FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

A Gap Analysis Methodology for the Team Software Process

Luís Manuel Gonzalez Amaral

FINAL VERSION

Dissertation Report
Integrated Master in Informatics and Computing Engineering

Advisor: João Pascoal Faria (Aux. Professor)

June 2009

A Gap Analysis Methodology for the Team Software Process

Luís Manuel Gonzalez Amaral

Dissertation Report
Integrated Master in Informatics and Computation Engineering

Approved in oral examination by the committee:

Chair: José Faria (Prof. Dr.)

External examiner: Alberto Silva (Prof Dr.)

Internal Examiner: João Pascoal Faria (Prof. Dr.)

July 15th, 2009

Resumo

Ao longo dos anos a qualidade do software tem-se tornado cada vez mais importante na engenharia do software. Tal como noutras disciplinas da engenharia onde a qualidade é já um dado adquirido, a engenharia do software está em movimento nessa direcção.

O Team Software Process (TSP) é uma metodologia criada pelo Software Engineering Institute (SEI) cujo principal objectivo é ajudar os engenheiros de software e as equipas a assegurarem a qualidade, a elevada *performance*, o estabelecimento de objectivos e a gestão dos processos numa organização.

Esta dissertação apresenta uma metodologia para avaliar as diferenças (*gaps*) entre as práticas correntes da organização e as práticas do TSP de forma a ser possível avaliar os ganhos futuros e as necessidades que a organização vai ter durante e após a implementação do TSP.

A metodologia de *gap analysis* tem dois pilares fundamentais em termos de recolha de informação: as entrevistas e a análise de documentação. Foram desenvolvidos questionários para guiar a equipa avaliadora na tarefa de condução das entrevistas e ajuda suplementar foi desenvolvida baseada no conhecimento e experiência de trabalho em diversas organizações em qualidade de software. Finalmente um protótipo de relatório também é apresentado neste documento onde um exemplo do seu preenchimento é detalhado.

A metodologia desenvolvida foi aplicada numa conhecida empresa Portuguesa com o apoio e validação do SEI. Foi baseada nos modelos de engenharia de software mais conhecidos CMMI e ISO/EIC 15504 que são hoje standards no processo de qualidade de software. O objectivo desta metodologia é ser rápida e barata quando comparada com os modelos referidos ou com o projecto piloto de avaliação do TSP promovido pelo SEI. Após a primeira utilização da metodologia houve grande satisfação de todas as partes envolvidas. A aplicação da metodologia foi também utilizada na melhoria do processo através das lições aprendidas desta experiência.

Abstract

Over the years software quality is becoming more and more important in software engineering. Likewise other engineering disciplines where quality is already a commodity, software engineering is moving into these stages.

The Team software Process (TSP) is a technology created by the Software Engineering Institute (SEI) with the main objective of helping software engineers and teams to ensure quality, high performance, establish goals and improve both personal and process management in the organization.

This dissertation presents a methodology for assessing an organization against the TSP practices so that it is possible to assess the future gains and needs an organization will have during and after the implementation of TSP.

The gap analysis methodology has two pillars in terms of data collection: interviews and documentation analysis. Questionnaires have been developed to guide the team on the task of conduction interviews and further guidance has been developed in what and where to look for information in an organization. A model for the rating has also been developed based on the knowledge and experience of working in several organizations on software quality. Finally a report prototype is also presented in this document where an example of its filling is included.

The methodology developed was applied in one well known Portuguese organization with the support and validation of SEI. It is based on the most know software engineering models CMMI and ISO/EIC 15504 that are nowadays standards for software process quality. The objective of this methodology is to be fast and inexpensive when compared with the models stated or with the SEI TSP assessment pilot. After the first use of the methodology there was great satisfaction of all the parties involved. The application of the methodology has also been used to improve the process with the lessons learnt from this experience.

Acknowledgements

A special thanks to...

… my parents for their mentoring and support

… my wife, Marta, for her patience and dedication

… my advisor and co-worker, João Pascoal Faria, for his guidance and teaching

… and my co-workers at FEUP CIQS, Pedro Castro Henriques, Pedro Gomes and Raul Vidal for the great sharing of experiences.

