSMM). This model attempts to cover the strengths and weaknesses determined in Table 1.

This model enables the organization to improve its software-subcontracting management process. We start with the idea that to perform service management, it is necessary to define the activities/tasks to execute; when these should be performed; and what entries and outputs should be obtained. Our model is a guideline for suppliers and helps them in the project transference and operative service procedures execution. SSMM stages insist on those aspects that have major difficulties. The service management that our model offers is based on the efficient coordination of the three "Ps": *people*, *processes* and *products* through the three stages depicted in Table 2.

**Table 2.** Phases and generic objectives of SSMM

| Stage name | Objectives per Stage |
|---|---|
| Initial stage | This should be planned with the client. A Project Plan must be established. |
| Stabilization stage | Supplier takes total responsibility for service according to Service Level Agreements (SLA). The duration is set in the contract and is "specific" for each client. |
| Closure stage | The closure of the service is prepared and the transition to the client is performed. A formation plan should be developed to prepare those personnel which assume the service maintenance and continuity. |

SSMM helps to define the Software Subcontracting Management process tasks, identifying the correct moment for its application and recognizing the inputs and outputs required by the process. This model begins when the supplier is in charge of the service and finishes when the contract finalizes. SSMM is a guide for the operative transference of the project and it focuses on those aspects detected as deficient and incomplete, reducing the stages of the life cycle [5], [17], [18], [10] from four to three. Previous research determined that many subcontracting practices had not been used by the assessed small enterprises. Now, we are trying to define and implement effective practices according to the real characteristics and necessities of these very important organizations.

Each stage of SSMM defines a set of effective practices that it is necessary to accomplish to continue to the next stage. The model also provides a set of activities examples to help the companies to implement the methodology, and includes an expected list of work products for each stage (see Figure 1). The model stages are defined as follows:

- **Initial Stage.** In this stage the SLA is established, and the timescale for this is not less than one month and no more than three months. Of course this depends on the scope and the complexity of the service to be subcontracted. The activities of the stage are planned with the client, producing a Project Plan and the SLA. The sequence of activities is neither “rigid” nor “sequential”, as it is possible to perform these in parallel.
- **Stabilization Stage.** The external supplier takes control of the services previously defined in the SLA. The stage duration is established in the contract (in years) and is specific to each client.
- **Closing Stage.** The objective of the stage is to prepare the conclusion of the service and perform the transference of the service to the client or a third party supplier. This stage covers the following aims:
  - To perform the activities of evolutionary and corrective maintenance and/or new developments, in such a way that the SLA is not affected.
  - To plan the service transference defining the assumptions and conditions, and economic impact of the same.
  - To analyze the services to return from the point of view of the complexity of resources, significance and availability.
  - To develop a training plan oriented to the personnel that will assume the continuity of maintenance.
  - To transfer existing knowledge in documents, records, and more.
  - To transfer sources, libraries, supports and services.

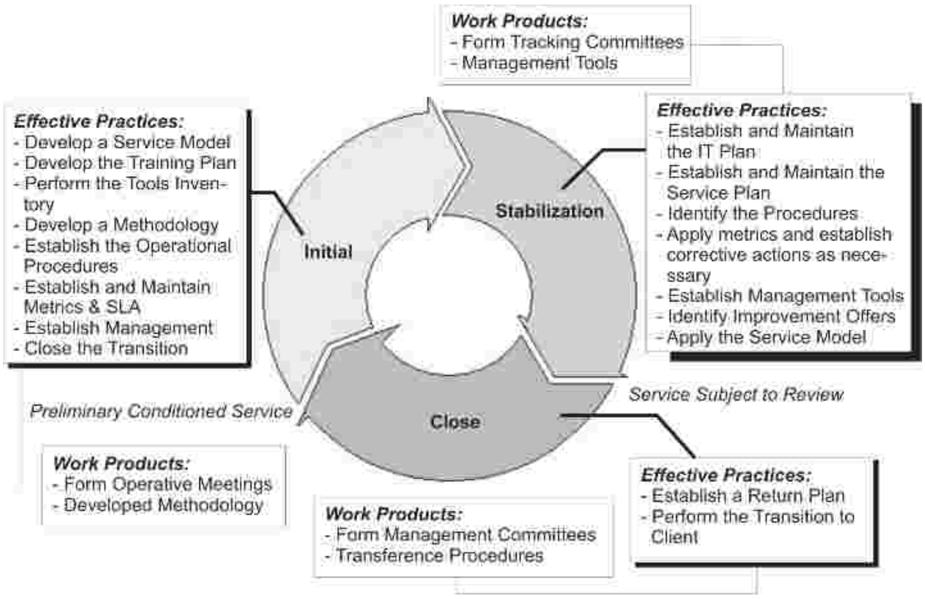
Work Products:
- Form Tracking Committees
- Management Tools
Effective Practices:
- Develop a Service Model
- Develop the Training Plan
- Perform the Tools Inventory
- Develop a Methodology
- Establish the Operational Procedures
- Establish and Maintain Metrics & SLA
- Establish Management
- Close the Transition
Effective Practices:
- Establish and Maintain the IT Plan
- Establish and Maintain the Service Plan
- Identify the Procedures
- Apply metrics and establish corrective actions as necessary
- Establish Management Tools
- Identify Improvement Offers
- Apply the Service Model
Initial
Stabilization
Close
Service Subject to Review
Preliminary Conditioned Service
Work Products:
- Form Operative Meetings
- Developed Methodology
Work Products:
- Form Management Committees
- Transference Procedures
Effective Practices:
- Establish a Return Plan
- Perform the Transition to Client

