ENGINYERIA EN ORGANITZACIO INDUSTRIAL

PROJECTE FINAL DE CARRERA

CRITICAL SUCESS FACTORS IN SOFTWARE PROCESS IMPROVEMENT

Projectista: BERTA BORAO XANXO

Director/s: NURIA TALAVERA

Convocatòria: SETEMBRE 2012

Declaration

The author declares that the text and work presented in this thesis is original and that no sources other than those mentioned in the text and its references have been used in creating the thesis.

The author is responsible for its contents.

Acknowledgements

I am heartily thankful to Dr. Ir. Jos Trienekens, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding the subject.

Contents

1. Abstract

The aim of the thesis project was to identify Software Process Improvement (SPI) success factors through a deeply literature review.

Objectives:

- Identify the models of SPI
- To identify SPI success factors through literature review.
- To identify the most important SPI success factors based on the frequency of occurrence in published research.
- To identify the variety of definitions or explanations of the most important SPI success factors in published research.

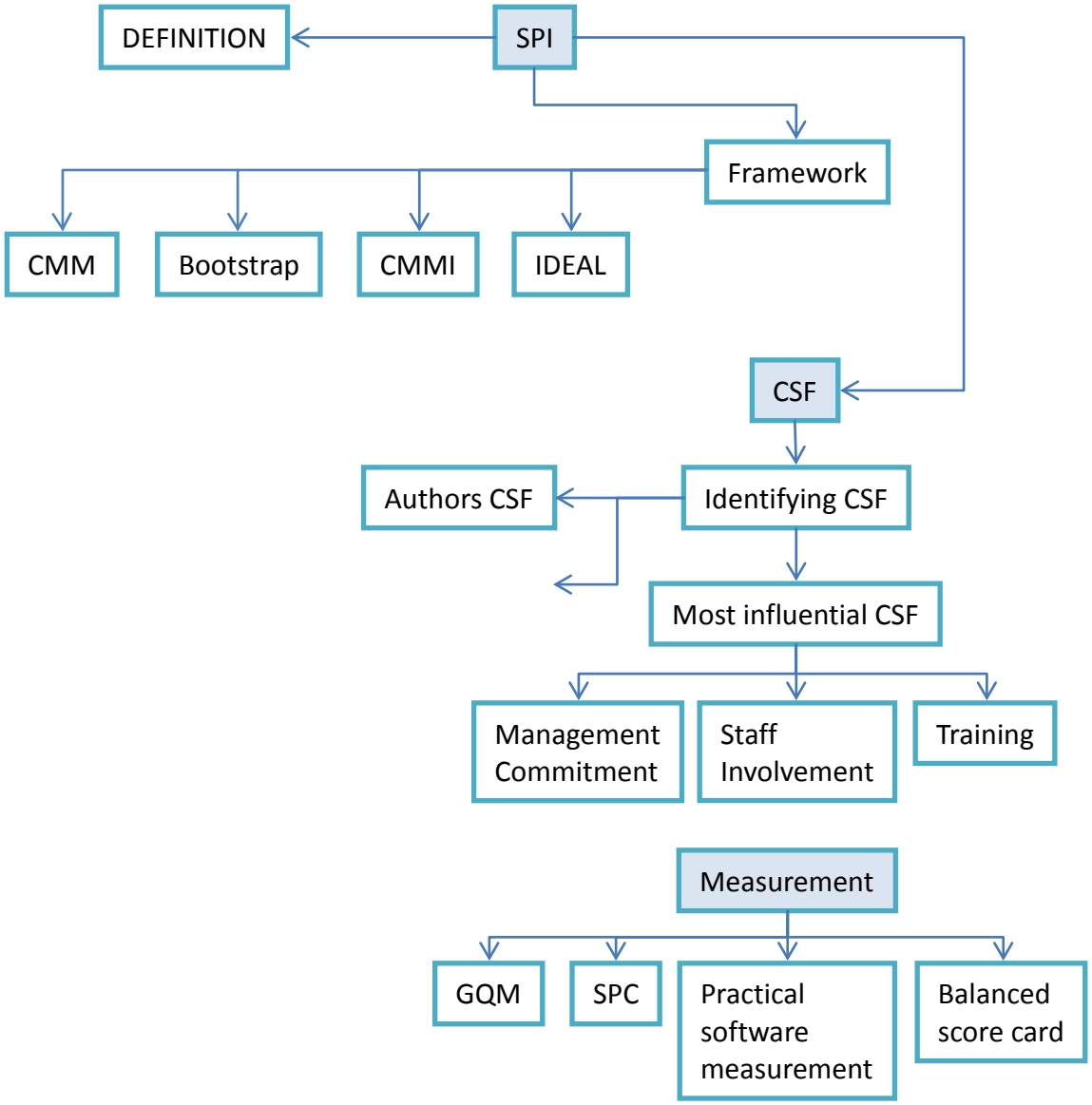

Figure 1: Abstract

2. Introduction

The role of software in the industry is becoming strategic due to software is an integral and crucial part of most products and services used in our daily lives, such as mobile phones and vehicles and performs vital functions in lot of fields. Because of these, it has increased the number of companies developing software, so the need to produce software more rapidly has emerged.

Software process improvement has become one of the main aims of these kinds of companies, because of the fact that quality of the product is closely related to the quality of the process that produces it. In order to get better products, companies need to improve their software development process.

The concept of Software Process Improvement (SPI) was developed by the Software Engineering Institute (SEI), based on the work of Watts Humphrey. SPI has influenced the software industry changing focus from the systematic approaches into improving process.

There are some models to improve software development performance like Capability Maturity Model (CMM) and the ISO 9000. These models focus on processes in order to produce quality software, reduce cost and time and increase productivity.

In addition to these models, there are factors that play a positive or negative role in the implementation of the SPI programs. These factors are called “Critical Success Factors”.

This thesis has focused on these critical success factors (CSF) that are necessary to carry out to develop and implement SPI. There are lot of theoretical and empirical studies that have research about CSF and each of them have their own list of CSF.

Humprhrey and Basili have been the pioneers and leader in the field of SPI, identifying the basic principles of software process change and improvement.

Goldenson and Herbsleb in 1995 identified the factors necessary to implement a successful SPI program, Stelzer recognized 10 factors that affect organizational changes in SPI, El-Emam (1999) studied the factors that have no impact in SPI and Niazi on 2006 analyzed all these experiences and compared with his own empirical study, the result was a list of the factors that have a positive impact on the SPI.

Nasir in 2001 conducted a study about the errors and mistakes produced because of the software development and maintenance. The term “Software Crisis‟ emerged to describe the software industry’s inability to provide customers with high quality products within schedule and under budget [Nasir, 2001].

A very clear example that Nasir give is:

> “The delay of over 16 months in the opening of Denver International Airport and the over 100 million dollars in excess of the budget in the airport’s construction cost (Swartz, 1996). “One main reason for the delay and overrun was the presence of major bugs in the baggage handling control software” (Glass, 1998).

Even today, this situation has not changed much. Software development projects are known for being completed far over budget and behind schedule [Gray and Larson, 2008].

In the same article, Nasir present a survey conducted in the United States (by The Standish Group, 1995) which reported data from several thousand IT projects, revealed a success rate of only 16% of software projects. Mean-while, 31% of projects failed while the remaining 53% had cost overruns, time overruns and impaired functionality. Of these, the average cost overrun was 189%, and the average time overrun was 222%. A recent report by The Standish Group showed a slight improvement, yet the figure remained troublesome, with a success rate of less than 40%.

Table 1 tracks the progress of Standish software project performance over a decade.

Benchmark/year	1994	1996	1998	2000	2004	2006	2008
Succeeded (%)	16	27	26	28	29	35	32
Failed (%)	31	40	28	23	18	19	24

Table 1: Standish Group: Software project performance

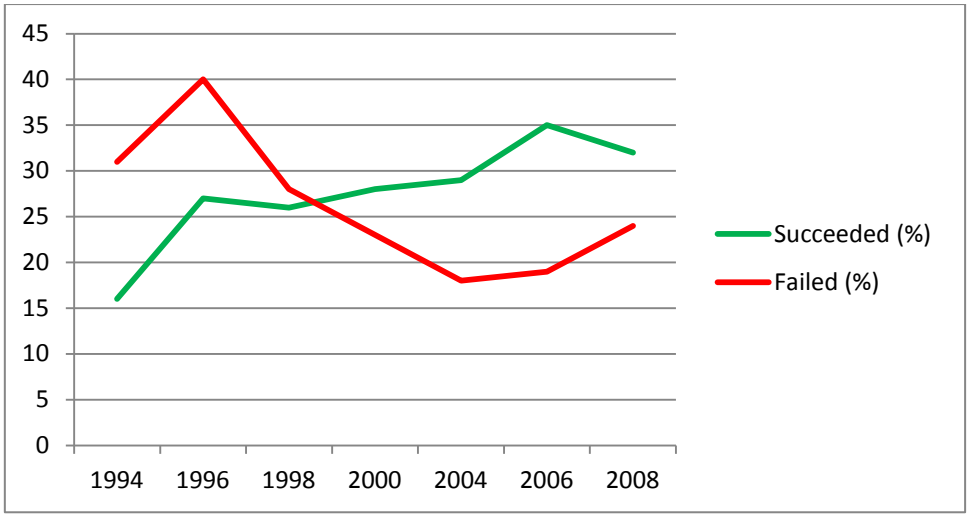

Figure 2: Standish Group: Software project performance

As Pinto and Rouhiainen said in 2001, Critical success factors will significantly improve the chances of project success.

However, after reading and analyzing all these papers, we have found a lack of definition and we have not found a method to measure and monitor the CSF.
Theses missing topics need to be investigated in order to assist SPI practitioners in designing effective SPI implementation strategies.

This thesis will focus on the most important CSF through the existing literature and it will try to find a good definition and the method to measure and monitor them.

2.1 Literature research

In this thesis it has been used a systematic literature review based on Barbara Kitchenham's study [Kitchenham et al., 2009], she proposed a guideline for systematic reviews appropriate for software engineering researchers

The guideline covers three phases of a systematic review: planning the review, conducting the review and reporting the review.

The systematic literature review is the best evidence-based approach to cover a broad spectrum of research literature on a particular subject. The advantage of the systematic literature review is that it provides an evidence of the robustness of the phenomenon under investigation [Kitchenham et al., 2009].

Research questions

For this thesis I formulated three questions:

- What factors, as identified through the theoretical and empirical studies, have a direct effect on implementing SPI?
- How these factors can be measured?
- How these factors can be monitored?

Search process

The purpose of search process is to formulate the search strings/terms, identify the search resources (databases) and describe the study search procedure

Keywords: Critical Success factors, SPI, Measurement.

I used the following databases as the primary source for publications:

- Google Scholar http://scholar.google.nl/
- IEEExplore
- ACM Database
- http://www.mendeley.com/
- http://www.computer.org
- http://www.sciencedirect.com
- http://www.emeraldinsight.com
- http://www.springerlink.com

Inclusion and exclusion criteria

It has been developed a basic process for including / excluding publications as recommended by Kitchenham et al. It comprises four steps shown in the following Figure 3.

First of all, the publication was required to be peer-reviewed (i.e. if it is published in journal, conference proceedings). Then I looked into the title and the abstract of the publication to ensure their relevance into the topic. But even if the publications did not discuss SPI in their abstract, they were not rejected initially due to there was a high risk that abstract could have been poorly formulated.

The last step was to review the conclusions, the main difference with the step before is that in this case if there was no point of relevance in the conclusions the publication was rejected immediately.

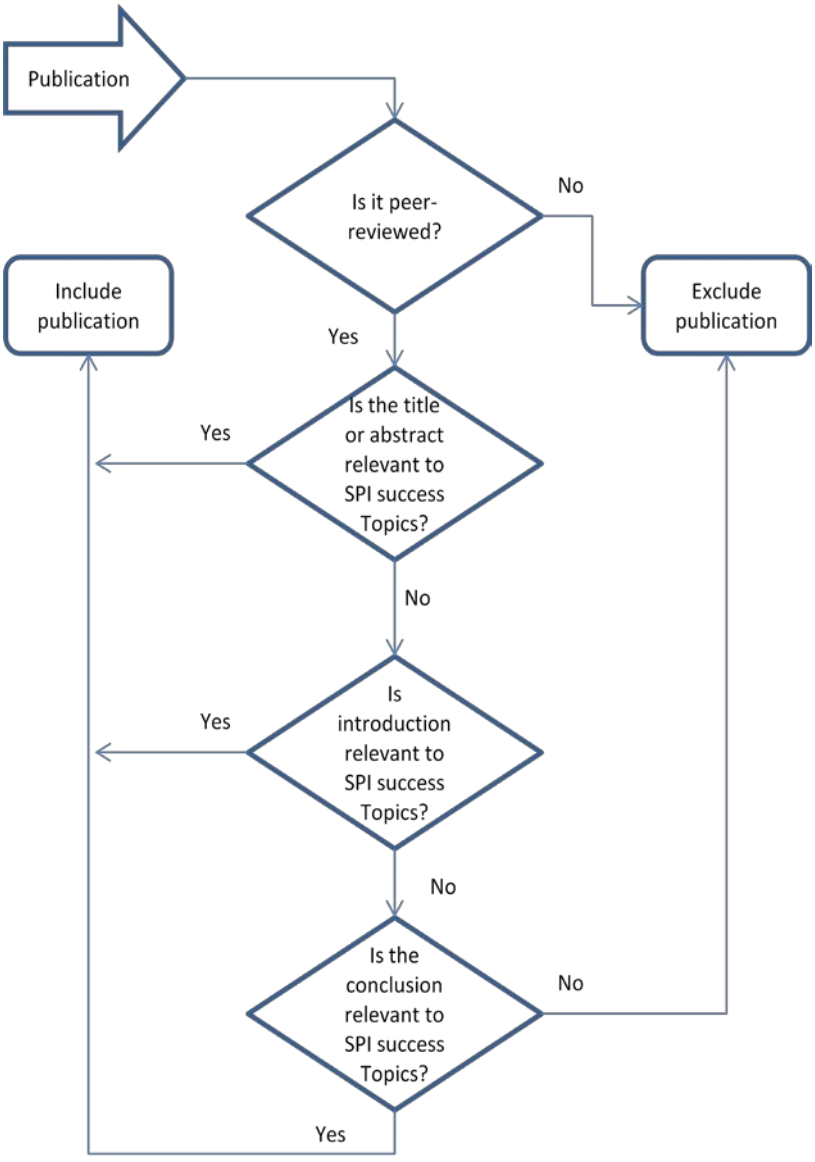

Figure 3: Inclusion and exclusion criteria

3. Software Process Improvement (SPI)

3.1 Definition of SPI

According to Watt S. Humphrey, a "software process is the sequence of steps required to develop or maintain software. It sets out the technical and management framework for applying methods, tools and people to the software task. The process definition defines roles, specific tasks, establishes measures and provides entry/exit criteria for each step."