Index

1 Introduction....1
1.1 Context and motivation....1
1.2 Objectives....1
1.3 Method....2
1.4 Structure....2

2 State of the art....4
2.1 What is the TSP....4
2.1.1 Context....4
2.1.2 Project life cycle....5
2.1.3 The TSP launch and re-launch processes....6
2.1.4 The development phase or cycle....7
2.1.5 The phase, cycle or project post mortem....7
2.1.6 Quality management....7
2.1.7 Measurement....8
2.1.8 Benefits and performance results....9
2.1.9 How to introduce TSP in an organization....10
2.1.9.1 Roadmap....10
2.1.9.2 Criteria for choosing people and projects....10
2.1.9.3 Synergies with other process improvement initiatives....11
2.1.9.4 Tools....11
2.2 TSP based certification....11
2.3 CMMI and ISO/EIC 15504 and their evaluation methods....14
2.3.1 CMMI....14
2.3.2 ISO/IEC 15504....15
2.3.3 SCAMPI....16
2.3.3.1 Context....16
2.3.3.2 Objectives and characteristics....17
2.3.3.3 Usage....17
2.3.3.4 Performance....18
2.3.3.5 Verification and discovery....18
2.3.3.6 Types of evidences....19
2.3.3.7 Investigation....19
2.3.3.8 Practice Implementation Indicators (PII)....20
2.3.3.9 Rating process....20
2.3.3.10 Practices characterization....21
2.3.3.11 Effective team practices....22
2.3.4 ISO/EIC 15504 assessment method....23
2.4 Conclusion....24

3 TSP gap analysis methodology....26
3.1 Method overview....26

3.2 Activities....28
3.2.1 Prepare....28
3.2.2 Gather information....29
3.2.3 Confirm information....30
3.2.4 Produce and present report....32
3.3 Success factors....32
3.4 Constraints....33
3.5 Risks....33
3.6 Assessment approach....33
3.7 The team....34
3.8 Analysis perspectives....34
3.9 Questionnaires....35
3.9.1 General guidelines for the interview....36
3.9.2 Project Manager....36
3.9.2.1 Open questions....36
3.9.2.2 Closed questions....36
3.9.3 Tester....38
3.9.3.1 Open questions....38
3.9.3.2 Closed questions....38
3.9.4 Developer....40
3.9.4.1 Open questions....40
3.9.4.2 Closed questions....40
3.10 Report template and example (RTE)....42
3.10.1 Report index....42
3.10.2 Gap analysis (RTE Section 6)....43
3.10.2.1 Analysis Perspectives....43
3.10.2.2 Detailed results....44
3.10.2.3 Teamwork....44
3.10.2.4 Measurement....45
3.10.2.5 Quality Management....47
3.10.2.6 Project Management....48
3.10.2.7 Engineering....50
3.10.2.8 Verification & Validation....51
3.10.2.9 Process Management....53
3.10.3 Performance Analysis (RTE section 7)....54
3.10.3.1 Data received....54
3.10.3.2 Analysis and Benchmarking....55

4 Practical implementation....56

5 Conclusions and future work....57
5.1 Fulfillment of objectives....57
5.2 Future work....58

References....59

List of Figures

Figure 2.1 - TSP Life Cycle [OVER2008].....5
Figure 2.2 - Launch process [HUMP2006b].....6
Figure 2.3 - Roadmap for TSP implementation.....10
Figure 2.4 - Performance profile [HUMP2008].....13
Figure 2.5 - Fidelity Profile [HUMP2008].....13
Figure 2.6- Coverage profile [HUMP2008].....13
Figure 2.7 - Organization profile [HUMP2008].....14
Figure 2.8 - SCAMPI rating process [SCAMPI2006].....21
Figure 2.9 - Example of ISO/EIC 15504 capability profile [CAPPRO2009].....24
Figure 3.1 - Plan for preparation of the gap analysis.....27
Figure 3.2- Effort Evaluation - The lower values in the graph show higher gaps.....44
Figure 3.3 - Teamwork Evaluation.....45
Figure 3.4 - Measurement Framework Evaluation.....47
Figure 3.5 - Quality Management Evaluation.....48
Figure 3.6 - Project Management Evaluation.....50
Figure 3.7 - Engineering Evaluation.....51
Figure 3.8 - Verification and validation Evaluation.....53
Figure 3.9 - Process Management Evaluation.....54

List of tables

Table 2.1 - Microsoft IT project survey [SMITH2008].....9
Table 3.1 - Typical schedule TSP introductory course.....28
Table 3.2 - List of artifacts to look for in the QMS.....29
Table 3.3 - Analysis perspectives.....35
Table 3.4 - Gap Analysis Perspectives.....43
Table 3.5 - Analysis and benchmarking.....55

