AALTO UNIVERSITY
School of Science and Technology
Faculty of Electronics, Communications and Automation
Department of Communications and Networking

Antti J. Hätinen

A Method for Evidence Based Quality Practice Engineering

Master's Thesis

Espoo, March 11th, 2010
Version 1.0-rc5

Supervisor: Prof. Jukka Manner

Instructor: Jari Vanhanen, Lic.Sc.

Aalto University School of Science and Technology Faculty of Electronics, Communications and Automation Degree Programme of Communication Engineering	ABSTRACT OF MASTER’S THESIS
Author Antti J. Hätinen	Date 11.3.2010 Pages 90+10
Title of thesis A Method for Evidence Based Quality Practice Engineering	
Professorship Network Engineering	Professorship Code S-38
Supervisor Prof. Jukka Manner	
Instructor Lic.Sc. Jari Vanhanen	

The quality of the software has been and remains as a key problem of the software industry. Especially interesting questions is, how a certain level of quality can be systematically reached. Evidence-based software engineering (EBSE) tries to provide answer to this question by collecting empirical evidence on different aspects of the software engineering process and deliverables. In this work the perspective of quality practices and goals has been selected for constructing and evaluating a method that could be used for industrial software process improvement (SPI).

Four subject Finnish middle-sized software product companies were studied by performing in total five action research and constructive interventions. First the novel Quality Palette Analysis –method was applied for three subject companies in different variations. Next the Indicator Analysis and the New Method A (NMA) –brainstorming method were constructed and applied by the author. As a final constructive step, the author designed a novel Semantic Web –based EBSE experience factory for mapping the empirical evidence on the relationship of the quality goals and practices.

The results of the study are two-fold. While the collected data on the relationship between the quality goals and practices remains insufficient to draw definitive claims, it seems that the EBSE DB provides theoretically a very high utility model for software process improvement (SPI) initiative evaluation, training & education, and for the scientific research. The database is able to answer questions such as “*which practices should be used to ensure reaching of effort of 8h per update*” with an answer vector of practices “smokeTesting” and “alphaBetaTesting”. Due to small amount of samples the database is currently unable to answer for example, how an update effort of 1 hour less could be reached and the results can be considered unreliable. However, the reliability and range of answerable questions could be easily improvement by performing systematic literature review on all available scientific evidence on the software engineering practices.

While such a system remains as a prototype, the NMA -brainstorming method provided clearly the best yield of SPI initiatives compared to the time invested in the data collection. The other methods were by the best cumbersome and can’t be recommended for industrial application in their current forms. However, the author provides suggestions how the QPA-method could be altered to function in the Future as a primary data collection method for the EBSE DB in context of individual companies by omitting the workshop –phase and developing an automatic data collection tool similar to the current QPA pre-assignment.

Keywords
Software engineering, quality, practices, semantic web

Aalto-yliopiston teknillinen korkeakoulu Elektroniikan, tietoliikenteen ja automaation tiedekunta Tietoliikenteen tutkinto-ohjelma	DIPLOMITYÖN TIIVISTELMÄ
Tekijä Antti J. Hätinen	Päivämäärä 11.3.2010 Sivumäärä 90+10
Aihe Menetelmä näyttöperustaiseen laatukäytäntöjen kehittämiseen	
Professuuri Tietoverkkotekniikka	Professuurin koodi S-38
Valvoja Prof. Jukka Manner	
Ohjaaja Tekn.Lis. Jari Vanhanen	