**Fig. 1.** SSMM stages

### 5.1 The SSMM structure.

We believe that adequate management of the subcontracting process needs a set of elements that it is necessary to implement during the Transition stage, incorporated in SSMM. These elements can be provided by both the client and the supplier. However it is recommended that it comes from the providing company in order to obtain homogeneous services. Figure 2 shows the general SSMM structure.

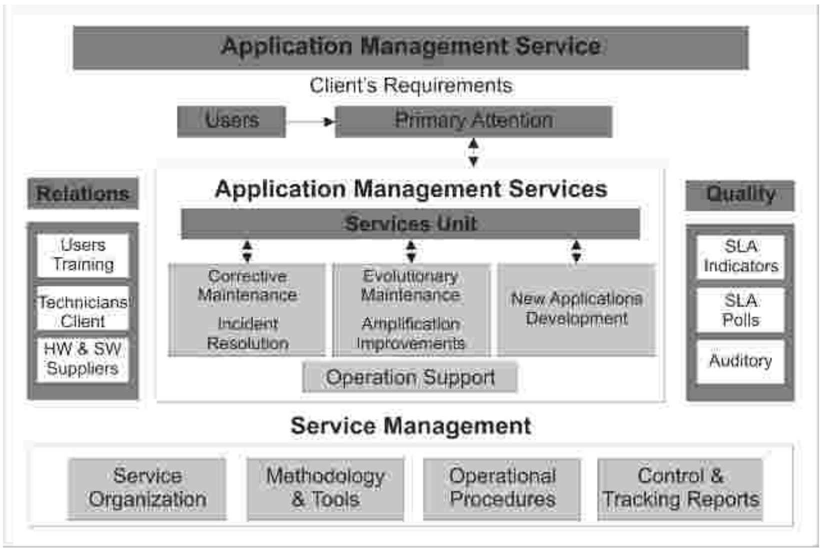

**Fig. 2.** SSMM structure.

We provide a brief description of SSMM elements:

- Primary Attention. This is the communication channel for managing incidents and requests between final users and the Services Unit. Commonly known as the Help Desk.
- Services Unit. This is the set of technical personnel in charge of service delivery. Among the developed activities are:
    - Corrective Maintenance. Activities for incidents and failures resolution produced in the software under operation.
    - Evolutionary Maintenance. Activities for the accomplishment of modifications and improvements in the software under operation.
    - New developments. Activities for management and implementation of new software projects.
    - Operational Support. Activities for the resolution of specific or information management requests not structured as available functionality for the final user.
- Relations. These relations determine the responsibility matrix in the different activities carried out during the service: Users Training, Client Technical Personnel and HW and SW Suppliers.

- Quality. The quality enables performing the objective or subjective evaluation of the performed services.
- Service Organization. This describes the terms and conditions of the agreement between client and supplier, by which the latter company will provide IT services during the Ongoing Stage. The services scope and company responsibilities would be specified.
- Methodology. This systematically defines the way “to do things” in the software life cycle.
  - Support Tools. The set of client tools and supplier management tools, or a third party that can complement them.
- Operational Procedures. These define the operative flows between different client areas and the service unit.
- Control and Tracking Reports. These are the sets of generated reports that realize the service control and tracking.