Abbreviations and Symbols

ARC	The Appraisal Requirements for CMMI v1.2
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
EIC	International Electrotechnical Commission
ESA	European Software Agency
FTE	Full-time equivalent
ISO	International Organization for Standardization
IT	Information Technology
KLOC	One thousand lines of code used to measure the size of a software program
KSLOC	One thousand source lines of code used to measure the size of a software program
ODC	Orthogonal Defect Classification
PII	Practice Implementation Indicators
PIP	Process improvement Proposal
PROBE	Proxy Based Estimation
PSP	Personal Software Process
QMS	Quality Management System
SCAMPI	Standard CMMI Appraisal Method for Process Improvement
SDP	Software Development Process
SEI	Software Engineering Institute
SPICE	Software Process Improvement and Capability Determination
TSP	Team Software Process

1 Introduction

1.1 Context and motivation

In the context of software engineering, software quality is the degree to which a system, component, or process meets specified requirements as well as customer or user needs or expectations [IEEE1990]. Nowadays software quality is becoming increasingly more important as software engineering is becoming more and more an exact science where errors are less and less tolerated. Software engineering is moving towards the direction of other engineering disciplines, likewise civil engineering where bridges are standing and in use for more than 2000 years. The Team Software Process (TSP) methodology was developed based on these premises.

After the release of TSP, it has been implemented in a wide range of organizations world wide. Over recent years its growth has increased specially in the USA, Mexico and Asia among other countries.

It is a fact that TSP contributes greatly to the improvement of team and personal performance as well as the quality of the products developed. These facts make TSP contributing to a high cost reduction in the organizations making them more efficient and effective. TSP has also been used very broadly by organizations that want to reach the higher levels of CMMI (levels 4 and 5). CMMI is today the most known and used benchmark model regarding software process quality.

Due to these facts, it is becoming more and more important for organizations to know how far they are from TSP practices not only because of the cost of implementing TSP, but also to evaluate the gains that they can have when applying this methodology when compared with their current practices.

1.2 Objectives

The project described in this dissertation has the main objective of defining a methodology to help in the evaluation of the current practices of an organization that is considering implementing TSP by means of a gap analysis.

Being TSP a technology that brings big gains in terms of quality, productivity and predictability in projects, leading to cost savings, it is important to create an easy to use methodology to evaluate the distance of an organization practices to TSP.

The methodology should be easy to use and understand by someone that is familiar with software engineering and it should be a methodology that is quick to follow and produce results. The results should be fast to read and there must be material produced that is easy to read by senior management of organizations.

It is one of the objectives that this methodology can be applied both from the inside of the organization, meaning that an internal assessment team can follow the steps provided or from the outside of the organization, where an external entity is hired to perform the assessment and provide the results.

It is also objective of this work to show results of a case study of the methodology.

1.3 Method

The method for producing this work started with the analysis of the TSP methodology together with the state-of-the art methodologies for process assessment in software engineering: The Standard CMMI Appraisal Method for Process Improvement (SCAMPI) and the ISO/EIC 15504 assessment framework. Part of this study includes the investigation of articles about these 3 main topics.

After this initial research the methodology of assessment started to be built. It was decided to produce as part of this methodology, not only a planning and executing framework but also questionnaires to help the assessment team to perform their job and a skeleton of the final report to be delivered to the organization's management.

The following steps after the method was developed and reviewed were the real life test in an organization. A company that was interested in TSP has been approached in order to test the methodology and the needed steps described in it have been followed and improved based on the experience of this implementation.

After this the final adjustments have been done as well as this dissertation has been written and reviewed.

1.4 Structure

The current dissertation is organized in 5 chapters. In the first one an introduction to this work is done including the objectives of this dissertation.

In the second chapter the themes of TSP, SCAMPI and ISO/EIC 15504 are presented, including its advantages and best practices. This chapter details the methodology for implementing TSP as well as its success factors. Regarding the assessment methods CMMI and ISO/EIC 15504, the way they should be performed is described as well.

In chapter three the methodology of assessing organizations according to TSP is described. In this chapter, besides the method used to do this assessment, it is also included a proposed plan for the assessment, questionnaires to help the assessment team and examples of parts of the final assessment report.

In the next chapter, the forth one, a short description of the practical application of this methodology is described. No further information about this methodology evaluation can be shown due to confidentiality reasons.

In the fifth chapter the conclusions of this work are described. There are also references to the future work beyond this dissertation.

2 State of the art

In this section it is described the TSP methodology that is a key point of this dissertation. This chapter also describes the two most common methodologies used to assess the software development processes in an organization. They are based on the most know models for process improvement in software organizations: CMMI and ISO/EIC 15504.