Ohjelmistotuotannon laatu on yhä yksi IT-teollisuuden suurimpia ongelmia. Erityisen kiinnostava kysymys on, kuinka tietty laatutaso voitaisiin saavuttaa systemaattisesti. Empiirinen ohjelmistotuotanto (EBSE) pyrkii vastaamaan tähän kysymykseen keräämällä todistusaineistoa aitojen ohjelmistotuotantoprosessien ja -tuotoksien toimivuudesta. Tässä työssä tutkimusnäkökulmaksi on valittu laatukäytäntöjen ja –tavoitteiden valinen suhde konstruoimalla ja vertailemalla uusia menetelmä ohjelmistotuotantoprosessien kehittämiseksi. Tutkimusta varten suoritettiin yhteensä viisi toimintotutkimuksellista ja konstruktiivista interventiota neljässä suomalaisessa keskikokoisessa ohjelmistotuote-yrityksessä. Ensiksi sovellettiin laatupalettianalyysi –menetelmää kolmessa yrityksessä. Tämän jälkeen konsturoitiin uudet indikaattorianalyysi ja "NMA"-aivomyrsky menetelmät. Viimeisenä konstruktiona rakennettiin semanttisen verkon -teknologiaan pohjautuva tietämyskanta laatutavoitteiden ja –käytäntöjen valisten vaikutusten tutkimiseen.
Tutkimuksen tulokset ovat kaksiosaiset. Huolimatta siitä, että todistusaineiston määrällisen puutteen takia tilastollisesti merkittäviä tuloksia ei voida tässä työssä esittää, uusi tietämyskanta vaikuttaisi teoreettisesti pystyvän tutkituista menetelmistä ainoana vastaamaan tutkimuskysymyksessä esitettyyn kysymykseen. Teollisuuden lisäksi tietämyskantaa on mahdollista hyödyntää akateemisessa tutkimuksessa, opetuksessa, koulutuksessa ja ohjelmistotuotantoprosessin kehitysideoiden arvioinnissa. Tietämyskanta pystyi vastaamaan tiettyihin kysymyksiin, kuten "*mitä käytäntöjä tulisi soveltaa varmistuakseen, ettei ohjelmistopäivitykseen kuluva työmäärä ylitä 8h:a?*". Vastauksena saatiin käytäntövektori "savutestiasennus" ja "alfa/beta –testaus". Vähäisen tutkimustiedon takia tietämyskanta ei kuitenkaan pysty toistaiseksi vastaamaan esimerkiksi miten päivitykseen kuluvaa työmäärää voitaisiin lyhentää. Lisäksi tuloksen luotettavuutta voidaan pitää heikkona. Tietokannan antamien vastauksien laajuutta ja luotettavuutta voitaisiin kuitenkin helposti parantaa laatimalla systemaattisia kirjallisuuskatsauksia kaikesta saatavilla olevasta ohjelmistotuotantokäytäntöihin liittyvästä tieteellisestä kirjallisuudesta ja syöttämällä tulokset tietämyskantaan.
Kunnes tietämyskannasta pystytään kehittämään teolliseen käyttöön soveltuva versio, NMA –aivomyrsky on selkeästi tehokkain menetelmä kehitysideoiden tuottamiseen. Muut tutkitut työpajamenetelmät eivät aina tuottaneet lainkaan kehitysideoita eivätkä siten sovellu teolliseen käyttöön. Kuitenkin tämän työn lopussa esitetään ehdotuksia siitä, kuinka laatupalettianalyysiä voitaisiin hyödyntää tietämyskannan tiedonkeruumenetelmänä mm. poistamalla työpaja-vaihe ja kehittämällä uusi esitehtävätyökalu.

Avainsanat
Ohjelmistotuotanto, laatu, käytännöt, semanttinen verkko

Acknowledgements

I would like to thank SoberIT / SPRG research group for providing the data and financial support to perform this thesis. Finally I would like to thank my wife, who is working as a quality manager in the construction industry, for her reflections on the key quality concepts such as Target Costing, Six Sigma and TQM.

The work of writing this thesis has been also a great personal journey challenging my remaining disillusions and enabling to see the reality clearly especially from the TQM and queuing theory point of view. At last after my second master's degree I feel to have learned enough to engage the world in my full capacity in the craft of creation.