The Software Engineering Institute (SEI) defined the software process as:

> "The set of Activities, Methods and Transformations that people use to Develop and Maintain Software and the Associated Products, for example: product plans, design documents, code, test cases and user manuals"

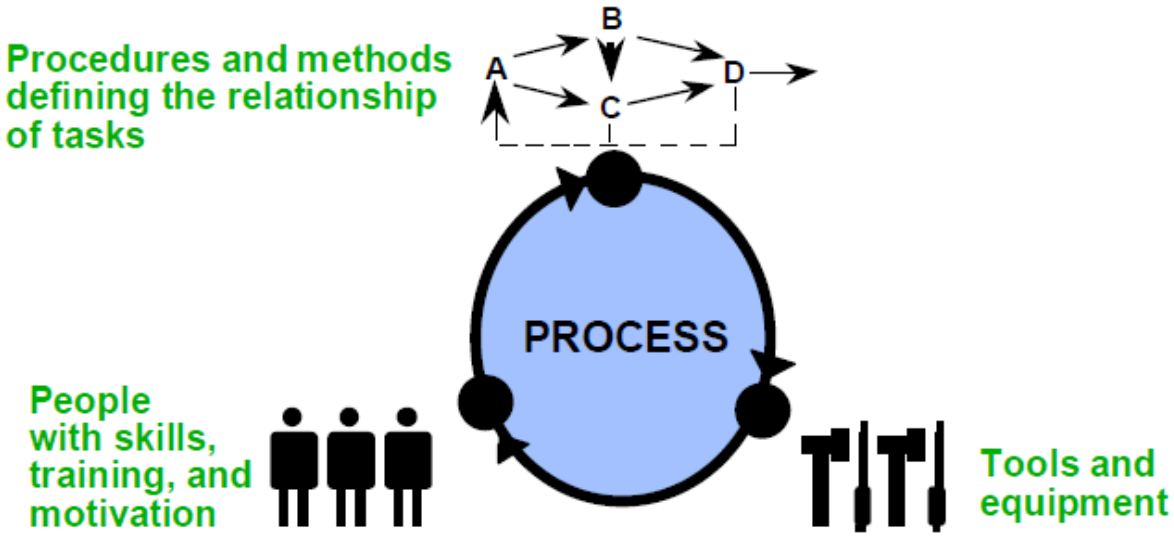

Figure 4: Graph definition of a process [Paulk et al., 1993]

The term "Software process improvement" (SPI) denotes the changes implemented to a software process that bring about improvements [Stelzer,1999]. So the intent of software process improvement is improving software product quality, increasing productivity software product quality, increasing productivity and reducing the cycle time for product development [Paulk 1993, via Stezer 1999].

Software Process Improvement (SPI) is defined as follows by Seija:

> "The purpose of improvement is often to enhance software development in order to raise the quality of software. On the other hand the goal may be to shorten the delivery cycle, to lower the costs and thus improve profitability, or to strengthen the market position. There may also be a need to prove the maturity of development, which many require changes in software development processes" [Komi-Sirviö, S, 2004].

David Rico defines Software Process Improvement (SPI) as "the discipline of characterizing, defining, measuring, and improving software management and engineering processes, leading to successful software engineering management, higher product quality, greater product innovation, faster cycle times, and lower development costs, simultaneously."

SPI is the mechanism through which the quality of software processes is improved [Aaen et al. 2001]. Thus, SPI is concerned with changing the way software development organizations, teams, and individuals perform in their work. The focus of SPI is therefore on software processes, including the actors executing these processes and structures governing the activities [Zahran 1998].

The Authors Paul et al (1993) thought that the Software process improvement aims is to:

- Improve the software product quality
- Improve productivity
- Cut-down the cycle time for do the development

3.2 Needs for Process Improvement

The Organizations that produce software wants to improve their software development process for business competitiveness and profitability; this is achieved through:

- Improving product quality
- Improving team productivity
- Reduce product development cycle time

3.3 Approaches to SPI

When we are referring to SPI approaches we are talking about the way in which changes are applied in a software development organization [Aaen et al. 2001]

Over the past decade, a number of different approaches for implementing SPI have been proposed. CMM (Paulk et al. 1994), SPICE (Melo et al. 1998) and Bootstrap (Kuvaja et al. 1994), in particular, have shaped the way how SPI is perceived in practice.

This thesis will focus on 2 of the approaches that Pekka Abrahamsson analyzed in his study:

- Evolution approach
- Norm based approach

Evolutionary approaches

The evolutionary SPI approach is based on the idea that the changes in software engineering working processes should be evolutionary rather than revolutionary [Aaen *et al.* 2001].

This approach is based on the idea that the changes have to be the outcome of a sequence of small changes over the time, not a huge change involving dramatic and extreme consequences.

The evolutionary approach is advocated by a number of SPI authors (e.g., Basili & Green 1994, Johnson & Brodman 1996). Examples of this approach are the GQM method (van Solingen & Berghout 1999) and the CMM model (Isacsson et al. 2001).

These two methods (GQM and CMM) will be introduced in the following chapter.

The evolutionary approach to SPI provides several opportunities that are beneficial for SPI efforts: Practitioner involvement, experience based learning and on-going evaluation.

However, the evolutionary approach also involves certain risks, such as invisibility of low-level changes, and inability to anchor the process in daily practices.

Norm based approaches

The field of software process improvement has been largely dominated by the norm based approaches.

A set of norms that has been previously defined are applied to guide and control the results of a change process [Arent 2000]. Monitoring and comparing the results across companies and countries.

The most significant benefits for using the norm based SPI approaches are: to envision the future state clearly, to compare results across companies and countries, along with the use of benchmarking and the use of clear criteria for prioritizing improvement areas.

However, the use of this approach involves a number of risks: the strategy may be overly ambitious, the use of norm for its own sake is likely to distance the organization from its purpose, and it may be difficult to obtain reliable assessment results [Aaen et al. 2001].

4. Software process improvement models

To improve the capabilities of software organization it is necessary to understand the current status of the software development practices in the organization [Humphrey, 1989].

There are some approaches to implement SPI:

- CMM (Paulk et al., 1993)
- Bootstrap (Kuvaja et al., 1994)
- Capability Maturity Model Integration (CMMI) SEI (Chrissis *et al.*, 2003).
- IDEAL Model

All these 4 models present practical roadmap for improving organization's processes, and they also specifies a method for appraising current processes for identifying their strengths and weaknesses, and proposing recommendations for process improvements [Zahran, 1998].

In this chapter, there is a brief introduction about each model.

4.1 CMM Capability Maturity Model

The Capability Maturity Model (CMM) is a common-sense application of process management and quality improvement concepts to software development and maintenance [Paulk et al., 1993].

The CMM identifies and support five levels of maturity:

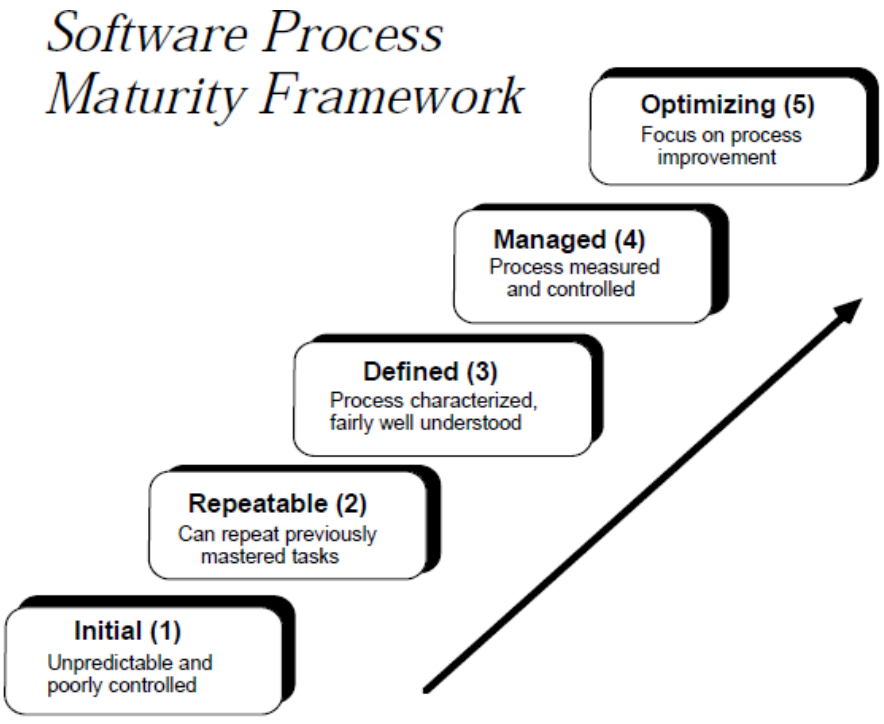

Figure 5: Software process Maturity framework [Paulk et al., 1993]

1. Initial.

At this point or the level, the organization does not have a steady development environment for software's. In addition, the organizations adopt adhoc and chaotic processes and developed products are over budget and behind schedule. Such organizations are not consistence with the decision making and success merely depends upon the individuals who are champion, and seasonal and adhoc teams. If the process manger, who is champion, leaves the organization then, the impact of his absence will create a serious project management problem and issue in the organization [Paulk et al., 1993].

2. Repeatable

Organization at level 2 has established policies, standards for managing the projects and sets the procedure for implementing them but processes may vary among different projects. Basic software management controls has been installed in the organization and new project planning and management is repeated on the based on the earlier project success and previous learning experience. Processes are stable and a project manager well in control of the budget and schedule. The communication of the identified problem is done appropriately as arise in the project [Paulk et al., 1993].

3. Defined

At this level of an organization, documented processes categorically are used across the organization for the development and maintenance of the software. These processes also integrate into coherent whole of the processes of Software engineering and management processes and CMM as "organization standard software processes". A software engineering process group (SEPG) has facilities for the organization process improvement efforts. To enhance the product quality peer reviews is in place and the Organization also launch training program to ensure that entire employee holds necessary skills that helps to perform their duties efficiently [Paulk et al., 1993].

4. Managed

At the managed level, Software products are of high quality and management sets quantitative and quality goals for product and process. There are well-defined consistence mechanism for evaluating process and product. Under Organizational measurement program, the productivity and quality of the key process activities of all the projects across the organization are measured. The organization also maintains the process database which

use, collect and analyze the data accessible for “projects defined processes” [Paulk et al., 1993].

5. Optimizing

At this level, the organization goals are to preventing defects and main focused on continuous process improvement. The organization has the channel to identify the weakness of the process and make them stronger to processes. A cost analysis of new technologies is performed, on the basis of process effectiveness and appropriateness changes and suggestion is provided to the process. The modernization that use best software engineering practices are identified and shifted all over the organization [Paulk et al., 1993].

The next table shows the SEI’s software process-maturity framewrok. The SEI derived this empirical model from the collective experiences of many software managers and practitioners.

Level	Focus	Characteristics	Key process Areas	Result
5 Optimizing	Continuous process improvement	• Improvement feedback into process • Data gathering is automated and used to identify weakest process elements • Numerical evidence used to justify application of technology to critical task • Rigorous defect – cause analyses and detect prevention	• Defect Prevention • Technology Change Management • Process Change Management	Productivity & Quality
4 Managed	Product and process quality	• Measured process • Minimum set of quality and productivity measurements established • Process database established with resources to analyze its data and maintain it	• Quantitative Process Management • Software Quality Management	
3 Defined	Engineering process and organizational support	• Process defined and institutionalized • Software Engineering Process Group established to lead process improvement	• Organization Process Focus • Organization Process Definition • Training Program • Integrated Software Management • Software Product Engineering • Intergroup Coordination • Peer Reviews	
2 Repeatable	Project management process	• Process dependent on individuals • Established basic project controls • Strength in doing similar work, but face major risk when presented with new challenges • Lacks orderly framework for improvement	• Requirements Management • Software Project Planning • Software Project Tracking & Oversight • Software Subcontract Management • Software Quality Assurance • Software Configuration Management	
1 Initial	-	• No formal procedures, cost estimates, project plans • No management mechanism to ensure procedures are followed, tools not well integrated, and change control is lax • Senior management does not understand key issues	-	Risk

Table 2: SEI’s software process-maturity framewrok [Humphrey et al, 1991] & [Paulk et al., 1993].

4.2 Bootstrap

The European counterpart to CMM is the Bootstrap assessment method, developed using the standard ISO 9000.

The original objective of the BOOTSTRAP project was to act as a lead-in and preparation project for the European System and Software Initiative (ESSI programme) and is funded by the Commission of European Countries (CEC). Therefore the BOOTSTRAP project was to lay the groundwork for introducing modern and adequate software technology into industry. The BOOTSTRAP methodology includes a guided assessment process, maturity and capability determination instruments (questionnaires and algorithm), guidelines for process improvement (standards for action plan generation), and assessor training program. [Kuvaja, 2011]

Although the BOOTSTRAP methodology was formed by extending the original SE1 model with new and reshaped features, it is still possible to distinguish the maturity levels that are equivalent to the SEI model.