2.1 What is the TSP

2.1.1 Context

The success of organizations that produce software-intensive systems depends on well managed software development processes. Implementing disciplined software methods is often a challenge. Organizations seem to know what they want their teams to be doing, but they have difficulties on how to do it. The Team Software Process (TSP), together with the Personal Software Process (PSP), were designed to provide both a strategy and a set of operational procedures for using disciplined software process methods at both the individual and the team levels. Organizations that have implemented the TSP and PSP have experienced significant improvements in the quality of their software systems and have reduced schedule deviation. Countries like Mexico for example are betting to become more efficient in software development by introducing a national TSP initiative [GARCIA2008].

The TSP is designed to guide the development team while they design and develop software-intensive systems. It will work for many other kinds of teams, however its focus is on software development work being that the industry where it has been most widely used. The TSP process guides development teams and their management in planning and developing quality products on predictable schedules. It provides detailed guidance and its process scripts lead the developers through launching and operating their teams. During the initial launch process the development team learns the product requirements and project goals from management. The team produces a plan to meet the objectives. At the end of the generally four-day launch, the team meets with management to discuss and agree on the team's plan and delivery commitments [HUMP2006a].

After the launch, the TSP process guides the team and the team leader in working as a self-directed team. A self-directed team defines its own strategy and processes, produces its own

plan, and negotiates its own commitments with its management. When the team members are properly trained, they typically deliver quality products on schedule. They also find their work much more productive and much more rewarding [HUMP2002].

The individual skills that engineers need to wok in a TSP team are previously developed in one or two weeks of PSP training. The PSP is a self-improvement process designed to help each engineer control, manage and improve his way of work. It is a structured framework of forms, guidelines and procedures for developing software. Properly used, the PSP provides the historical data needed to better make and meet commitments and it makes the routine elements of the job more predictable and more efficient [HUMP1997].

As each person becomes more familiar with PSP, he will learn how to define, measure and analyze the processes. With increased experience each person will be able to enhance his processes to take advantage of new technologies, tools and methods. Above all, it enables each person to understand his own performance and to see where and how to improve it.

The PSP purpose is to help the software engineer to be better. It is a powerful tool that can be used in many ways, for example to manage work, assets and talents and to build skills. It can help to improve the planning, the performance tracking and the measurements of the quality of the products.

2.1.2 Project life cycle

Figure 2.1 details the TSP project life cycle where an iterative approach is followed. Over the next sections a detailed description of each phase is provided in order to give full understanding of the project life cycle TSP implements.

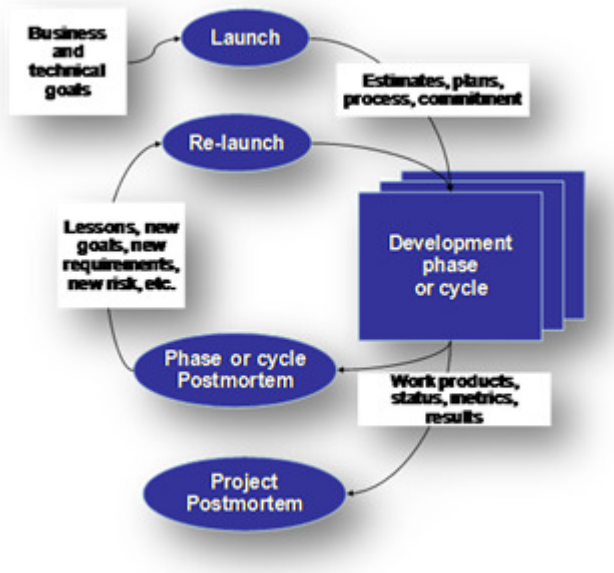

Figure 2.1 - TSP Life Cycle [OVER2008]

2.1.3 The TSP launch and re-launch processes

The TSP cycle starts with a 4-day launch that follows the process shown in figure 2.2:

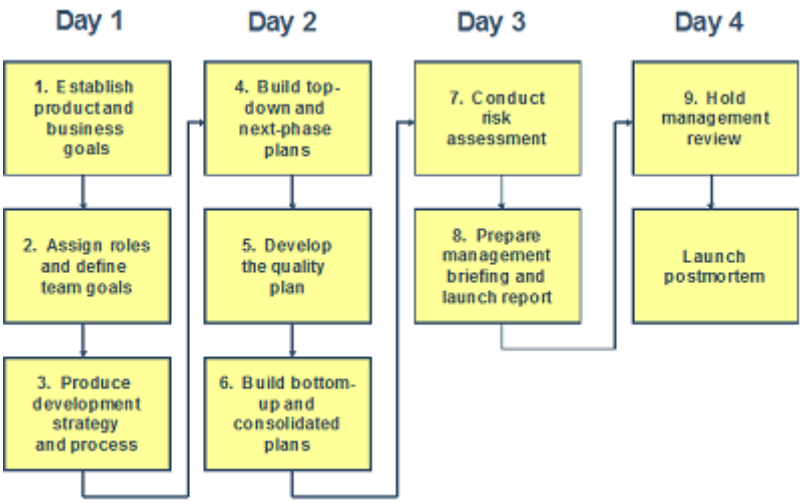

Figure 2.2 - Launch process [HUMP2006b]

There are many reasons to start projects with a launch. The process of forming and building a team does not happen by accident and it takes some time. Teams need to establish their working relationships, determine members' roles and agree on their goals. An hour or so spent on team building issues at the beginning of the project saves time later.

Defining the roles and agreeing on who will handle each role is an essential first step in formation. The team leader together with the team make the initial assignment of the roles based on the preferences of each of the elements. These roles can change as time goes by according to preferences and adaptation of the team to the project and team members.

During the launch period the team will get together and do the planning in detail for the next cycle using all the TSP concepts and techniques. These go from team building to detailed planning always using as much as quantitative data as possible. After the completion of the launch or re-launch there are a set of documents that are produced and that will support the management and execution of the cycle.

Prior to these launch activities there are launch preparation tasks that usually range from 6 to 8 weeks before the start of the project (in case of a re-launch this period is highly shortened and the project does not stop due to this fact). These tasks include all the preparations for the launch including team readiness, team training, scheduling among other relevant tasks.

The concept of self-directed teams is introduced and institutionalized. Besides having a high degree of commitment to the plan these teams sense what is needed to be done without being told and they search for help to do whatever is needed to get the job correctly done. Members have both execution and management roles giving them ownership of the process and the plan. They grow a dedication to excellence.

The launch and execution of a TSP project requires a coach. The coach can help the team to follow the process making sure that any deviations are fixed as soon as possible. The coach also brings an independent view to solve the problems as well is better accepted to the team members once he is not part of the team.

The relaunch is a copy of the launch process with the difference that by the time a relaunch is done one or more launches haver already been made by team, making the process more possibly shorter and more fruitful.

2.1.4 The development phase or cycle

A TSP development iteration can have multiple phases. Either it will contain one only phase of the development life cycle (e.g. high level design) or it will contain several phases that will form a cycle. One TSP cycle can contain a full software development life cycle. The TSP does not enforce any specific technology regarding the way any development should be done once it can be applied to different phases of software engineering (design, development, etc.).

However TSP always enforces discipline, measurements, reviews and inspections. These areas are key to success of a TSP implementation. During the development phase or cycle the documents produced during the launch together with the TSP tools available for managing the process will be the main management tools. They will allow the team to track its progress, assess risks, assess goal completion, follow the earned value plan, among other important tasks. The use of these tools and documents will also produce reports to higher management and customer at specified intervals.

During this phase coaching is crucial to making sure the team follows the TSP process and works towards meeting the planned goals. Improvements in performance depend much on the team coach. Without a good team coach the teams will not reach their best performance.

2.1.5 The phase, cycle or project post mortem

Every TSP development iteration ends with a postmortem, which provides a structured way to learn and improve. In the postmortem what has been done is examined and compared with what was planned to be done. The team looks for improvement opportunities and decides how to change their practices for the next cycle, phase or the next project. Merely trying harder will not produce consistent improvement. The teams must actually change what they do. They can either change the process or discover how to better follow the process that exists.

By using the postmortem process there will be changes from one iteration to the next. The first TSP iteration provides a baseline. The team assess the products produced, the effort spent to produce them and the process steps followed. The accuracy of the plans is determined as well as the suitability of the processes. Problems are identified, their causes determined, and prevention measures devised. The postmortem is the appropriate point to identify specific improvement opportunities and decide where and how to incorporate these changes into the personal and team processes.

2.1.6 Quality management

A quality product satisfies the customer. To satisfy customers, products must provide the functions and performance the users need, be delivered on time and must work reliably enough to do the customer's job.