The researchers have a large ethical responsibility, since the recommendations they give to the society and the companies is perceived to have higher validity than from other sources. Thus, an emphasis in this thesis and in general research should be put on the quality of the results presented, the interpretation of the meaning so that they truly present the truth, the current best scientific evidence available. A sufficient rigor should be placed on the literature review and the evaluation of the research methods. This is also the ethical objective of the thesis that has bended the original topic into somewhat new directions to bring forth the truth.

Contents

Glossary

AHP	Analytical Hierarchy Process
BPCH	Best Practices Clearinghouse https://bpch.dau.mil
CSF	Critical Success Factor
CMM	A software maturity model by SEI (Software Engineering Institute, USA)
COPQ	Cost of Poor Quality – a Six Sigma -concept
CRC	Class-Resposibility-Collaboration –index card for technical design in XP
CTQ	Critical to Quality – a Six Sigma -concept
DB	Database
DBR	Drum-Buffer-Rope –production system
DPMO	Defects per Million Opportunities – a Six Sigma -concept
EESWS	ESPA Experience Exchange Workshop
EBSE	Evidence-Based Software Engineering
EF	Experience Factory
ESPA	The name of the research project
GEQ	Good Enough Quality
GQM	Goal-Question-Metric
IA	Indicator Analysis – a SPI method by the author
JIT	Just-in-Time –production system
N/A	Not Available
NMA	New Method A – a brainstorming method by the author
MASTO	ESPA sub-research project at Lappeenranta University of Technology
MBA	Master of Business Administration
MbO	Management by Objectives
OPEX	Operational Expense
OWL	Web Ontology Language
PDAC	Plan-Do-Act-Check – TQM cycle
QFD	Quality Function Deployment – a Japanese product design method by [Akao90]
QFF	QPA for Features – an extension of QPA for feature level SPI
QGP	Quality Goals and Practices -method
QGWS	Quality Goal Workshop
QPA	Quality Palette Analysis – a SPI method by [Itkonen07]
QPWS	Quality Practice Workshop
QUPER	Quality Performance product design method by [Regnell08]

R&D	Research and Development
RDF	Resource Description Framework http://www.w3.org/RDF/
RQ	Research Question
SEEDS	Software Engineering Evidence-based Database http://www.evidencebasedse.com
SLR	Systematic Literature Review
SME	Small to Medium-sized Enterprises
SMED	Single Minute Exchange of Die –production system
SPC	Statistical Process Control
SPI	Software Process Improvement
SoberIT	Software Business and Engineering Institute at the Aalto University School of Science and Technology
SQUID	The name of the sub-research project at the SoberIT
SW	Software
TQM	Total Quality Management
TPS	Toyota Production System
WP	Work Package
WS	Workshop
XP	Extreme Programming by [Beck00]
XSLT	Extensible Style-sheet Language Transformations

1. Introduction

"Give a man a hammer, and he will begin to see the world as a collection of nails"

- Barry Boehm [Boehm81, p8]

The topic of this thesis is to develop a method to improve the match between the quality practices and the quality goals for small to medium sized (SME) software companies. In the other words, the main problem at hand is how to improve the fitness of the organizations social structure to its competitive environment's quality requirements. The process of improving the *structural fitness* of an organization can be described also as a search process of the *possibility space* [Battram99, p.108]. The term possibility space derives from the mathematical concept of a "*search space*", the set of all possible solutions or the domain of the function to be optimized. The *fitness landscape* is a related metaphor in complexity theory, showing the locale of the fitness minima and maxima. Thus, the process to search the possibility space is also a process to map the fitness landscape, or the competitive environment of a company. Normally, when people search the possibility space, they explore only a tiny, familiar part of their surroundings. The dominant *organizational culture*, including predominantly the language, sets the constraints for the allowed search space. The self-preservation process of the culture self-censors the thinking of the individuals and forbids the thinking outside the *predictable zone of order*. However, while the individuals of the organization have different *perspectives,* the collective union of the perceived individual possibility spaces is larger than of any single persons search space. Unfortunately, for example due to ineffectiveness of communication, the groups also experience process loss that reduces their actual performance compared to the potential [Sauer00].