## 6 Testing the Model

To evaluate the applicability of our model, we have implemented an improvement initiative within one Mexican small enterprise. TICDES® Software is a privately owned small software development enterprise with about twenty software engineers, five administrative staff, and two project leaders. This organization specializes in software development and software consultancy. The size of the projects that TICDES® has developed in the past are up to 35,150 LOC with a duration ranging from a couple of weeks to 4,000 hours. The number of people participating in the teams has ranged from three to seven. From the beginning, the company was created with the goal and commitment to develop quality software. Before this research, the company had not decided to launch an improvement initiative because of the lack of information on it. The current software demand exceeds the enterprise capabilities and a software subcontracting process is an alternative solution to avoid the reduction of client portfolio. This year, TICDES® decided to start an alternative software process initiative to attempt to establish an effective subcontracting process according to its characteristics.

### 6.1 Obtaining historical data.

Testing SSMM required selecting data from historical projects of TICDES®. It was possible to obtain data from one software project that failed in the external development delegation. The project PV0K_EXT1 was selected because it conserved all estimations and development data just before it was canceled. TICDES® does not have a formal process established to acquire external products or services, so our validation was focused on historical data. According to this, the supplier organization used a group of practitioners with university career formation to develop the required software products. A project manager and a four-member team were assigned to develop the project PV0K_EXT1. We discarded the “distance” factor as a cause of failure because both enterprises are located in the same city. Given the social and cultural factor of the development environment, we decided to use four students from last year as the development team. These students have the same capabilities and knowledge as the original development team. We played the role of project manager (supplier) and TICDES® used the same project manager as in the original project (client). We satisfied the infrastructure requirement and assigned one PC per student; a server was used to set the SMMM assets and templates; and TICDES® used Internet connection to control the subcontracted project. Table 3 shows that TICDES®' historical data reflected the same problems determined in the previous work: inability to manage the subcontracted project.

**Table 3.** Historical data from project PV0K_EXT1

| Objective | Estimated value TICDES® | Real value (at project cancellation) |
|---|---|---|
| Project planning (days) | 180 | 300 |
| Staff distribution | 4 | 7 |
| Role assignation | <1% of total time | > 5% of total time |
| Milestones reached (at first stage of project) | >30% | <10% |
| Predicted risks | 5 | 15 |
| Minutes of performed meetings | 7 (one by life cycle phase) | 2 |
| Redacted agreements | 7 (one by life cycle phase) | 1 (development contract) |
| Modified documents (configuration management) | None | All TICDES® documentation was modified (project plan, roles, cost estimations, etc.) |
| Time for formation (days) | 10 | 25 |
| Time for tool identification (tools inventory for project) | 20 | 35 |
| SLA penalization | None | There is no SLA established |
| % of injected defects | < 5% | > 15% |
| Incidences in product quality | < 2% in relation to defects | Never used |
| Metrics for SLA | 5 (all related to usability) | None |
| Effort deviation | < 15% of total time | > 32% |

### 6.2 Establishing the evaluation objectives.

TICDES®' historical data enabled us to identify four specific issues: Lack of planning of subcontracted project; deficient monitoring of subcontracted project; nonexistence of SLA by each subcontracted project; poor quality in final product, if the project can be closed.

All these issues were coherent with the obtained results resumed in [11] with the application of a two-phase questionnaire. We launched a pilot project supported by SSMM called AP_PV0K (identical to PV0K_EXT1) and the preliminary results are showed in the following section.

### 6.3 Results

The pilot project AP_PV0K was implemented following our SSMM model to manage the software-subcontracting process. The student supplier development team followed the instructions of the TICDES® project manager and reported the project status using the SSMM assets.

*Objective 1: Establish agile project planning in subcontracted projects.*
SSMM introduced the Service Model to establish the project plan and IT plan. Table 4 depicts that the responsibilities matrix included in the Service Model considerably reduced the time spent in assigning roles and tasks for the project, and TICDES® was able to estimate the project risks with more accuracy.

**Table 4.** Reaching the objective 1 for pilot project

| Objective | Estimated value TICDES® | Real value (at project cancellation) | Obtained value SSMM |
|---|---|---|---|
| Project planning (days) | 180 | 300 | 160 |
| Staff distribution | 4 | 7 | 4 |
| Role assignation | <1% of total time | > 5% of total time | <0.25 of total time |
| Predicted risks | 5 | 15 | 7 |

*Objective 2: Improve the monitoring process in subcontracted projects.*
The contract asset of SSMM identified the inconsistencies between TICDES® and its original supplier, which may have resulted in the project failure after 10 months. The Service Model established monitoring meetings and used the SSMM templates to understand the agreements; the work was assigned between a Monitoring Committee and Maintenance Committee; and the criteria to establish monitoring milestones were established. The indicator of "Minutes of performed meetings" in Table 5 was incremented because SSMM established and maintained an Acceptation Criteria Plan, Final Acceptation Plan and Project Delivery Plan that TICDES® had never considered as formal documents.