The principles of software quality management are not new and they come directly from general quality improvement principles (most of them applied in the automotive industry since many years). The basic principles of quality management have been used successfully in many other fields. These principles are as follows:

1. In any quality program the principal objective is to satisfy the customer
2. Management must make quality the top priority, not a top priority
3. The people doing the work must own the quality program
4. To improve quality the process must be changed
5. Quality must be measured
6. Doing the job right always costs less in the long run

People tend to think of quality as the final result or destination. It is not. It is a journey that never ends. As you measure and manage quality you will learn more about it. Then each improvement will provide the knowledge, experience and data needed for the next steps. Organizations should focus on continuous improvement and help the teams to truly believe and follow principles of quality management.

First of all there should be a focus on the personal responsibility to ensure that what is delivered is well done. This responsibility and its results can be boosted by doing personal reviews that after this can be improved by running team inspections. Team inspections together with personal reviews are two of the most effective methodologies to find and correct issues. Quantitative management is the next step in making sure that the delivered artifacts have good quality and that they improve from delivery to delivery.

2.1.7 Measurement

Measurement is fundamental to the TSP process. To measure the work and the work products, the team resources must all use the same defined process, know how to measure that process and work products and consistently measure them. If they do not act this way, their measurements will be inaccurate, inconsistent and largely useless.

It is necessary to measure to get data and this data will help the organization on the following:

- Gain quantitative understanding of their performance
- Evaluate a product, process or organization
- Control a product or a process
- Make an estimate or plan

TSP focuses on four base measures:

- Size
- Time
- Defects
- Schedule

Some of the high level measures are divided in lower level ones in order to clear the scope and need of each of them. As an example the size can be divided in baseline code, deleted code, modified code, new code and reused code.

With these four base measures it is possible to extrapolate a large number of other measures that will be helpful in guiding future projects and evaluating the current one. There are quite a big number of important measures that can be extrapolated from these measures such as

estimation accuracy (size/time), prediction intervals (size/time), time in phase distribution, defect injection phase distribution or defect removal phase distribution.

2.1.8 Benefits and performance results

The success of organizations that produce software intensive systems depends on well-managed software development processes. Software nowadays is critical to most of the businesses and if it is not well managed the organization using the software might incur into issues in other areas related with this specific topic: software. TSP can help organizations in managing and improving the quality of the software needed in the organizations.

TSP was developed to improve the performance of organizations having always in mind the four base measures that any organization should focus when developing software: size, time, effort and defects. The TSP processes will bring knowledge organizations and their resources on how to manage these measures, how to to estimate using them and how to improve their performance. The ultimate goals in mind are to better serve the customer, better treat the employees and increase profitability.

The SEI has done several studies on the advantages of TSP [DAVIS2003]. The numbers show excellent performance results. Looking at the defect density statistics, in a study of 20 projects in 13 organizations, TSP teams had an average defect density in released code of 0,06 defects per added and modified KSLOC. This is more than one order of magnitude better than the average CMMI level 5 organizations. In the same study, cost and schedule have been inquired and measured. Organizations adopting TSP report that project team actual cost and schedule are typically within 5% to 10% variance without sacrificing features. Another study has shown one organization delivering 250KSLOC on time and budget and had only 3 weeks of acceptance testing and 4 post-release defects.

Some known organizations published case studies of using TSP on their software development departments. Microsoft IT invested $3M over six years and accounted to have saved $84M on 200 software projects (Table 2.1). Intuit, a company which flagship products manage small businesses and personal finances routinely achieves 30% more functionality in each release after introducing TSP. ABB, one of the world's leading engineering companies, reduced system test to 4.2% of software development life cycle.

	Non-TSP Projects	TSP Projects
Released On Time	42%	66%
Average Days Late	25	6
Mean Schedule Error	10%	1%
Production Defects/KLOC	1.8	0.5
Sample Size	80	15

Table 2.1 - Microsoft IT project survey [SMITH2008]

All in all TSP can bring enormous results to organizations that have a focus on software development by improving their processes and institutionalize measurements among the people, the teams and the whole organization always improving the quality and performance of the deliverables, the customer satisfaction and saving money.

2.1.9 How to introduce TSP in an organization

In this section it is presented a typical roadmap for implementation of TSP, including the criteria for project choice, strong synergies that result from the application of TSP to achieve high maturity levels, as well as the expected development on tools.

2.1.9.1 Roadmap

The first step for the beginning of the project is the approval by the organization and signing the joint contract between the parties which already will include the dates and selection of the two projects for launch.

After signing day, the setup takes between 1 and 2 months. After that period, the following main activities will take place:

- Meeting for detailed planning [1 day]
- Executive Seminar [1 day]
- PSP Training [5 days]
- TSP training [5 days]
- Projects begin [3-6 months]
    - Launch
    - Coaching
    - Post Morten
- Executive review [1 day]

Figure 2.3 - Roadmap for TSP implementation

This roadmap details the process of launching a pilot with 1 or 2 projects. After the successful pilot projects it is high time for the rollout throughout the organization which involves training internal change agents (instructors and coaches) that lead the spreading of the TSP practices to the rest of the organization on a project by project basis.