The scale includes five capability stages known as maturity levels: initial level, repeatable level, defined level, managed level, and optimizing level. [Kuvaja, 2011]

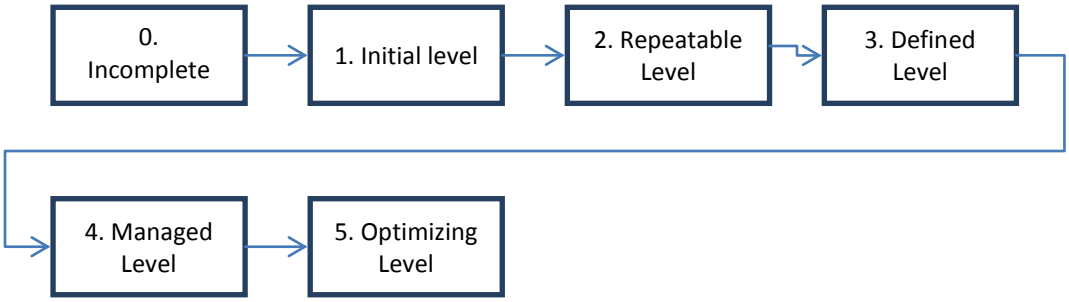

Figure 6: Bootstrap

The BOOTSTRAP methodology includes two separate questionnaires that are used as data gathering instruments and that support and guide the conduct of the interviews performed during the data collection. One of the questionnaires is intended for gathering data on the organization level and the other is focused on the project level [Kuvaja, 2011].

Bootstrap decouples the process model and the capability model. In Boostrap, the improvement philosophy states that improvement should be driven by organizational needs, whereas the process model provides an outline for the improvement of individual processes. Compared to SW-CMM, the organization maturity level is replaced by process capability profiles showing the capability level of each process.

4.3 Capability Maturity Model Integration (CMMI)

CMMI framework has different goals, best practices and process areas. The CMMI model consists of five levels with key process areas and four main categories.

Each process area has specific defined goals for the improvement. There are specific practices to achieve specific goals. Process areas are grouped into the following four main categories:

- Project Management: This category has process areas that are related to project management e.g. project planning, controlling, or integrating teams and supplier management.
- Support: This category has process areas that are related to process and product quality, configuration management and others.
- Process Management: This category has process areas that are related to organizational training, innovation, deployment and others.
- Engineering: This category has process areas that are related to requirements development, verification, validation, technical solution and others.

A suite of models developed by the SEI including the SW-CMM, the Systems Engineering Capability Maturity Model, and the Integrated Product Development Capability Maturity Model have recently been merged and extended into an integrated CMM called CMMI. The CMMI team's mission included the objective of ensuring that all of the products developed are consistent and compatible with ISO/IEC 15504 (CMMI Product Team 2002). The CMMI provides two views of capability: a staged view and a continuous view. The staged view gives five levels of evolution towards organizational maturity (initial, managed, defined, quantitatively managed, and optimizing). The continuous view provides six levels of process capability (incomplete, performed, managed, defined, quantitatively managed, and optimizing (CMMI Product Team 2002, p. 18-9).

CMMI best practices enable organizations to do the following:

- Link management and engineering activities more explicitly to business objectives
- Expand the scope of and visibility into the product lifecycle and engineering activities to ensure that the product or service meets customer expectations
- Incorporate lessons learned from additional areas of best practice (e.g., measurement, risk management, and supplier management)
- Implement more robust high-maturity practices
- Address additional organizational functions critical to its products and services

- More fully comply with relevant international standards such as ISO 9000 and ISO/IEC 15504 (Chrissis, Konrad & Shrum 2003; CMMI Product Team 2002).

4.4 IDEAL Model

In the Goldenson and Herbsleb (1995) study shows that little attention was paid to implement the SEI models (CMM, CMMI…) and standard effectively.

SEI proposed the framework called IDEAL Modal to support the implementation of software process improvement.

The IDEAL model is a life-cycle approach that can be used for SPI managers to manage and drive the SPI initiatives in an organization; it consists of five phases which provide structures for continuous improvement [McFeeley, 1996].

From the model derives its name –IDEAL- from the first letters of these phases. The five main phases are **I**nitiating, **D**iagnosing, **E**stablishing, **A**cting and **L**earning.

These five phases consist of 14 activities.

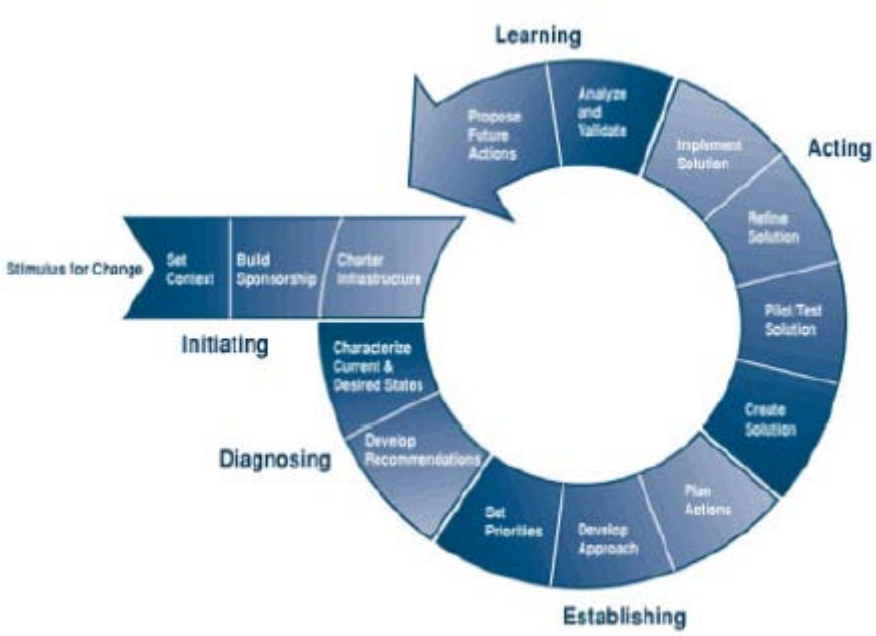

Figure 7: Ideal Model

Below are the purposes of each phase:

1. **The Initiating phase:**

The Initiating phase of the IDEAL model is the starting point. Here is where the initial improvement infrastructure is established, the roles and responsibilities for the infrastructure are initially defined, and initial resources are assigned. In this phase, a SPI plan is created to guide the organization through the completion of the Initiating, Diagnosing and Establishing phases. Approval for the SPI initiative is obtained along with a commitment of future resources for the job ahead [McFeeley, 1996].

2. Diagnosing Phase:

The Diagnosing phase of the IDEAL model starts the organization on the path of continuous software process improvement.

This phase lays the groundwork for the later phases. In this phase, the SPI action plan is initiated in accordance with the organization's vision, strategic business plan, lessons learned from past improvement efforts, key business issues faced by the organization, and long-range goals. Appraisal activities are performed to establish a baseline of the organization's current state. The results and recommendations from appraisals and any other baselining activities will be reconciled with existing and/or planned improvement efforts for inclusion into the SPI action plan[McFeeley, 1996].

3. Establishing Phase.

During the Establishing phase, the issues that the organization has decided to address with its improvement activities are prioritized; strategies for pursuing the solutions are also developed. The SPI action plan draft will be completed in accordance with the organization's vision, strategic business plan, lessons learned from past improvement efforts, key business issues facing the organization and long-range goals [McFeeley, 1996].

4. Acting Phase

In the Acting phase of the IDEAL model, solutions to address the areas for improvement discovered during the Diagnosing phase are created, piloted, and deployed throughout the organization. Plans will be developed to execute pilots to test and evaluate the new or improved processes.

After successful piloting of the new processes and determining their readiness for organization-wide adoption, deployment, and institutionalization, plans to accomplish the roll-out are then developed and executed [McFeeley, 1996].

5. Learning Phase

The objective of the Leveraging phase is to make the next pass through the IDEAL model more effective. By this time, solutions have been developed, lessons have been learned, and metrics on performance and goal achievement have been collected. These artifacts are added to the process database that will become a source of information for personnel involved in the next pass through the model.

Using this collected information, an evaluation of the strategy, methods and infrastructure used in the SPI program can be performed. By doing this, corrections or adjustments to the strategy, methods, or infrastructure can be made prior to the start [McFeeley, 1996].

5. Critical success factors (CSF)

5.1 Definitions

The first part of this section defines critical success factors taking account different author that have research about this field.

The concept of critical success factor (CSF) has been used in several fields. The general concept of CSF was introduced by Rockart [Rockart 1979] as a mechanism to identify the information needs of chief executive officers. CSFs are defined as those few key areas where things must go right for a business to grow. Since then it has become a widely used technique in a number of situations.

In the field of Knowledge Management **Digman** in 1990 defined CSFs as the areas where things must go right for the business to flourish. [Alazmi M., Zairi M. 2010]

In the same field **Oakland (1995)** defined them as what the organization must accomplish to achieve the mission by examination and categorization of the impacts. He adds that they are the minimum key factors or sub-goals that the organization must have or need, and which together will achieve the mission [Alazmi M., Zairi M. 2010].

It was not till 2001 that CSF have been identified in the SPI field, when **Somers and Nelson** said that *"Critical success factors can be viewed as situated exemplars that help extend the boundaries of process improvement, and whose effect is much richer if viewed within the context of their importance in each stage of the implementation process*" [Somers and Nelson, 2001].

Niazi wrote that "CSFs are a small number of important issues on which management should focus their attention" [Niazi, 2006]

Finally, in 2004 **Caralli** said that Critical success factors (CSFs) define key areas of performance that are essential for the organization to accomplish its mission. Managers implicitly know and consider these key areas when they set goals and as they direct operational activities and tasks that are important to achieving goals. However, when these key areas of performance are made explicit, they provide a common point of reference for the entire organization. Thus, any activity or initiative that the organization undertakes must ensure consistently high performance in these key areas; otherwise, the organization may not be able to achieve its goals and consequently may fail to accomplish its mission [Caralli, 2004].

A CSF is a key area of performance that is necessary to get in order to be effective and successful; this element must have a target goal and be measurable. It can also evolve over time, depending on the project phase.

Thus, the next step of this thesis is to find the 3 most important and influential CSF in the literature and get a correct definition and a method to measure and monitor them.

5.2 Identifying CSF

There are a lot of studies that have researched about the most important CSF in the software process implementation (SPI).

Studies and researches

This section will analyze how different authors have found the list of the most important CSF in SPI.

Researchers	
Goldenson and Herbselb's	1995
El-Emam	1998
Stelzer and Mellis	1998
Badoo and Hall	2002
Rainer and Hall	2002
Dyba	2005
Niazi	2006

Table 3: CSF Researchers

Goldenson and Herbselb's studied appraisals and process improvement efforts from a broad cross-section of software organizations. The sample includes software process assessments (SPAs) that were conducted in the United States and Canada during calendar years 1992 and 1993 – long enough ago for genuine change to have taken place (at least one year), yet recent enough to expect accurate recall from people familiar with the appraisals and their aftermaths (no more than three years).

They obtained information from 167 specific individuals, each of whom was in a good position to observe the aftermath of one of 61 appraisals. Using an intensive schedule of reminders and email, they received 138 completed questionnaires, which is 83 percent of the total number sent. They represent 56 of the 61 appraisals (92 percent) from which they sampled [Goldenson and Herbselb's, 1995].

El-Emam (1998) surveyed responses from 14 companies involves in the SPICE trials and made a reanalysis of Goldenson and Herbselb's study, using multivariate analysis instead of the simple statistical analytic methods used in the initial report. Based on this reanalysis, they identified the following key factors:

- Focused SPI effort
- Commitment to SPI
- Politics
- Respect
- Turnover

Stelzer and Mellis (1998) conducted a review of experience reports and case studies of 56 software organizations. In their study identified ten factors that influence the organizational change in SPI initiatives based on CMM or the ISO 9000 quality standards.

- Management commitment and support
- Staff involvement
- Providing enhanced understanding
- Tailoring improvement initiatives
- Managing the improvement project
- Change agents and opinion leaders
- Stabilizing changed processes
- Encouraging communication and collaboration
- Setting relevant and realistic objectives
- Unfreezing the organization

Badoo and Hall (2002) present empirical findings from their study of Software Process Improvement (SPI) motivators in 13 UK software companies. Their analysis aims to provide SPI managers with some insight into designing appropriate SPI implementation strategies to maximize practitioner support for SPI. They identify what motivates developers, project managers and senior managers to be actively involved in SPI. They characterize motivations according to classic motivation theory. They find that most motivators are specific to a particular practitioner group with only a few common to all practitioner groups. Most of the common motivators are `rewarding', according to classic motivation theory [Badoo and Hall, 2002].

Rainer and Hall (2002) explore 26 factors that potentially affect SPI. They consider the research strategies used to study these factors. They have used a multi-strategy approach for this study: first, by combining qualitative and quantitative analysis within case studies; second, by comparing their case study results with the results of a previously conducted survey study. Seven factors relevant to SPI were identified by the case studies and the survey study:

- Executive support
- Experienced staff
- Internal process ownership
- Metrics
- Procedures,
- Reviews
- Training

Two factors (reward schemes and estimating tools) were found, by both the case studies and the survey study, not to be relevant to SPI. Three additional factors (people, problems and change) were identified by the case studies. The frequency with which people, problems and change are discussed by practitioners suggests that these three factors may be pervasive in SPI, in a way that the other factors are not. These factors, however, require further investigation [Rainer and Hall, 2002].