**Table 5.** Reaching the objective 2 for pilot project

| Objective | Estimated value TICDES® | Real value (at project cancellation) | Obtained value SSMM |
|---|---|---|---|
| Milestones reached (at first stage of project) | > 30% | < 10% | 32% |
| Minutes of performed meetings | 7 (one by life cycle phase) | 2 | 10 |
| Redacted agreements | 7 (one by life cycle phase) | 1 (development contract) | 9 |
| Modified documents (configuration management) | None | All documents | 0 |

*Objective 3: Establish SLA to manage the subcontracting process.*
The Metrics and SLA asset of SSMM established 24 metrics to manage the service level of products subcontracted by TICDES®. This activity increased the rate of confidence in the supplier work because it was constantly monitored, and TICDES® obtained what it really wanted. Using SSMM, a penalization percentage of 3% over total number of incidences was reflected in Table 6. We decide to show this penalization because the development team did not attend two of the monitoring meetings programmed by TICDES®.

**Table 6.** Reaching the objective 3 for pilot project

| Objective | Estimated value TICDES® | Real value (at project cancellation) | Obtained value SSMM |
|---|---|---|---|
| SLA penalization | None | There is no SLA established | 3% |
| Metrics for SLA | 5 (all related to usability) | None | 24 |

*Objective 4: Establish and maintain the desired quality of product.*

TICDES® used to apply two metrics to control the external products' quality: percentage of injected defects and the number of incidences in the final product. The person responsible for ensuring that these values do not exceed the established limit is the Software Quality Assurance Group. Unfortunately, as we explained before, it is too difficult for SMES to employ skilled personnel to perform this task. SSMM implemented 24 metrics and SLA from the Metrics and SLA asset through templates and activities to obtain better results.

The AP_PV0K project was closed after 4 months and, to this day, is working without quality problems. The project delivery was performed at the end of 2008 and obtained data was collected in February 2009. We are collecting more data to evaluate the effectiveness of the Service Model of SSMM.

**Table 7.** Reaching the objective 4 for pilot project

| Objective | Estimated value TICDES® | Real value (at project cancellation) | Obtained value SSMM |
|---|---|---|---|
| % of injected defects | < 5% | > 15% | < 1% |
| Incidences in product quality | < 2% in relation to defects | Never used | 1% |

## 7 Conclusions

The externalization of software services is more frequently becoming an option among small enterprises as a solution for maintenance and new development of software projects. The most important choice for enterprises is focused on choosing, in a formal way, who will be their supplier or technological partner. As part of this difficult decision, these organizations normally consider, among other aspects, the solid experience of their companion in this new journey. The model presented summarizes this experience in a document that enables the company, from the start, to take control of the service, making the right decision at the right moment and paying special attention to the relevant issues in each situation. We think that this model is the instrument to helping any small enterprises that provides software-subcontracting services, in managing it.

This alternative model represents the first step in this research. The next step is related to the validation of the model. For this purpose, the model is being experimented on 30 small-size enterprises through a project funded by the Spanish Ministry of Industry, Tourism and Trade.

## Acknowledgement

This paper is sponsored by everis Consulting Foundation and Sun Microsystems companies through “Research Group of Software Process Improvement in Latin America”.

## References

1. Adler, P. *Practice and Process: The Socialization of Software Development*. Working Paper Series 03-12. Univ. Southern California (2003).
2. Bach, J. “The Immaturity of CMM” *American Programmer*, 7(9): 13-18 (September 1994).
3. Bernard, T., Gallagher, B., Bate, R. and Wilson, H. *CMMI Acquisition Module (CMMI-AM) Version 1.0*. Software Engineering Process Management. CMU/SEI-2004-TR-001 (2004).
4. Calvo-Manzano, J., Cuevas, G., Garcia, I., San Feliu, T., Serrano, A., Arboledas, F. and Ruiz de, F. “Requirements Management and Acquisition Management Experiences in Spanish Public Administrations” *International Journal of Knowledge Societies and Technologies*, 1(2): 116-121 (2007).
5. Clark, T. “Corporate Systems Management: An Overview and Research Perspective” *Communications of the ACM*, 35(2): 61-75 (1992).
6. CMMI Product Team. *CMMI for Development (CMMI-DEV, V1.2)*. CMU/SEI-2006 TR-008, Software Engineering Institute, Carnegie Mellon University.
7. COBIT 4th Edition Framework. COBIT Steering Committee and the IT Governance Institute (July 2005).
8. Dahane, M. Clementz, C. and Rezg, N. “Effects of extension of subcontracting on a production system in a joint maintenance and production context” *Computers & Industrial Engineering*, 58(1): 88-96 (2010).