2.1.9.2 Criteria for choosing people and projects

The projects and teams to be chosen for TSP should follow the considerations bellow:

- **Projects that involve champions and volunteers** – it is very important to have energetic, competent and motivated organization champions to catalyze the results and amplify and multiply them in the organization, spreading the word and good practices.

Volunteers in these situations are usually seen as the early adopters and being volunteers usually gives the organization a higher level of confidence on their will to adopt the new processes.

- **Projects of 3 to 6 months duration** – The 3 to 6 months duration is ideal to see the results in short term in a consistent way and to guarantee that there is time enough to see the improvements.
- **Projects that do not have very unknown technologies** – It is recommended to have projects where the technologies are known so that the advantages of TSP can be shown faster as they will not be dependent on the technology.
- **Projects with stable scope** – Chaotic projects could increase the risk of the loss of focus on the learning aspects by organization teams and could lead to not applying some of the needed practices for TSP being successful.
- **Projects that are starting** – The projects starting bring a cleaner environment for project and new best practices instantiation. Maintenance projects are not recommended for pilot projects (although TSP can also be applied to this kind of projects).

2.1.9.3 Synergies with other process improvement initiatives

The six sigma initiative will provide a framework of tools and training in the organization in order to quantitatively analyze the information. This project will be an excellent complement to the TSP project that will have a more bottom-up approach.

It was verified that the base measures utilized by the six sigma project are the same as the TSP project. TSP offers a more complete set of derived measures and tools for the quality and engineering measures.

Several organizations have included in the context of a CMMI implementation with the objective of accelerating the achievement of a maturity level. NAVAIR, with the combined implementation of CMMI and TSP managed to reach the level 4 of CMMI in a record time with a return on investment of several million US dollars [ARMAND2006].

2.1.9.4 Tools

TSP comes with a set of tools which are recommended for the pilot projects.

The analysis of the TSP tools and the organization tools will allow the identification of functionalities needed to be incorporated in the organization own legacy tools making sure tools will not be problems but always solutions for further TSP growth.

2.2 TSP based certification

The SEI started to develop a method for certifying software organizations according to the TSP methodology. This methodology has been developed together with the Mexican government initiative based on the national objective of being the world's highest quality software supplier by 2013 [HUMP2008].

This method can be performed in 3 ways having each of them its strengths and weaknesses:

- Certification of individuals
- Certification of processes

- Certification of products

When one is certifying an individual the main criteria are the skills, the experience and the knowledge. This can be done through verifying references and experience and has the advantages of having a clearly defined scope and criteria and being easy and cost-effective to administer.

Moving to the process certification, the way of doing this would be to certify according to a defined capability criteria likewise CMMI. This can be done through an organization assessment (as the method described in the next chapter). It also has the advantage of having a clearly defined scope and criteria as well as enabling flexibility on implementation and on the technologies used. The main disadvantages are the difficulties in assuring that the process will be used properly, and that the process will produce the desired results. It also makes the process of assessment more expensive as it has to be quite comprehensive.

When looking at certifying products some criteria has to be met in order to proceed with the method: function, performance and quality. The great advantage of this method is that it addresses specific needs, however it is only available after the product is ready or almost ready and has to rely on testing making it most of the times expensive and time consuming.

This method will surely be very accurate, objective and repeatable in organizations using TSP, as TSP data is very good and broad for this purpose. With the data different certification profiles can be produced.

The TSP organization certification strategy uses a combination of the 3 different strategies, making it very complete, but also expensive. The certification of team members ensures that they know the proper methods and how to gather and use data. The process validation ensures that an effective and defined process is faithfully followed. The product quality verification checks the quality of the products as well as the customer satisfaction.

The certification profile is a visual display of numeric scores. It provides a comprehensive picture of organizational performance being a product of the certification process and a scorecard that characterizes the organization capability. It has 3 components:

- Performance
- Fidelity
- Coverage

The performance profile shows the actual product and project performance. The Figure 2.4 shows this profile.