Dyba (2005) presents the results from an empirical investigation of the key factors for success in SPI. A quantitative survey of 120 software organizations was designed to test the conceptual model and hypotheses of the study. The results indicate that success depends critically on six organizational factors:

- Business orientation
- Involved leadership
- Employee participation
- Concern for measurement
- Exploitation of existing knowledge
- Exploration of new knowledge

The main contribution of Dyba was to increase the understanding of the influence of organizational issues by empirically showing that they are at least as important as technology for succeeding with SPI and, thus, to provide researchers and practitioners with important new insights regarding the critical factors of success in SPI [Dyba, 2005].

Niazi (2006) conducted a comparative between an empirical study of the CSF for the SPI implementation with 34 SPI partitiones and a literature survey of CSF that impact SPI. The objective of that study was to provide SPI practitioners with sufficient knowledge about the nature of issues that play a positive role in the implementation of SPI programmes in order to assist them in effectively planning SPI implementation strategies. Through the empirical study they identified seven factors that were considered critical for successfully implementing SPI:

- Higher management support
- Training
- Awareness
- Allocation of resources
- Staff involvement
- Experienced staff
- Defined SPI implementation methodology

They also report on a literature survey of CSFs that impact SPI and identify six factors (senior management commitment, staff involvement, staff time and resources, training and mentoring, creating process action teams and reviews). They compared the empirical study results with the literature and confirmed the factors identified in the literature, and also identified two new CSFs (SPI awareness and defined SPI implementation methodology) that were not identified in the literature. Finally, they analyzed the CSFs identified by different groups of practitioners and found that they are aware of what is imperative for the successful implementation of SPI programmes [Niazi et. all, 2006].

5.3 CSFs identified through literature

After a wide review of the literature we are able to summaries in a table the most important CSF.

The following Table 4 shows the list of CSFs that I identified through the literature review. The most citied factor in the literature is senior management commitment (86%).

Other frequently cited factors in the literature are staff involvement, experienced staff and SPI awareness and implementation.

Categories	Goldenson and Herbslebs,(1995)	Stelzer and Melis ,(1999)	Rainer and Hall ,(2001)	El Emam et al ,(2001)	Badoo and Hall ,(2002)	Niazi et al ,(2006)	Dyba,(2006)	Frequency	Percentage (N=7)
Senior Managment Commitment	x	x	x	x		x	x	6	86%
Staff Involvement	x	x	x			x	x	5	71%
Training and mentoring	x		x			x	x	4	57%
Allocation of Resources		x		x	x	x		4	57%
Exprience Staff	x		x			x		3	43%
SPI goals and Objective	x	x					x	3	43%
Organization Politics				x		x	x	3	43%

Table 4: List of CSFs identified through the literature review

Senior Management Commitment

Senior management commitment is the most cited factors in the available literature (Goldenson and Herbslebs, 1995; Stelzer and Melis, 1999; Rainer and Hall, 2001; El Emam et al, 2001; Badoo and Hall, 2002; Niazi et al , 2006 ; Dyba, 2005), it's cited in the literature as an important CSF an 86%.

These researchers use different key words to define the "management commitment" term, for example, higher management commitment, executive support, top down commitment etc. The management commitment is the degree of interest for process improvement and the extent to which the resources make available for SPI by the management.

Staff Involvement

Staff involvement is among a key factor which helps to facilitate successful SPI program. This is agreed by many researchers such as: Goldenson and Herbslebs, 1995; Stelzer and Melis, 1999; Rainer and Hall, 2001; Niazi et al , 2006 ; Dyba, 2005. It's cited in the literature as an important CSF an 71%.

Dyba (2005) defined staff involvement factor as "the extent to which employees use their knowledge and experience to decide, act, and take responsibility for SPI and this is positively associated with SPI success" while, Stelzer and Melis (1999) defined staff involvement as "the degree to which staff members participate in the improvement activities".

Training and Mentoring

After a carefully revision of the CSF's literature it's possible to see that "staff training" is the third most cited CSF. Our study revel that it has been cited a 57%.

This is agreed by many researchers such as: Goldenson and Herbslebs, 1995; Rainer and Hall, 2001; El Emam et al, 2001; Badoo and Hall , 2002; Niazi et al , 2006 ; Dyba, 2005.

These studies reveal that training staff involve an improvement of the efficiency.

Allocation of Resources

According to El-Emam: management commitment can be determined by the degree to which management seem ready to make available the resources for SPI and it is considered one of the strong indicator of management commitment towards SPI [El Emam et al, 2001].

Also Stelzer and Melis, 1999 cited that senior management should have a broader picture of the resources and time required in order to conduct the SPI initiatives [Stelzer and Melis, 1999].

Experience Staff

More than 40% of the literature cited reviews, experienced staff, clear and relevant SPI goals and assigning of responsibilities as the CSFs

The experience and expertise play an important role in the successful implementation of SPI program.

Niazi cited in 2006: "When the organizations became more established they realized that experienced staff and training are an integral part of SPI implementation. This is because the organizations realized during the early period of SPI initiatives that their managers and employees have only a general idea of the SPI and do not have a complete understanding of the necessary details, and also that they do not understand how their work adds to the mission and vision of the organization" [Niazi, 2006].

SPI objectives and goals

It is important for the organizations to set realistic and relevant goals for SPI.

These objectives need to be crystal clear and, SPI managers need to communicate to all the actions groups within the organization. This CSF belongs to the “organizational” categories, so according to Niazi 2006, these factors are not critical anymore for SPI implementation.

Stelzer and Melis:

“Setting relevant objectives means that the improvement efforts attempt to contribute to the success of the organization. Setting realistic objectives means that the goals may be achieved in the foreseeable future and with a reasonable amount of resources.

It is essential that staff members understand the relationship between the objectives of software process improvement and revenues, cash flow, or other business results.

The real test of the improvement objectives is the degree to which everyone can make the translation from top management goals to the goals that each person is being asked to achieve” [Stelzer and Melis, 1999].

Organizational Politics

Some researchers consider politics as a factor or a barrier in the implementation of SPI.

> “There are many factors that can trigger organization politics, such as reallocation of the resources, promotions opportunities, low trust, times pressure, and role ambiguity.” [Niazi, 2009].

Figure 3 shows an example typical of these relationships.

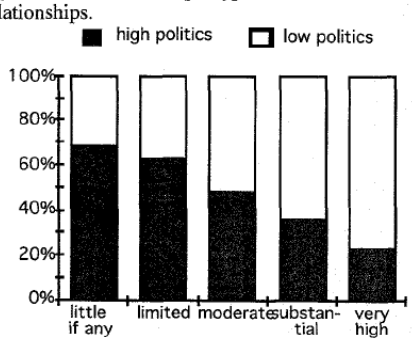

Figure 3. For each category of success with the SPI effort, the bars represent the percent of respondents who agreed ("high") or disagreed ("low") that their organizations are characterized by a high level of "organizational politics".

Figure 8: Organizational Politics [Goldenson, Herbsleb , 1996]

5.4 Relationship between CSF and SPI frameworks

Once the Critical success factors (CSF) had been found and defined and the different SPI frameworks had been clarify, it was pretend to find the relationship between them.

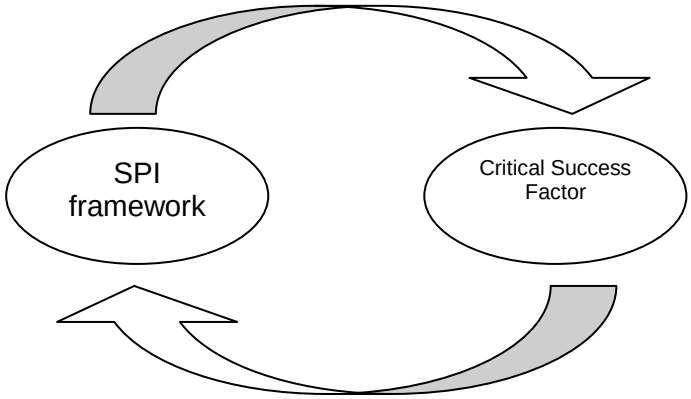

Figure 9: Relationship SPI framework - CSF

This relationship has been extracted from the review of the selected publications.

The intentions was to know which of the most cited CSF are important for the different framework

Categories	CMM	CMMI	Bootstrap	Spice	IDEAL	Mention without SPI framework	TOTAL (Max 6)
Senior Management Commitment	X	X	X	X	X	X	6
Staff Involvement	X	X	-	X	X	X	5
Training and mentoring	X	X	-	-	X	X	4
Allocation of Resources	X	-	-	-	X	X	3
Experience Staff	X	X	-	-	-	X	3
SPI goals and Objective	X	-	-	-	-	X	2
Organization Politics	-	-	-	-	-	X	1
Total (max 7)	6	4	1	2	4	7	

Table 5: Relationship SPI framework - CSF

In the following graph it s observed that the Senior Management Commitment is cited in every single SPI frameworks, and the following are Staff involvement and training.

All the factors are also related to “Mention without framework”, this is maybe because the relation between this two factors are not the center of attention of the researchers.

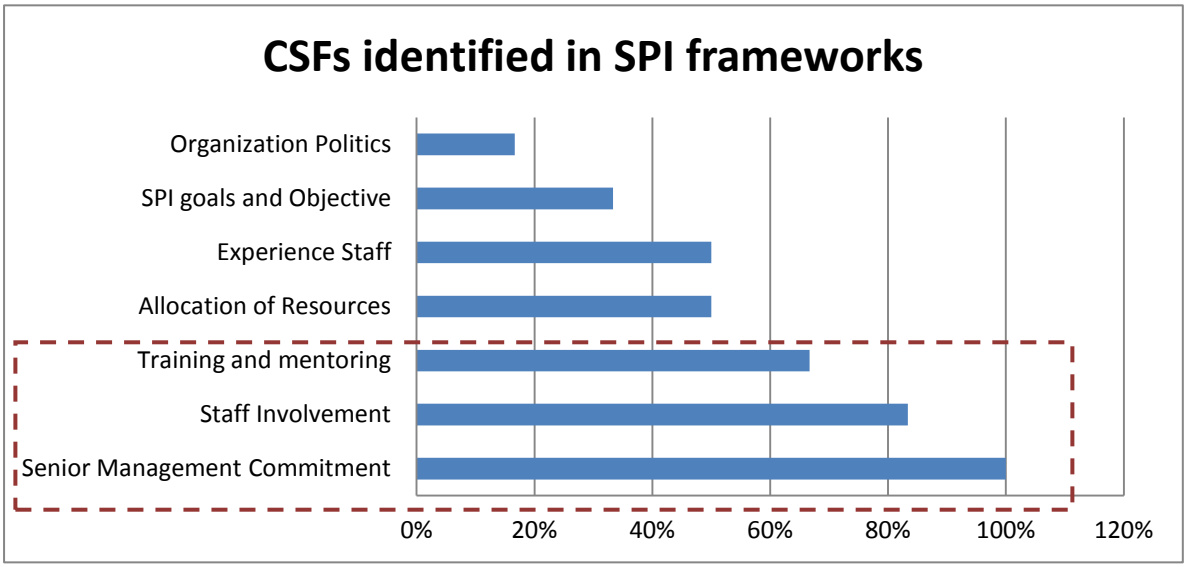

Figure 10: CSFs identified in SPI frameworks

Summary:

There is a lack of information related to this relationship. Most of the researches don’t explicitly related SPI success factors to SPI frameworks.

6. Top 3 CSF

Based on the outcome of the research study developed in the previous it is possible to come with the list of the 3 most important critical success factors in the software process improvement field.

The top 3 critical success factors are:

- Management commitment
- Staff Involvement
- Training

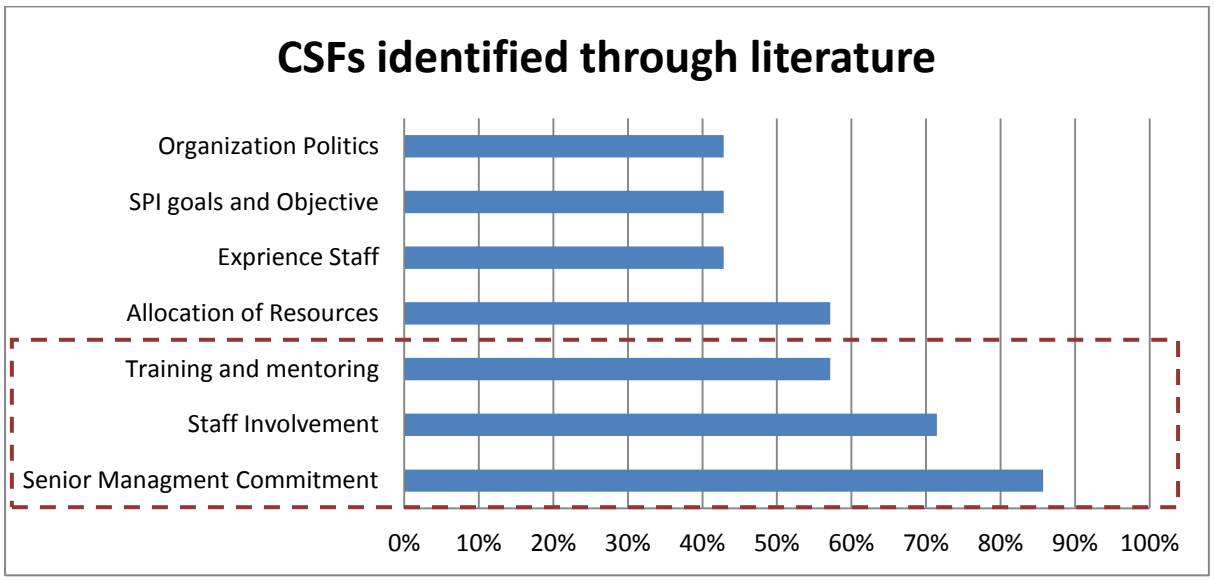

Figure 11: CSFs identified through literature

6.1 Management commitment

Literature is full with reports explaining the necessity to have a solid top management commitment in order to achieve successful results.