9. Farbey, B. and Finkelstein, A. “Software acquisition: A business strategy analysis” *Proc. of the Fifth IEEE International Symposium on Requirements Engineering* (RE’01), IEEE Computer Society, pp. 76-83 (2001).
10. Ferguson, E., Kussmaul, C., McCraken, D. and Robbert, MA. “Offshore Outsourcing: Current Conditions and Diagnosis”, *Proc. of the ACM Special Interest Group on Computer Science Education* (SIGCSE' 04), ACM Publications, pp. 330-331 (2004).
11. Garcia, I., Pacheco, C. and Sumano, P. “Use of Questionnaire-Based Appraisal to Improve the Software Acquisition Process in Small and Medium Enterprises” *Software Engineering Research, Management and Applications. Series: Studies in Computational Intelligence*, 150(14): 15-27. Springer-Verlag Berlin Heidelberg (2008).
12. Gilley, K., Greerb, C. and Rasheed, A. “Human resource outsourcing and organizational performance in manufacturing firms” *Journal of Business Research*, 57(3): 232-240 (2004).
13. Goldenson, D. and Fisher, M. *Improving the Acquisition of Software Intensive Systems*. Technical Report CMU/SEI-2000-TR-003. Software Engineering Institute, Carnegie Mellon University (2000).
14. Hietala, J., Kontio, J., Jokinen, J. and Pyysiäinen, J. “Challenges of Software Product Companies: Results of a National Survey in Finland” *Proc. of the 10th IEEE International Symposium on Software Metrics* (METRICS'04), IEEE Computer Society, pp. 232-243 (2004).
15. ISO/IEC 15504-2:2003/Cor.1:2004 (E). Information Technology –Process Assessment –Part2. International Organization for Standardization: Geneva, 2004.
16. Kulpa, M. and Johnson, K. *Interpreting the CMMI: A Process Improvement Approach*. Auer Bach Publications. 2003.
17. Lee, J., Huynh, M., Ron, K. and Shih-Ming, P. “The Evolution of Outsourcing Research: What is the Next Issue?” *Proc. of the 33rd Hawaii International Conference on System Sciences*, Hawaii, USA (2000).
18. Lee, J., Huynh, M., Chi-Wai, R. and Shih-Ming, P. “IT Outsourcing Evolution: Past, Present, and Future” *Communications of the ACM*, 46(5): 84-89 (2003).
19. Office of the Government Commerce. *ITIL Lifecycle Publication Suite, Version 3: Continual Service Improvement, Service Operation, Service Strategy, Service Transition, Service Design*. Stationery Office Publisher (2007).
20. Rodgers, T. J. “The truth about outsourcing” *IEEE Design and Test of Computers*, 22(1): 12-13 (2005).
21. Software Acquisition Capability Maturity Model (SA-CMM) Version 1.03. Technical Report CMU/SEI-2002-TR-010 (March 2002).
22. Software Engineering Standards Committee of the IEEE Computer Society. IEEE Recommended Practice for Software Acquisition. IEEE STD 1062, 1998 Edition (Includes IEEE STD 1062-1993 and STD 1062A-1998) (1998).
23. Surmacz, J. *Take my Hosting Please*. Outsourcing Research Center Reports (June 2003).
24. Venkatraman, N. and Lohl, L. “Determinants of Information Technology Outsourcing: A Cross Sectional Analysis” *Journal of Management Information Systems*, 9(1): 7-24 (1992).

25. Venkatraman, N. and Lohl, L. “Diffusion of Information Technology Outsourcing: Influence Sources and the Kodak Effect” *Information Systems Research*, 3(4): 334-358 (1995).

26. Vivatanavorasin, C., Prompoon, N. and Surarerks, A. “A Process Model Design and Tool Development for Supplier Agreement Management of CMMI: Capability Level 2” *Proc. of the 13th Asia Pacific Software Engineering Conference* (APSEC'06), IEEE Computer Society Press, pp.385-392 (2006).

27. Wang, Y. *Software Engineering Standards: Review and Perspectives*. World Scientific Publishing. 2002.

28. Weber, K., Araújo, E., Scalet, D., Andrade, E., Rocha, A. and Montoni, M. “MPS Model-Based Software Acquisition Process Improvement in Brazil”, *Proc. of the Sixth International Conference on the Quality of Information and Communications Technology*, IEEE Computer Society, pp. 110-119 (2007).

29. Wong, S. “Software Acquisition Management Experience Learnt in a Multi Discipline and Multi Contract Project Environment” *Proc. of the First Asia-Pacific Conference on Quality Software*, IEEE Computer Society, pp. 239-247 (2000).