In this work the process of mapping the possibility space constrained by the organizational culture is called as *social search.* In contrast according to Senge a *learning organization* is able to question the unspoken cultural taboos to enable collective mutual learning and an improved adaption to the contemporary business environment [Senge90]. A possibility to quantify the intellectual capital can be the *systems thinking* and modeling of the organization in terms of components (queues) and links (information flows).Thus the second approach in this work is to construct an *Experience Factory* [Basili92] combining an union of diversified perspectives to overcome the potential dysfunctions the social search process and to enable a learning organization by application of holistic systems thinking.

Another perspective for justifying the topic is the nurturing and acquiring of *intellectual capital* [Stewart98]. Stewart suggests the shift of the primary factor of production in the knowledge age from the financial capital to the intellectual capital. Further, he recognizes the structural capital more important for the companies than the human capital, since it's owned by and can be built as the company's asset, while the human capital moves easily when the employees change their jobs [Stewart98, p.107]. According to Stewart, the best structural capital is one that obstructs the least the employees from working with the customers. The two purposes of the structural capital are to first amass stockpiles of knowledge that is valuable to customers, and second to speed up the information flow inside the company. Third, he defines the customer capital as "*the likelihood that the customers will continue to do business with us*". As an example, he mentions a university to usually have a high condensation of human capital as a collection of brilliant researchers, but poor in structural capital. In comparison McDonalds has a very high staff turnover rate (low human capital), but has invested heavily to build the franchise structure to facilitate its world-wide operations (high structural capital). The process of improving the social structure of an organization can be viewed as build-up of the intellectual capital, which can therefore be regarded as the primary resource of software production.

Bontis found significant and substantive impact ($R^2 = 56\%$) by the structural capital on the financial performance by performing a survey on 64 MBA candidates with previous work experience in executive positions [Bontis98]. However, he found no correlation between the human capital and financial performance alone, meaning that hiring the brightest people available will not result in financial performance, if there are not sufficient structural and customer capital present to utilize and support the human capital. Bontis describes structural capital as the "organizational routines", "internal organizational links" and suggests that it can be measured by evaluating "efficiency" and "accessibility" of knowledge. Thus the structural capital can be also interpreted as the collection of software engineering practices employed in the organization. In particular the structural capital can be understood as the capabilities of the production system of a company. In the Total Quality Management (TQM)-model [Deming86], which is close to the ideology of both the Statistical Process Control and the Agile methodologies, the structural capital is in the essence the capabilities of the pull queue network, its throughput and yield (i.e. the amount of errors and waste produced).

Next the basic concepts of this work are introduced by describing the meaning of quality, experience factory, short history of the context of application, quality goals, and the change management. After this the research methodology is described, followed by the results, analysis and the conclusions.

2. Software Process Improvement

"There is no substitute for knowledge" - *W.E. Deming*[1]

Studies have identified the continued management commitment, involvement of respected technical staff, allocation of sufficient resources and a clear statement of improvement goals as the critical success factors for software process improvement (SPI) and software process re-engineering (SPR) [Napier08]. The SPI is the process of developing the structural capital of a software engineering organization.

> Principle 1: SPR should consider the organizational context by identifying goals and policies for SPI and incorporating viewpoints of internal and external stakeholders [Napier08].