Humphrey argues that all major changes to the software process must start at the top and ultimate everyone must be involved in the change process. Statz identified management commitment as one of the potential risk factors in SPI

Comparing several studies “Management commitment” is the most frequently cited factors in the CSF. In Niazi research said that Management Commitment is cited in the literature about 70%, and in the Stelzer this percentage rise till the 91%.

There are many reasons why executive management needs to be continually committed and involved in implementation and maintenance of the business management system.

- The main reason is that if commitment isn’t evident to the employees then the system will never truly be implemented throughout the organization.
- Employees at all levels of an organization won’t be committed to a system that’s not driven, supported, and believed-in by upper management.

[Kelly B., 2008]

The research of Abrahamsson P., Jokela T. (2000) [Abrahamsson P., Jokela T. 2000] focus on Management Commitment as a success factor. They reported the results from a study aimed at developing a model of management commitment process and analyzed the existing models of management commitment and finally they have proposed a new model. [Abrahamsson, 2002]

The respondents of Goldenson and Herbsleb study tended to agree or strongly agree with the following: “managers actively monitor progress”
as it is possible to see in the following figure.
The bars represent the percent of respondents who agreed (“high”) or disagreed (“low”) that seniors management actively monitors SPI progress. [Goldenson and Herbsleb, 1996]

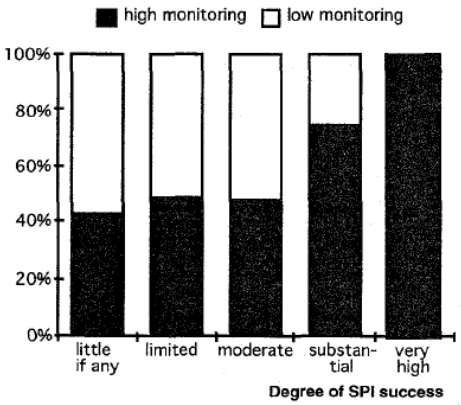

Figure 12: Degree of SPI success [Goldenson and Herbsleb, 1996]

Defining Management Commitment

Abrahamsson consider manager anyone who meets any of the following characteristics:

- Person(s) in the organization with the authority to fund the process improvement initiative
- Person(s) in the organization with the authority to provide resources for SPI
- Person(s) in the organization, with the authority to decide to what extent the SPI activities are carried out in respective software development projects.

Some definitions of commitment are:

- The Capability Maturity Model (CMM) defined commitment as “a pact that is freely assumed, visible, and expected to be kept by all parties” (CMU 1994).
- A more broad definition is given by O’Reilly and Chatman (1986). They view commitment as a psychological state of attachment that defines the relationship between a person and an entity. They also described commitment as the degree to which an individual internalizes or adopts the goals and values of the organization.

Stelzer & Mellis in 1999 defined Management commitment:

> “Management commitment and support is the degree to which management at all organizational levels sponsor the change” [Stelzer & Mellis, 1998].

Dyba in 2005 improve the definition as:

> “Management can also be defined as the extent to which leaders at all levels in the organization are genuinely committed to and actively participate in SPI [Dyba, 2005].”

"... senior management sponsorship proved critical to the success of the process improvement efforts. This means not only taking an active interest in the progress of various process improvement initiatives, but also providing funding and time to do the work, and rewarding those who contributed" [Diaz, Sligo 1997].

Management commitment is needed from all levels; commitment from upper management won’t be enough unless individual project leaders and managers are also determined to succeed. Managers must be convinced of process improvement’s value; it’s not free, but in the long run it more than pays for itself [Diaz, Sligo 1997].

Laporte and Trudel (1998) state in the lessons learned that “in addition to the senior management commitment it is essential that middle management and first line managers become strong supporters of the process improvement program’” [Niazi et al, 2006]

After carefully review of all studies, management commitment is defined as:

“The capacity to achieve the results of a whole team within a specific timetable and deadlines, developing personal interest in software process improvement.”

Management Commitment models

Pekka Abrahamsson realized a deeply research about the commitment management and a definition of this concept in the context of software process improvement. The elaboration of the concept is based on a literature study, which makes the research done in behavioral psychology and organizational science applicable in the field of software process improvement.

Abrahamsson conclusions:

Abrahamsson illustrated three misconceptions that existing models on commitment development are based on.

Assumptions that practitioners (the model developers) hold are:

1. The linearity of human cognitive process in the development of commitment,
2. The controllability of this process
3. The sole utility aspect of commitment phenomenon.

So, he elaborated a new model inspired by the writings of Winograd, Flores and Spinosa.

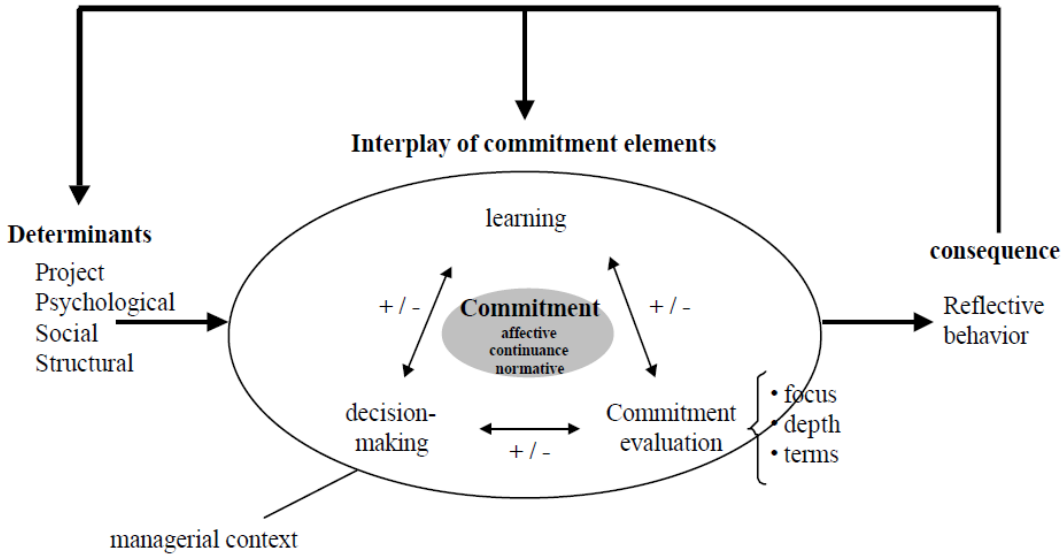

Figure 13: Management commitment Model

The model is divided in three parts:

1. Determinants affecting the managerial commitment process.
    - The projects determinants are objective attributes of the SPi projects: the cost and benefits.
    - Psychological determinants involve the key individual that participate in the project.
    - Social determinants originate from the group surrounding the individual
    - Structural ones are the conditions surrounding the project: the SPI infrastructure, administrative inertia…
2. Managerial commitment process as interplay of commitment elements.

    This second part is the core of the model, it is based on the analysis of exiting models.

    There are three processes:
    - Learning: Its purpose is to reduce the inherent information asymmetry between the supplier and the buyer.
    - Decision-making. This is the central process to all managerial levels. The more unstructured the commitment decision is, the more important role the trust plays.
    - Commitment evaluation. A person evaluates the commitment from time to time when triggered by a certain stimulus.
3. The consequence of the process. This is the visible action of the model
    - Reflective behavior: The last part of the model is the consequent of the process, a visible action.

6.2 Staff Involvement

Another important factor in the success or failure of any organization is the power of its people, and how well that power is focused towards meeting the organization's objectives. Comparing several studies "Staff Involvement" is the second most frequently cited factors in the CSF. In Niazi research said that Staff Involvement is cited in the literature about 70%, and in the Stelzer this percentage rise till the 84%.

This is agreed by many researchers such as (Dyba, 2005; Rainer and Hall, 2001; Goldenson and Herbslebs, 1995).

As seen in the following figure, most of the respondents report that, for the success of their organizations it is necessary that staff is actively involve

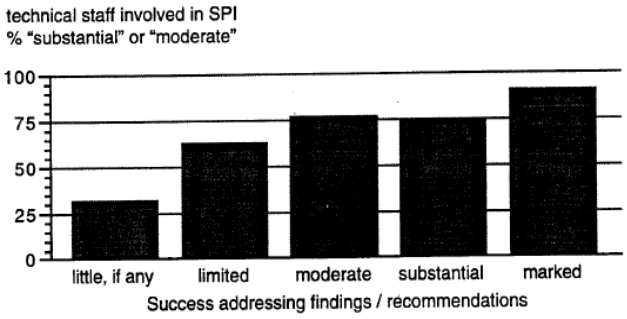

Figure 14: Success addressing findings [Goldenson and Herbslebs, 1996]

Defining Staff involvement

Dyba (2005) defined staff involvement factor as

> *"The extent to which employees use their knowledge and experience to decide, act, and take responsibility for SPI and this is positively associated with SPI success".*

while, Stelzer and Melis (1999) defined staff involvement as

> "T*he degree to which staff members participate in the improvement activities*."

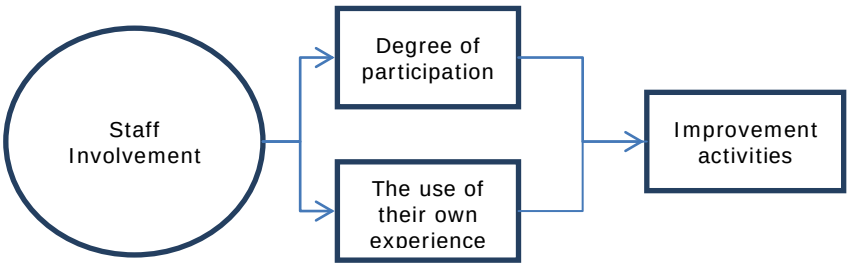

Figure 15: Staff Involvement

So, following the definitions of above researchers it can be said that:

“Staff involvement is the degree of participation of the staff, while they are adopting responsibility in SPI initiative, using their own experience and skills for successful implementation of Software Process Improvement”.

In the last years, the companies have realized the better the people have been treated the best results are obtained.

Staff involvement also means employee participation, as Basri in 2011 said, employee participation is the strongest influence on Software Process Improvement (SPI) success and, in general, peoples are the main factor in software process improvement that needs to be encouraged and support in an organization.

Furthermore, lack of people involvement in development activities will disturb the improvement process. Hence the aim of process improvement will be fail if people are not commit to all the propose change activities. In addition the strengths and weaknesses of the current process are inside the staff hands and knowledge. Hence, that even though people are the main driver for software quality but the processes has been given more attention [Basri, et al. 2011].

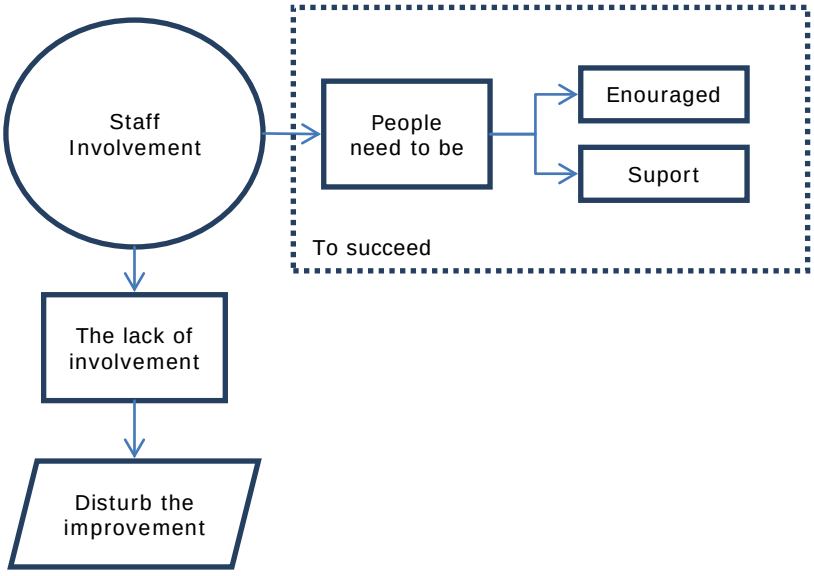

Figure 16: Staff involvement

Stelzer and Melis (1999) stated that

> “to ensure grass root staff involvement successful implementation initiative have established local process team, special interest groups, training scheme, forum for the exchange of ideas and for coordinating effort among project team” [Stelzer and Melis, 1999]

Some of the organizations are not able to understand the integrated approach and split the development project and the process improvement activities. While staff member should need to be involved in the improvement initiative as they used these processes in daily job routine and hence they have better understanding and strong knowledge of weakness and strength of current processes [Stelzer and Melis, 1999].

When the organizations became more established they realized that experienced staff and training are an integral part of SPI implementation. This is because the organizations realized during the early period of SPI initiatives that their managers and employees have only a general idea of the SPI and do not have a complete understanding of the necessary details, and also that they do not understand how their work adds to the mission and vision of the organization.

The lesson to be learned from this is that to implement SPI at least as effectively as their large counterparts, small software organizations should capitalize on their relative strengths in employee participation and exploration of new knowledge. This implies that small software organizations require learning strategies that are closely aligned with explorative behavior, while at the same time promoting the exploitation of past experience. Thus, to be successful, our findings suggest that formal processes must be supplemented with informal, inter-personal coordination about practice. [Dyba, 2005]

6.3 Training

A deep analysis of the CSF's literature reveals that "training" concept has been used in great amount of papers on the topic of SPI implementation.
The importance of training factor is recognized by different authors:

Niazi
Dyba
Rainar and Hall
Goldenson and Herbslebs

As Rainer and Hall (2002) said, training factor is having a major impact on SPI programmes [Rainer and Hall, 2002].

In order for SPI program to succeed, it is essential that staffs who are involved in process improvement initiatives should be trained with necessary skills and competences.

Rainer and Hall (2001) cited Paulk et al, who stated that

> "… *the most effective transfer occurred with the reassignment of people possessing the dynamic knowledge about how to apply mature processes and improvement methods*" [Rainer and Hall, 2001].