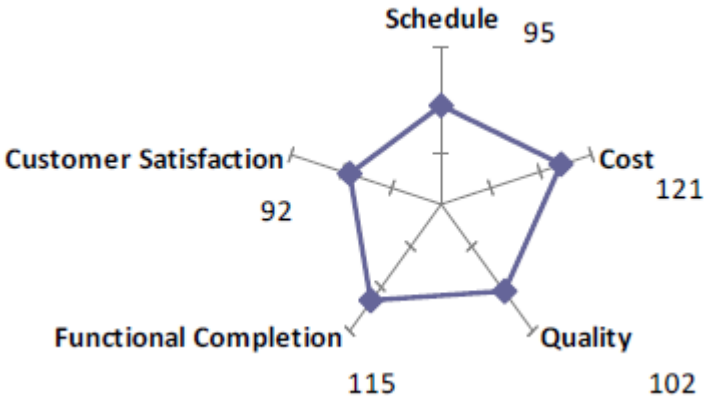

Figure 2.4 - Performance profile [HUMP2008]

The fidelity profile shows the project's ability to obtain consistent results. An example of this profile can be seen in figure 2.5.

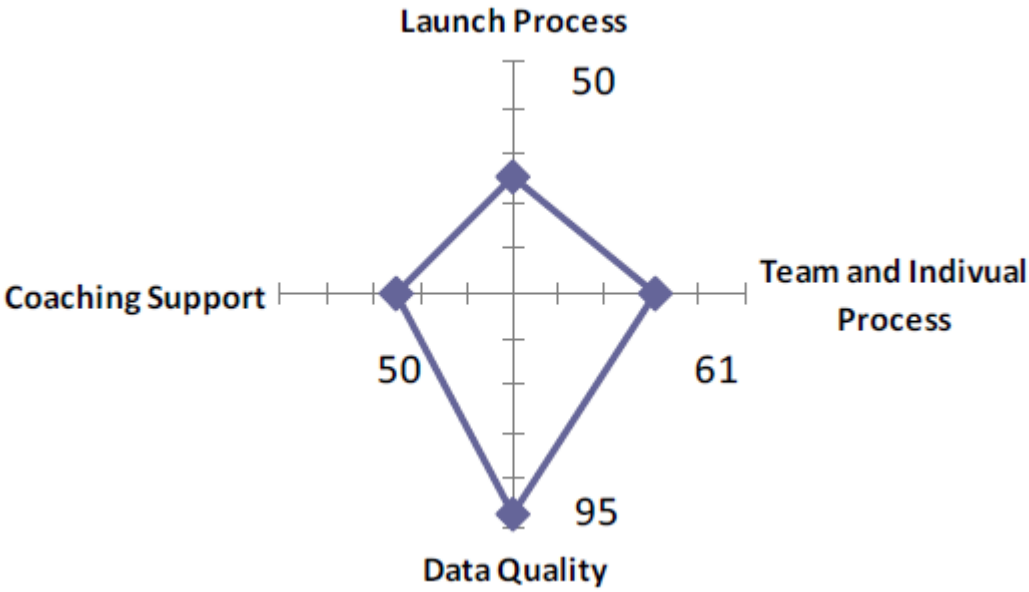

Figure 2.5 - Fidelity Profile [HUMP2008]

The coverage profile objective is to measure the institutionalization of TSP. Figure 2.6 shows an example of this profile.

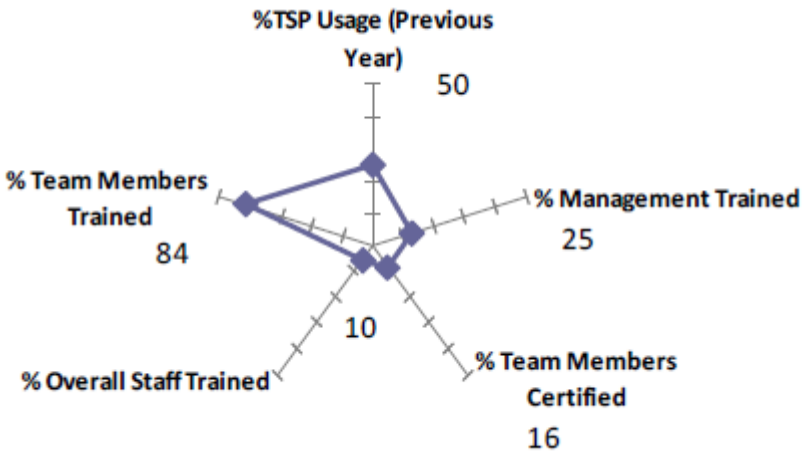

Figure 2.6- Coverage profile [HUMP2008]

All these profiles can now be combined in a graph that gives a 3D perspective of the company. Figure 2.7 shows this view.