Kitchenham rallied software engineering researchers to adapt the evidence-based paradigm to study, if similar advances than in medicine, psychiatry, nursing, social policy and education could be reached also for the software improvement [Kitchenham04]. The goal of the Evidence-based Software Engineering (EBSE) is *"to provide the means by which current best evidence from research can be integrated with practical experience and human values in the decision making process regarding the development and maintenance of software"*. However, while the evidence-based medicine has resulted in tens of thousands of articles and substantial advances in science, Kitchenham agrees that the success of the paradigm in medicine could be caused by the lower practioner skill requirements, which would lead into subject and experimenter biases when applied to the more demanding software engineering. A doctor doesn't require specific skill to prescribe and administer a treatment, while software engineering cannot be practiced without intense experience and skill. Thus, software engineering is more similar to evidence-based surgery than -medicine.

The ESPA – Towards Evidence-Based Software Quality -research project is funded by Tekes and divided into two subprojects SQUID by SoberIT at the Aalto University School of Science and Technology and MASTO at the Lappeenranta University of Technology. The objective of the 3 year research project is to find empirical evidence towards building new software assurance services and products. The main goal of the SQUID subproject is "*to develop a practical method for goal and evidence-based quality practice selection and improvement*" [SQUID08]. The project was granted funding in August 2008 and the research plan was accepted to start the implementation from the work package (WP) 1.1 "Quality Goal Setting and Quality Practice Selection" with two main research areas:

[1] http://statistical-process-control.org/dr-demings-revolution/

How the quality goals can be identified at different levels? Is it possible to create lists of quality goals that can support identifying relevant quality goals at different levels?

How the quality practices can be selected to ensure reaching of the quality goals? What kind of prioritization of quality goals supports the selection of quality practices? What kind of documentation of quality goals supports the selection of quality practices?

The main deliverables of the SQUID WP1.1 are guidelines for identifying, prioritizing and documenting quality goals, and a method for selecting quality practices based on the identified quality goals. This thesis is a subproject of the WP1.1 aiming to answer to the second research question and to deliver the method for quality practice selection.

From the companies' point of view yet another concern of interest is about whether the ESPA program could provide insight to how the software companies should be organized. According to the Company B, the organization of the software engineering unit is a matter of significant importance, since it affects greatly the efficiency of the production, for example by affecting how the people communicate with each other[2]. Thus the industry faces an additional optimization problem trying to organize the operations optimally by several contradicting goals.

2.1 Total Quality Management

While it is generally believed that improving the quality is not the sole objective of the companies, the TQM-school by Deming believes, however, that that improving the quality reduces the waste of the rework and scrap and thus improves the profitability. Other authors have given divergent definitions for quality. The differences in definitions seem to reflect the differences in the mental models and can be divided into at least four major schools of quality. The two most fundamental differences can be found between the "traditional" and Total Quality Management (TQM) schools.

The traditional belief of the Good Enough Quality (GEQ) is the opposite to the TQM arguing that the over-quality is expensive and only the minimum required quality level should be produced [Bach97]. Yet another traditional view includes so called maturity models such as CMM and SPICE that focus in the standardization. However, reaching a certain process maturity level does not guarantee high quality or low

[2] QPWS 12.11.2008 informant B2

cost. Also the maturity models don't provide means how to improve the maturity beyond for example the CMM level 5, while the TQM -school believes in continuous improvement ad infinitum.

Crosby, the author of the "zero defects" -concept, defines quality as "conformance to requirements" [Crosby79]. The TQM school's renowned author Juran disagrees with Crosby and states the quality is "fitness for use" and not conformance to specifications [Juran74]. Juran argues that companies are often unable to define the requirements to match the actual usage situation. The ISO9000 standard states the quality is "the degree to which a set of inherent characteristic fulfills requirements" [ISO9000]. To discuss the mentioned definitions for quality in detail, one could interpret the Crosby's and ISO definition actually to be equal to the Juran's by allowing the requirements to be implicitly defined by the customer. The explicit instantiation of the requirements may or may not match true requirements, but this can be regarded as a mere source for error. Inside the organizations, the customer of a certain process step is usually another internal customer, or externally some other company in the value chain to produce the end product to the customer. However, by definition, the customer always defines the expected level for the quality, not the producer [Drucker85], and thus all three definitions are in the essence equal.