In order for SPI program to succeed, it is essential that staffs who are involved in process improvement initiatives should be trained with necessary skills and competences. Organization need to have a must-continued training and evaluation program to keep track of the training of the personals that sponsor the change [Rainer and Hall, 2001].

All the software development staffs involved in process maturity improvement need to provide training according to their roles.
The large or young organization cannot provide this type of detail training to all the staffs and processes as it's too expensive so, instead, they should select an individual staffs and provide the specific processes training that s/he will execute [Guerrero, 2004].
Software process methods need specialized training for their implementation. These trainings help to transfer methods to organization so that it is accepted and become part of practice by the organization. For the software engineering staffs it is important to have general training to SPI that helps in addressing the environmental issues. This leads organization towards a "Quality culture" such as that of Siemens [Mehner et al., 1998].
Software organization face serious problem maintaining their capabilities in terms of both efficiency and flexibility. Dyba defined two broader categories and concepts of learning strategies i.e. "exploitation" and "exploration".

"Exploitation involves improving existing capabilities by refining, standardizing, routinizing, and elaborating established ideas, paradigms, technologies, strategies, and knowledge. In contrast, exploration involves learning through discovery and experimenting with ideas, paradigms, technologies, strategies, and knowledge in hope of finding new alternatives and untapped opportunities that are superior to current practice" [Dyba, 2005].

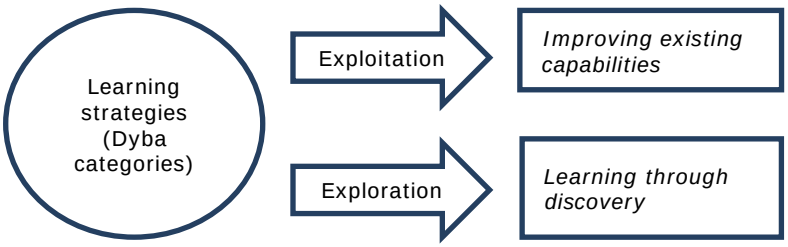

Figure 17: Training

Between 1987 and 1990 Hughes aircraft progressed for CMM maturity level 2 to CMM maturity levels - 3 based on SEI process maturity 1-5. It was conducted in collaboration with SEI (Software Engineering Institute).

Humphrey et al, 1991 outline the assessment method used and shows the result of the study.

During assessments teams found that Hughes had a well sponsor and comprehensive training program but certain training categories were either not available or were not used. Key examples were: training for assistant project managers, review leaders, and requirements specification.

The team recommended a revision of its software-training requirements.

Hughes and SEI selected the team members and SEI trained them for assessment methods. This program was for two-day duration and held at SEI. SED also maintain training records database to record the status of each employee at the time of performance appraisals. Also, form a committee that will periodically review the training requirement and effectiveness at Hughes [Humphrey et al, 1991].

7. Measurement

Basili:

> *"As with any engineering discipline, software development requires a measurement mechanism for feedback and evaluation. Measurement is a mechanism for creating a corporate memory and an aid in answering a variety of questions associated with the enactment of any software process"* [Basili et. Al, 2011]

7.1 Introduction to software measurement

Software measurement

The quality of the software can be determined by applying software measurements [Kitchenham, Pfleeger, 1996]. Software measurements provide a quantification of software quality, not just in terms of the software as a product but also by the performed process and spent resources for producing the software [Zuse, 1998]. Software measurement provides better control and visibility in the software development process and in the resulting product, and thus can be helpful in decision making [Costello, Liu, 1994]. The implementation of process improvement requires software measurement since if the results (whether in the development process or in the final product) of the process change are not measured, it is difficult to conclude that the improvement initiatives address the right issue [Zharan, 1998].

Categorization of software measurement

A categorization of software measurements can be done in several ways [Zuse, 1998]. Software measurements can be grouped into three categories based on the measured entities [Fenton, Pfleeger 1998]:

- *Product measures* are measurements that are collected from the software product, e.g. number of defects after release.
- *Process measures* are measurements that are collected from the methods, activities and practices used in developing a software product, e.g. the number of defects found during testing.
- *Resources measures* are measurements that are collected from the time, cost, effort, personnel or other kinds of resources used in the activities for developing a software product, e.g. the effort in man-months expended in the coding phase.

7.2 Measurement frameworks

Software development needs a measurement mechanism for feedback and for it own evaluation.

Measurement is a mechanism to answer a variety of questions associated with the development of any software process.

Using the results of the measurement is possible to determine the strengths and the weakness of the current process, it is also possible to evaluate the quality of specific process, and assess to evaluate the impact of particular actions.

Mendonça et al. define a measurement framework as a

"set of related metrics, data collection mechanisms, and data uses inside a software organization" [Mendonça et al, 1998].

Measurement must be based on goals and models. There are a variety of frameworks available for measurement and defining measurable goals. The Goal-Question-Metric (GQM) approach, Quality Function Deployment (QFD), Practical Software Measurement (PSM) and the Balanced Scorecard (BSC) approach are few of the major well-known frameworks.

These measurement frameworks are discussed briefly in the following sections.

GQM (Goal, Question, Metric)

Measurement, in order to be effective must be:

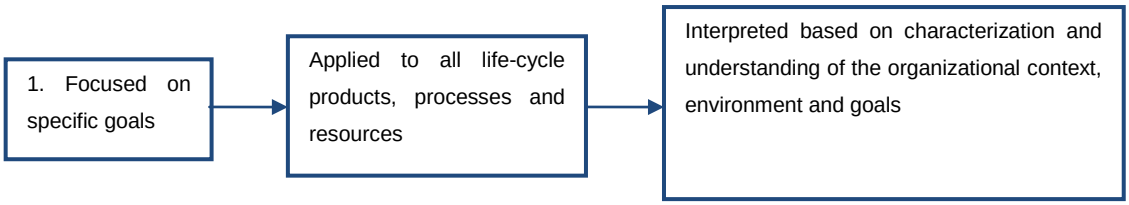

Figure 18: GQM

GQM [Basili, et al. 1994] is a method to guide the definition and exploitation of a goal-driven measurement program.

The Goal Question Metric (GQM) approach is based upon the assumption that for an organization to measure in a purposeful way it must first specify the goals for itself and its projects, then it must trace those goals to the data that are intended to define those goals operationally, and finally provide a framework for interpreting the data with respect to the stated goals. Thus it is important to make clear, at least in general terms, what informational needs the organization has, so that these needs for information can be quantified whenever

possible, and the quantified information can be analyzed to whether or not the goals are achieved [Basili, et al. 1994].

Figure 19: Steps GQM

The resulting measurement model has three levels:

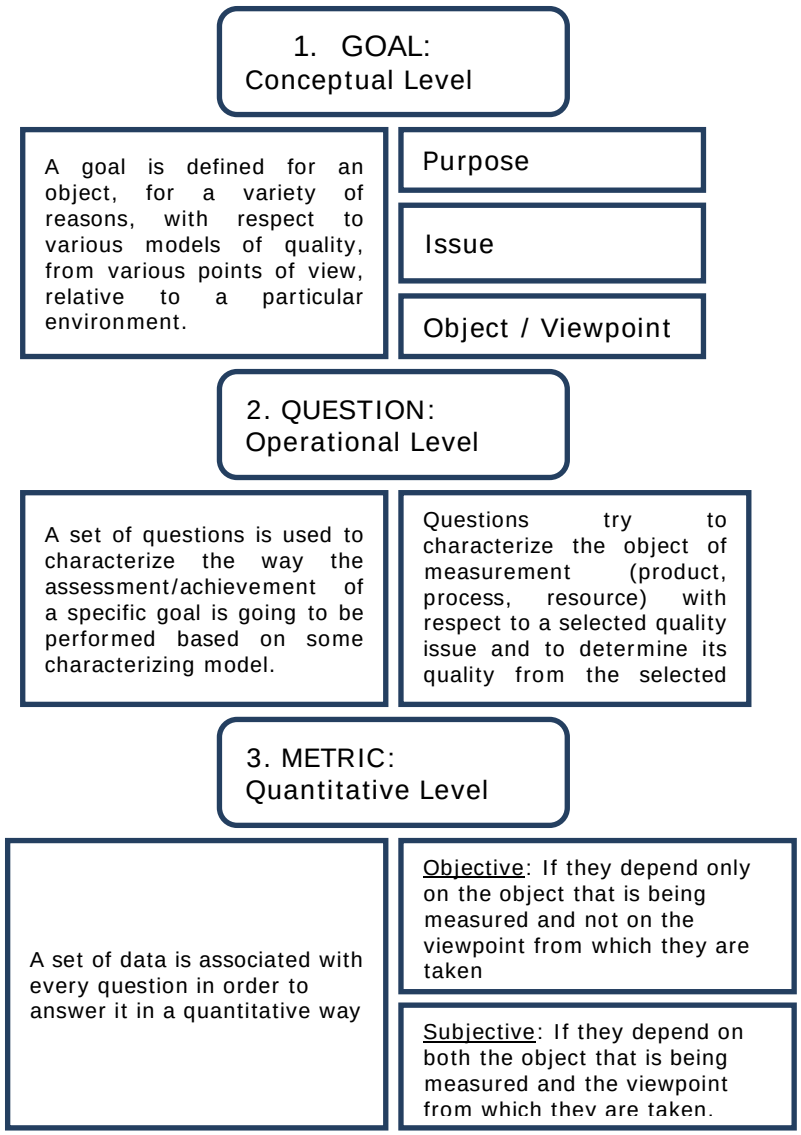

Figure 20: Levels GQM [Basili, et al. 1994]

A GQM model is a hierarchical structure:

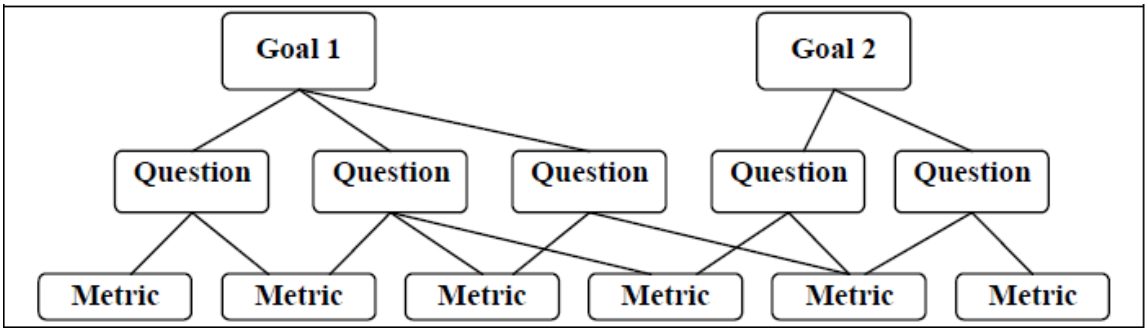

Figure 21: Structure GQM [Basili, et al. 1994]

Quality Function Deployment (QFD)

The QFD approach was conceived in the late 1960s [Akao, Mazur, 2003] and was mostly used in hardware development [West, 1991]. Software companies started employing this methodology in software development starting from the late 1980s [West, 1991]. QFD provides methods to ensure that the true customer needs are realized in design, development and delivery of a new product and helps to improve the product development process itself [Akao, Mazur, 2003]. QFD includes a series of matrices that are linked together which enables the requirements to be traced throughout the whole development process [West, 1991]. Analysis of these matrices helps to develop customer requirements, design requirements, test requirements, process requirements etc. [West, 1991] .

The most commonly known matrix of QFD is the "House of Quality (HOQ)" [Karlsoon, 1997] . The HOQ provides a conceptual map to the development organization to reason about the customer and user requirements so that the most important customer needs are identified and developers' best efforts are ensured to address these needs that eventually maximize the end-user satisfaction [Karlsoon, 1997].

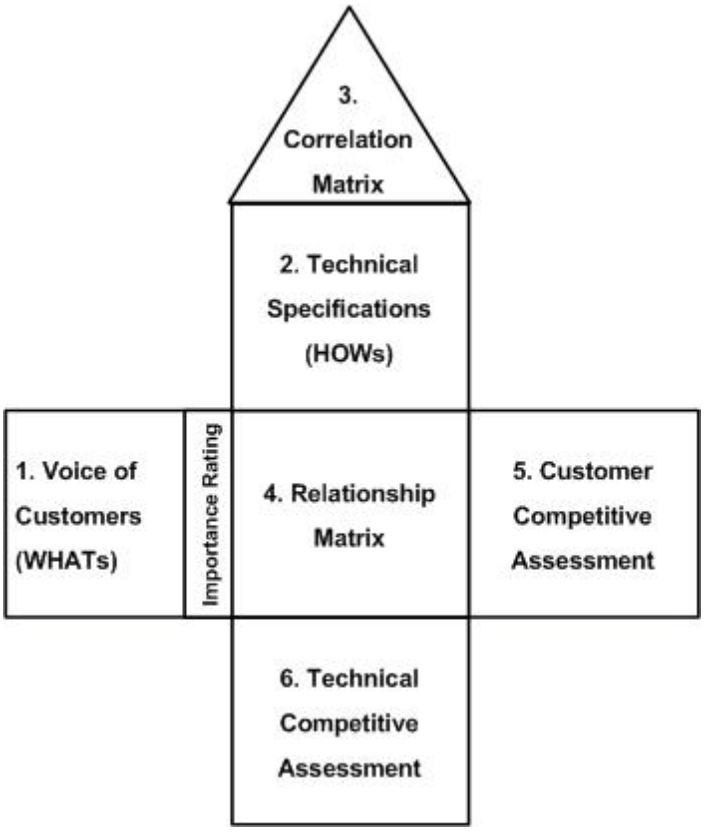

Figure 22: Simplified structure of "House of Quality" is illustrated inspired by [Liu, 2000] and [Buglione, Abran, 2000] .

The “House of Quality” in QFD consists of six main parts [Liu, 2000], as shown in Figure 22, and presents basically the intersection of two dimensions [Buglione, Abran, 2000] . The horizontal dimension located in Part 1 refers to the WHATs (also known as "Voice of the customer”) that identify the characteristics of the product desired by the customer while the vertical dimension located in Part 2 refers to the HOWs, that is the way (e.g. technical requirements) identified by the QFD team to achieve the WHATs [Buglione, Abran, 2000] . The QFD team consists of people from different parts of the organization (e.g. marketing, design, project management, QA, development, etc.) [64]. Part 3 specifies the trade-offs in the technical specification of the product identified in Part 2. The relationship matrix, Part 4, located in the central part of the “House of Quality”, represents the strength of the relationship between WHATs and HOWs, that is, it correlates what customers want from the product and how the company can meet those requirements. Part 5 describes how customers and users perceive the competitors' systems' abilities in meeting the requirements (the WHATs). In Part 6, the technical feasibility is assessed and a technical competitive analysis is conducted by designers and developers with respect to the perceived competitors' systems' abilities in meeting the HOWs. By integrating all those parts mentioned above in the “House of Quality”, customer's voice can be incorporated throughout all the development activities providing the traceability between development activities and customer requirements [Liu, 2000].

Balanced Scorecard (BSC)

BSC is a strategic planning and management system that helps to guide organizations to transfer abilities and specific knowledge held by people throughout the organization in order to achieve long-term strategic goals [Kaplan, Norton, 1996]. It provides a comprehensive framework for executives in an organization to translate company visions and strategies into a set of coherent performance measures [Kaplan, Norton, 1996].

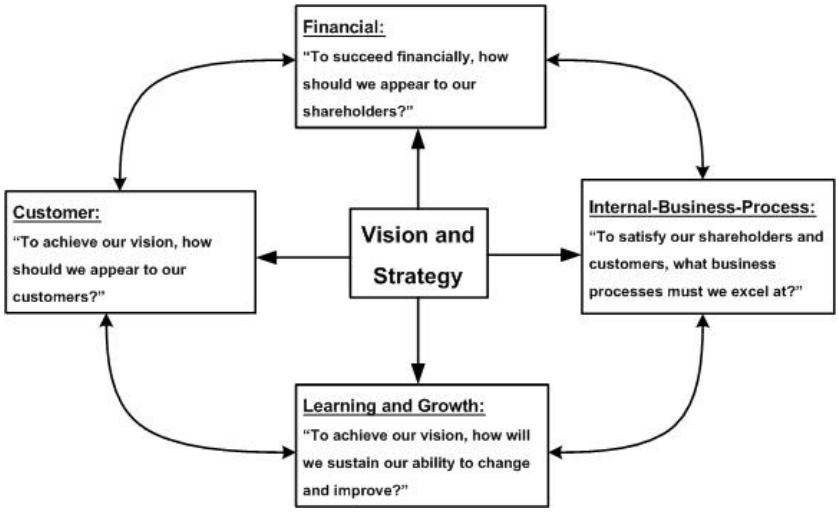

Figure 23: The Balanced Scorecard (inspired by [Kaplan, Norton, 1996])

The BSC proposes to view the organization from four perspectives (Figure 23) [Kaplan, Norton, 1996]:

- *Financial perspective:* It is of crucial importance to measure economic consequences of actions taken in the organization. The core measures include operating income, return-on-investment, sales growth and generation of cash flow.

- *Customer perspective:* This perspective measures the customer and market segment performance. The typical measures are customer satisfaction, customer retention, new customer acquisition, customer profitability and market share.

- *Internal-business-process perspective:* The internal-business-process perspective measures focus on internal processes that have impact on customer satisfaction and achieving an organization's financial objectives. Generic measures include quality, time and cost.

- *Learning and growth perspective:* This perspective identifies the infrastructure that the organization must build to create a long-term growth and improvement. The measures principally come from three sources: people, systems and organizational procedures.

8. Measuring CSF

After the analysis of these 3 methods to measure software, in this thesis we will apply the GQM due to it is the most appropriate, it is more flexible and we can measure abstract concepts.

Based on this method, first of all, it is necessary to define a Goal. This Goal should be converted into several questions, with these questions the goal is divided into different issues. Each question is redefined into metric. This metric can be used in order to answer the questions.

So, first of all it is necessary to identify the goals for each CSF of our study.

8.1 Goals:

There are templates available to support the definitions of measurement goal by specifying purpose (what object and way), Issue (what aspect and who) and context characteristics [Basili, et al. 1994].

Management Commitment Goals

MAIN GOAL	
Purpose	Monitor
Issue	Commitment
Object	Management
Viewpoint	From the project manager's viewpoint

Table 6: Management Commitment Goals

SUBGOALS – MANAGEMENT COMMITMENT					
1	Purpose	Control	**4**	Purpose	Giving
	Issue	Active Participation		Issue	formal and informal recognition
	Object	Management		Object	
2	Purpose	Observer	**5**	Purpose	Provide
	Issue	Visible Support		Issue	Resources and training
	Object	Management		Object	
3	Purpose	Monitor	**6**	Purpose	Communicate
	Issue	Help		Issue	Company evolution
	Object	To solve problems		Object	To all the staff

Table 7: Management Commitment SubGoals

Staff involvement Goals

MAIN GOAL	
Purpose	Monitor
Issue	Involvement
Object	Staff

Table 8: Staff involvement Goals

SUBGOALS – STAFF INVOLVEMENT					
1	Purpose	Perceived	4	Purpose	Intention
	Issue	Group support		Issue	Quit the Job
	Object	Staff		Object	Staff
2	Purpose	Satisfaction	5	Purpose	Degree
	Issue	Kind of job		Issue	Willingness to Cooperate
	Object	Staff		Object	Staff
3	Purpose	Participation	6	Purpose	Degree
	Issue	Decision Making		Issue	Task independence
	Object	staff		Object	Staff

Table 9: Staff involvement SubGoals [Dow Scott , et al. 2003]

Training Goals

MAIN GOAL	
Purpose	Monitor
Issue	Actual training
Object	Employee

Table 10: Training Goals

SUBGOALS – TRAINING					
1	Purpose	Structure the training	4	Purpose	Follow-up
	Issue	Support with materials		Issue	Training
	Object	Trainer		Object	Company
2	Purpose	Assistance	5	Purpose	Monitoring
	Issue	Training		Issue	The results
	Object	Employee		Object	Company
3	Purpose	Active Participation	6	Purpose	Resources
	Issue	Classes		Issue	Enough resources spend
	Object	Employee		Object	Company

Table 11: Training SubGoals

8.2 Question

It has been designed a questionnaire in order to evaluate and measure these goals.

The questions are based on the 59 indicator that Dyba [Dyba, 2000] found in his study by asking to 11 experts, it has also been add some other indicators based on the experience of the other authors.

It has been design two types of questions, in order to measure the qualitative and the quantitative data.

8.3 Metrics

There are two kinds of questions, Quantitative and Qualitative.

Qualitative Questions

A subjective rating scale has accompanied each question.

According to Dyba investigation, if too few scale points are used, the answer scale is obviously coarse, and some of the information can be lost due to the scale dos not capture the "little details". On the contrary, using too many scale points, the scale can become too finely that is can beyond the respondent's ability [Dyba, 2000].

In the same research, it has been cited some authors (Lissitz and Green 1975, Likert and Roslow 1934, Van de Ven and Ferry 1980) that concluded that the optimum answers question is a 5 point scale, where responses score from 1 to 5, with the value 1 indicating "strongly disagree" and 5 indicating "Strongly agree".

	Strongly disagree	**Disagree**	**Neither agree nor disagree**	**Agree**	**Strongly agree**
Management is visible supporting SPI activities	□	□	□	□	□

Table 12: Qualitative question/answer

Quantitative Questions

Quantitative Questions	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
How much influence do you have on how you perform your job?	□	□	□	□	□	□	□	□	□	□	□

Table 13: Quantitative question/answer

Questionnaire

Management commitment indicators:

As it was said by Stelzer and Mellis:

"Senior managers should actively participate in assessment meetings and improvement workshops to demonstrate the importance of the initiative. Active participation and visible support of senior management may give the necessary momentum to the initiative" [Stelzer & Mellis, 1998].

Also, Diaz & Sligo in their study in 1997 realizes that it was necessary weekly meetings of workings groups to address process, technology and people issues. Moreover it is also necessary meetings of chief software engineers and new-projects staff. [Diaz, Sligo 1997].

Thus, a very important issue is the number of meetings done in a project. This is one of the indicators is necessary to measure.

All the following question are based on the knowledge of the most important authors in the field (Goldenson and Herbslebs, 1995; Stelzer and Melis, 1999; Rainer and Hall, 2001; El Emam et al, 2001; Badoo and Hall, 2002; Niazi et al , 2006 ; Dyba, 2005

Qualitative Questions	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
Manager is visible supporting SPI activities	☐	☐	☐	☐	☐
Manager accepts responsibility for SPI	☐	☐	☐	☐	☐
Management considers SPI as a way to increase competitive advantage	☐	☐	☐	☐	☐
Management is actively participating in SPI activities	☐	☐	☐	☐	☐
There are enough meetings in order to control SPI	☐	☐	☐	☐	☐
The meetings have clear objectives	☐	☐	☐	☐	☐
There are enough presentations in order to	☐	☐	☐	☐	☐

Table 14: Management commitment indicators

Quantitative Questions	0	1	2	3	4	5	6	7	8	9	10
Control active participation											
How many presentations are given	□	□	□	□	□	□	□	□	□	□	□
How many meetings (for each project)	□	□	□	□	□	□	□	□	□	□	□
Visible support											
How many times per week does the manager influence do you have on how you perform your job?	□	□	□	□	□	□	□	□	□	□	□

Table 15: Management commitment Quantitative Questions

Staff involvement indicators:

All the following question are based on the knowledge of the most important authors in the field (Goldenson and Herbslebs, 1995; Rainer and Hall, 2001; El Emam et al, 2001; Badoo and Hall , 2002; Niazi et al , 2006 ; Dyba, 2005.)

Qualitative Question	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
Main Goal					
The employee is contributing with improvements proposals	☐	☐	☐	☐	☐
The employee participate in the formalization of routines	☐	☐	☐	☐	☐
The employee participate in the dialogue and discussion about software development	☐	☐	☐	☐	☐
The employee take on responsibility on SPI	☐	☐	☐	☐	☐
The employee participate in order to get the SPI goals.	☐	☐	☐	☐	☐
Perceived Group Support					
The group values the contribution of each employee to it	□	□	□	□	□
The group cares about the general satisfaction at work of each employee.	□	□	□	□	□
Help is available from the work group when any employee has a problem	□	□	□	□	□
Willingness to Cooperate					
I am willing to share information with other employees about work	□	□	□	□	□
I am willing to cooperate with other employees to get the work done	□	□	□	□	□
Cooperative problem solving is more effective than individual problem solving	□	□	□	□	□
Task Interdependence					
I work closely with others in doing my work.	□	□	□	□	□
I frequently must coordinate my efforts with others	□	□	□	□	□
My work requires me to consult with others fairly frequently.	□	□	□	□	□

Table 16: Staff involvement qualitative questions

Quantitative Questions	0	1	2	3	4	5	6	7	8	9	10
Job Satisfaction. – How satisfied are you with:											
Your job in general.	□	□	□	□	□	□	□	□	□	□	□
Your working conditions	□	□	□	□	□	□	□	□	□	□	□
The opportunity to use your skills and abilities	□	□	□	□	□	□	□	□	□	□	□
Participation in Decision Making											
How much influence do you have on how you perform your job?	□	□	□	□	□	□	□	□	□	□	□
How much influence do you have on what does on in your work group?	□	□	□	□	□	□	□	□	□	□	□
My supervisors are receptive and listen to my ideas and suggestions.	□	□	□	□	□	□	□	□	□	□	□
Intention to Quit											
It is very possible for me to leave for another company next year	□	□	□	□	□	□	□	□	□	□	□
I often think of quitting my current job.	□	□	□	□	□	□	□	□	□	□	□
I plan to stay with this company for a long time to advance my career	□	□	□	□	□	□	□	□	□	□	□

Table 17: Staff involvement Quantitative Questions

Training indicators:

All the following question are based on the knowledge of the most important authors in the field (Goldenson and Herbslebs, 1995; Rainer and Hall, 2001; El Emam et al, 2001; Badoo and Hall , 2002; Niazi et al , 2006 ; Dyba, 2005.)

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
The company has a must-continued training	☐	☐	☐	☐	☐
There is training according with the role of each employee	☐	☐	☐	☐	☐
The employee is attending to the training courses	☐	☐	☐	☐	☐
The employee participate actively in the training	☐	☐	☐	☐	☐

Table 18: Training qualitative questions

Quantitative Questions	0	1	2	3	4	5	6	7	8	9	10
The training give the skills and competences it was design for	□	□	□	□	□	□	□	□	□	□	□
The objectives of the course were achieved	□	□	□	□	□	□	□	□	□	□	□
Trainers											
Trainers have sufficient knowledge	□	□	□	□	□	□	□	□	□	□	□
Trainers communicate well	□	□	□	□	□	□	□	□	□	□	□
The difficulty level of this workshop was appropriate.	□	□	□	□	□	□	□	□	□	□	□
Time management											
Enough time was devoted to each module	□	□	□	□	□	□	□	□	□	□	□
Enough time was given for feedback from the participants	□	□	□	□	□	□	□	□	□	□	□
Evaluation											
General evaluation of the Course											
Grade of the final exam											

Table 19: Training quantitative questions

9. Results and conclusions

9.1 Results

SPI not only enable to improve the product quality within reasonable costs; additionally, the productivity of the organization is improved and the visibility of the process achieved which helps management to evaluate the results of their efforts of the SPI that aid to take appropriately timely measures ignored by the above authors.

CSF

Senior management commitment plays a vital role and is one of the most important success factors of organizational changes in SPI efforts.

In Niazi et.al study, the management commitment factor is citied 68% in their survey study. Their study further reported that their developers and mangers always looking for senior management support for implementation of SPI programs this is because of their past SPI experienced that realized the important of this factor.

Staff involvement is an important factor and no SPI programs can make successful if this factor is not addressed well. This factor ranked second position and was citied 88%. Their result shows that sr. managers wants to involve the staffs in SPI because they belief that without staffs involvement SPI cannot be made successful and developers also want to participate in the change programs because they like to participate and are eager to involve themselves in the hope of improving their working style and chance of training opportunity.

Training and mentoring are important factors that cannot be ignored and this factor was ranked at third position and was citied 50 %. In Niazi et al study, this factor was citied 68%. Their results show that developer and manager consider training as essential part of SPI and without training of change agents in respective KPAs, the success cannot be guaranteed..

9.2 Conclusions and future work

This study investigates the factors that have positive impact in a process of SPI implementation and offer recommendations to practitioners that promote the best practices in the software process improvements domain.

In particular I focused on critical success factors. A comprehensive literature review has been conducted to identify CSFs that citied most of the researchers.

The findings indicates that there are seven CSFs: (1) senior management commitment, (2) staff involvement, (3) experienced staff, (4) SPI goals and objectives, (5) training and mentoring ,(6) allocation of resources and (7) change management that are critical for SPI implementation.

This study has based on the 3 most cited CSF of the literature review that are: (1) senior management commitment, (2) staff involvement, (3) training.

During the research phase I found a lack of formal definitions in publications related to the CSF; therefore there were some explanations in relevant context of SPI success factors. With the help of these explanations, I was able to create definitions for the most important SPI success factors.

Software development needs a measurement mechanism for feedback and also for its own evaluation. Measurement is a mechanism to answer a variety of questions associated with the development of any software process. Using the results of the measurement is possible to determine the strengths and the weakness of the current process, it is also possible to evaluate the quality of specific process, and assess to evaluate the impact of particular actions.

After the analysis of the 3 methods to measure software, in this thesis I have applied the GQM due to it is the most appropriate, it is more flexible and we can measure abstract concepts.

9.3 Future work

The following are future directions that can complement the work I did here:
I would like to discover the SPI obstacles in order to see their effects on the failure of the SPI initiative. It is also important to identify failure factors of SPI in order to understand the underlying meaning of the SPI success factors.
This thesis has been based on the frequency of occurrence of the CSF in the publications I would also like to find success factors with respect to their directly application. However, it is an extensive effort but it can give very promising results. By finding the directly application and comparing with real results, it is possible to figure out more concretely which factors are the most important with respect to their scenario and study context.

10. List of Figures and Tables

11. References

[Aaen et al. 2001] Aaen I, Arent J, Mathiassen L & Ngwenyama O (2001) A conceptual map of software process improvement. Scandinavian Journal of Information Systems 13: 123-146.

[Arent 2000] Arent J (2000) Normative software process improvement. Deparment of computer science. Aalborg, Aalborg University.

[Basili, 1993] Basili, V.R. 1993. Applying the Goal Question Metric paradigm in the Experience Factory. In the 10th Annual CSR Workshop, October 1993.

[Basili, et al. 1994] Basili, V.R., Caldiera G. & Rombach. H. D. 1994 Goal Question Metric Paradigm. Encyclopaedia of Software Engineering. Vol. 1 ed. John Wiley & Sons.

[Bicego et. al, 1998] Bicego, A., Khirana, M., Kuvaja, P. & Lehtonen, J. 1998. Spice conformant assessment method for embedded systems, the Profes Project report 4.2.B-I. Version 1.1. September 9, 1998

[Costello, Liu,1994] Rita Costello and Dar-Biau Liu, “Metrics for Software and Systems Requirements Engineering for Large System Acquisition and Development,” AOWSM (Annual Oregon Workshop on Software Metrics), Silver Fall State Park, Oregon: 1994.

[Diaz, Sligo 1997] Diaz, Michael; Sligo, Joseph 1997. How Software Process Improvement helped Motorola. IEEE Software, 14, No. 5.

[Dow Scott , et al. 2003] Dow Scott, James W. Bishop, Xiangming Chen, 2003. An examination of the relationship of employee involvement with job satisfaction, employee cooperation, and intention to quit in u.s. invested enterprise in china. International Journal of Organizational Analysis, Vol. 11 Iss: 1,

[Dyba, 2000] T. Dybå, "Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context," New York, NY 10036-5701, United States, 2003

[Dyba, 2000a] T. Dybå, "Instrument for measuring the key factors of success in software process improvement," *Empirical Software Engineering,* vol. 5, 2000.

[Dyba, 2005] T. Dybå, "An empirical investigation of the key factors for success in software process improvement," *Software Engineering, IEEE Transactions on,* vol. 31, 2005.

[Fenton, Pfleeger 1998] N.E. Fenton and S.L. Pfleeger, Software Metrics: A Rigorous and Practical Approach, Course Technology, 1998.

[Goldenson, Herbsleb , 1995] Goldenson, Herbsleb, J.D., "After the Appraisal: A Systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success," Software Engineering Institute, Pittsburgh CMU/SEI-95-TR-009, 1995.

[Goldenson, Herbsleb , 1996] Goldenson, Herbsleb, J.D., "A systematic survey of CMM experience and results," in *Proceedings of the 18th international conference on Software engineering* Berlin, Germany: IEEE Computer Society, 1996.

[Karlsoon, 1997] J. Karlsson, “Managing software requirements using quality function deployment,” Software Quality Journal, vol. 6, 1997.

[K. El-Emam, et al, 2002] K. El-Emam, D. Goldenson, J. McCurley, and J. Herbsleb, "Modelling the likelihood of software process improvement: an exploratory study," *Empirical Software Engineering,* vol. 6, 2001.

[Kaplan, Norton, 1996] R.S. Kaplan and D.P. Norton, The balanced scorecard, Harvard Business Press, 1996.

[Kitchenham, Pfleeger, 1996] B. Kitchenham and S. Pfleeger, “Software quality: the elusive target,” Software, IEEE, vol. 13, 1996.

[Kitchenham et al, 2001] B.A. Kitchenham, R.T. Hughes, and S.G. Linkman, “Modeling Software Measurement Data,” IEEE Transactions on Software Engineering, vol. 27, 2001.

[Kitchenham et al, 2002] B.A. Kitchenham, S.L. Pfleeger, L.M. Pickard, P.W. Jones, D.C. Hoaglin, K. El Emam, and J. Rosenberg, “Preliminary guidelines for empirical research in software engineering,” IEEE Transactions on Software Engineering, vol. 28, 2002.

[Laird, Brennan, 2006] L.M. Laird and M.C. Brennan, Software measurement and estimation, John Wiley and Sons, 2006.

[Liu, 2000] X.F. Liu, “Software quality function deployment,” IEEE Potentials, vol. 19, 2000.

[McFeeley, 1996] McFeeley, B. (1996). *IDEAL: A User’s Guide for Software Process Improvement*. SEI Handbook CMU/SEI-96-HB-001, Carnegie-Mellon University.

[Mendonça et al, 1998] M.G. Mendonça, V.R. Basili, I.S. Bhandari, and J. Dawson, “An approach to improving existing measurement frameworks,” IBM Systems Journal, vol. 37, Nov. 1998.

N. B. Moe and T. Dyba, "Improving by involving: a case study in a small software company," Berlin, Germany, 2006.

[Niazi et al, 2003] M. Niazi, D. Wilson, and D. Zowghi, "A model for the implementation of software process improvement: a pilot study," Los Alamitos, CA, USA, 2003.

[Niazi et al, 2006] Niazi, D. Wilson, and D. Zowghi, "Critical success factors for software process improvement implementation: An empirical study," *Software Process Improvement and Practice,*.

[Niazi et al, 2007] M. Niazi, D. Wilson, and D. Zowghi, "Organisational readiness and software process improvement," Heidelberg, D-69121, Germany, 2007.

[Niazi et al 2008] M. Niazi, M. A. Babar, and N. M. Katugampola, "Demotivators of software process improvement: An empirical investigation," *Software Process Improvement and Practice,* vol. 13, 2008.

[Pikkarainen, Mäntyniemi 2006], Pikkarainen and Mäntyniemi 2006. An Approach for Using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies. SPICE 2006 conference, Luxemburg.

[Stelzer and W. Mellis , 1996] Stelzer, W. Mellis, and G. Herzwurm, "Software process improvement via ISO 9000? Results of two surveys among European software houses," in *System Sciences, 1996., Proceedings of the Twenty-Ninth Hawaii International Conference on*, 1996.

[Stelzer and W. Mellis , 1998] Stelzer and W. Mellis, "Success factors of organizational change in software process improvement," *Software Process Improvement and Practice,*1998

SEI (2002). *Capability Maturity Model® Integration (CMMISM)*, Version 1.1. Software Engineering Institute Technical Report CMU/SEI-2002-TR-029, Carnegie-Mellon University.

[Zuse, 1998] H. Zuse, A framework of software measurement, Walter de Gruyter, 1998.

[Zharan, 1998] Sami Zahran, Software process improvement: practical guidelines for business success, Year of Publication: 1998, Publisher, Addison-Wesley Longman Ltd. Essex, UK, UK.

[West, 1991] M. West, “Quality function deployment in software development,” IEE Colloquium on Tools and Techniques for Maintaining Traceability During Design, 1991,

Appendix 1: CSF Literature research (Extension)

Niazi

Niazi [M. Niazi, et al., 2006] carried out a research through the literature and he compared it with an empirical study to identify different CSF for software process improvement (SPI).

Success factors – Identified through **literature**	Success factors – Identified through **empirical study**
Senior management commitment	Senior management commitment
Staff involvement	Training
Training and mentoring	Awareness
Staff time and resources	Allocation of resources
Creating process action teams/ change agents and opinion leader	Experienced staff
Reviews	Defined SPI implementation methodology
Experiences staff	Staff involvement
Clear and relevant SPI goals	Facilitation
Assignment of responsibility for SPI	Communication
Process ownership	Project management
Encouraging communication and collaboration / sharing best practices	Quality assurance
Tailoring improvement initiatives	Formal documentation
Reward schemes	Reviews
Managing the SPI project	Automated tools
Providing enhanced understanding	Company culture
Internal leadership	Customer satisfaction
SPI people highly / well respected	External implementation agents
Standards and procedures	Logical sequence or order SPI implementation
-	Measurement
-	Tailoring improvement initiatives
-	Formalized relationship between development teams
-	Higher staff moral

Goldenson and Herbselb's

Four factors are associated with successful SPI efforts:

- The people involved in process improvement have been well respected in their software organizations
- There has been more involvement of technical staff in the SPI effort.
- The amount of staff time and resources dedicated to process improvement has been good or excellent since their appraisals
- Process improvement goals are clearly stated and well understood in their organizations

Three other factors are comparably associated with less successful SPI efforts. Similar results exist for:

- Discouragement and cynicism from previous experience
- The feeling among the technical staff that process improvement gets in the
- way of their "real" work

Stelzer and Mellis

Success Factor	ISO cases (n = 25)		CMM cases (n = 31)		All cases (n = 56)	
	Percentage	Ran	Percentage	Rank	Percentag	Rank
Management commitment and support	84%	1	97%	1	91%	1
Staff involvement	84%	1	84%	8	84%	2
Providing enhanced understanding	72%	3	87%	6	80%	3
Tailoring improvement initiatives	68%	4	90%	3	80%	3
Managing the improvement project	56%	6	94%	2	77%	5
Change agents and opinion leaders	52%	7	90%	3	73%	6
Stabilizing changed processes	52%	7	90%	3	73%	6
Encouraging communication and collaboration	64%	5	74%	9	70%	8
Setting relevant and realistic objectives	44%	9	87%	6	68%	9
Unfreezing the organization	24%	10	52%	10	39%	10

Rainer

Rainer [Rainer A, Hall T., 2002] focus his study in 3 researches (Emam, Stelzer and Mellis, and Goldenson and Herbsleb) who investigated standards that are recognized internationally.

In that study it was distinguished the more successful CSF, there were some similarities between the findings of the three studies. These factors are summarized in Table XX

	Goldenson	El Emam	Stelzer
Senior management commitment	Yes	Yes	Yes
Clear & relevant SPI goals	Yes	Yes	Yes
Clear, compensated assignment of responsibility for SPI	Yes	Yes	-
Staff involvement	Yes	Yes	Yes
SPI people highly/well respected	Yes	Yes	-
Staff time and resource.	Yes	Yes	-
Creating process action teams	-	Yes	-
Change agents and opinion leaders	-	-	Yes
Encouraging communication and collaboration	-	-	Yes
Managing the SPI project	-	-	Yes
Providing enhanced understanding	-	-	Yes
Stabilizing changed processes	-	-	Yes
Tailoring improvement initiatives	-	-	Yes
Unfreezing the organization	-	-	Yes

In Rainer study they identified 16 factors for further investigation:

Rainer – Critical Success Factors
Reviews
Training and mentoring
Standards and procedures
Internal leadership
Experienced staff
Inspections
Executive support
Internal process ownership

Metrics
Risk assessment
Automation
Project post mortems
Estimating tools
Reward schemes
External consultants
Stringent control

Toni M. Somers

[Toni M. Somers, 2001]

Critical success factor	Definition
Top management support	
Project team competence	
Interdepartmental cooperation	
Clear goals and objectives	
Project management	
Interdepartmental communication	
Management of expectations	
Project champion	
Vendor support	
Careful package selection	
Data analysis & conversion	
Dedicated resources	
Use of steering committee	
User training on software	
Education on new business processes	
Business Process Reengineering	
Minimal customization	
Architecture choices	
Change management	
Partnership with vendor	
Use of vendors' tools	
Use of consultants	