Since Deming introduced the TQM in Japan after the WWII, two major sub-schools of TQM emerged. While being a statistician, Deming promoted the idea of Statistical Process Control (SPC) that later developed into concepts such as Six Sigma and "Poka-Yoke" (a Japanese concept for error-proofing). Following Deming's ideas Taguchi defines quality as "Uniformity around a target value", referring to the idea of reducing the standard deviation of outcomes [Taguchi92].

The idea of the Poka-Yoke and methods such as "Designed for Six Sigma" (DFSS) [Harry00] are based on the idea of designing the products to minimize the possibility that a defect or error could be experienced in production or in use. In the DFSS –method the product is designed by compensating the quality problems by introducing for example 1.5σ or 3σ tolerances at the design phase, and by favoring the usage of known low-defect components. In the Six Sigma and Statistical Process Control -methodologies the *capability* of the process is evaluated to check whether the process has the possibility to meet the requirements [Vonderembse88, p.721]. Thus, the larger the deviation, the larger tolerances must be utilized to produce the desired quantity of quality conforming end products. In software engineering, the refactoring, complete redesigns and the iterations can be viewed as tolerances to produce a quality conforming product. However, large tolerances also imply higher cost, and thus the objective of these methods is to minimize the variation.

The Six Sigma is an extension of the SPC methodology. The more widely practiced Six Sigma production version defines quality as defects per million opportunities DPMO [Harry00], focusing both on Crosby's zero defects and Deming's SPC ideas. The assumption of the approach is that each critical-to-quality (CTQ) defect causes cost to deliver the product, and thus should be eliminated. The term sigma (the standard deviation) refers to the claim that one sigma improvement in defect mitigation improves the quality by roughly 10 -fold. If the standard deviation is larger than 6 standard deviations, no output conforms anymore to the quality standard. However, conceptually the notation is inverted and the 6σ quality level is regarded as the highest possible process capability. The Six Sigma method attacks the traditional assumption of quality per cost –ratio of Good Enough Quality (GEQ) [Bach97], that assumes that after a certain level of quality it is not anymore economically feasible to improve the quality level, by claiming that even larger cost reductions can be achieved when for example quality control practices can be removed due to close to perfect quality (less than 3 DPMO). The Six Sigma states that the Cost of Poor Quality (COPQ) optimum is below the traditional 3-4 sigma (7%-0.5% DPMO), because the *hidden factories* (such as non-documented bug fixes) makes the true cost of poor quality invisible for the cost accounting systems and the management. The Six Sigma approach states that the traditional optimum of 3 sigma quality includes 25% of hidden COPQ per revenue unit earned. In software the COPQ is even higher, typically between 30-90% of the total budget [NIST02], or on certain methods such as the waterfall, higher than 100% [Royce70]. A six sigma level process should have COPQ on the level of 1-2%, which means huge difference implications on profitability. Thus it seems that software engineer would provide a fertile ground for the Six Sigma –approach for software process improvement.

The two different definitions for quality, the average and the variance, can be also viewed from the Total Quality Management point of view that describes the whole picture in a concise manner [Feigenbaum51]. The second TQM -school of Lean Management focuses more on the overall production system throughput than the quality alone, but introduces a key concept of the Kaizen cycle of continuous improvement including three main phases: Standardization, Improvement and Innovation (Figure 1). The standardize phase is often less explicitly emphasized, but many especially software quality models such as CMM focus almost solely on documentation and instructions (i.e. the Standardization cycle). The idea of the TQM is to lock-in the improvements and innovations, or otherwise they are lost within a few years. The quality on the standardize phase can be viewed to focus on reduction of the variance. However, Six Sigma assumes also that the reduction of the variance is strongly linked also to the improvement phase, where the variance reduction is directly linked to reducing the Cost of Poor Quality, and therefore also improving profitability.

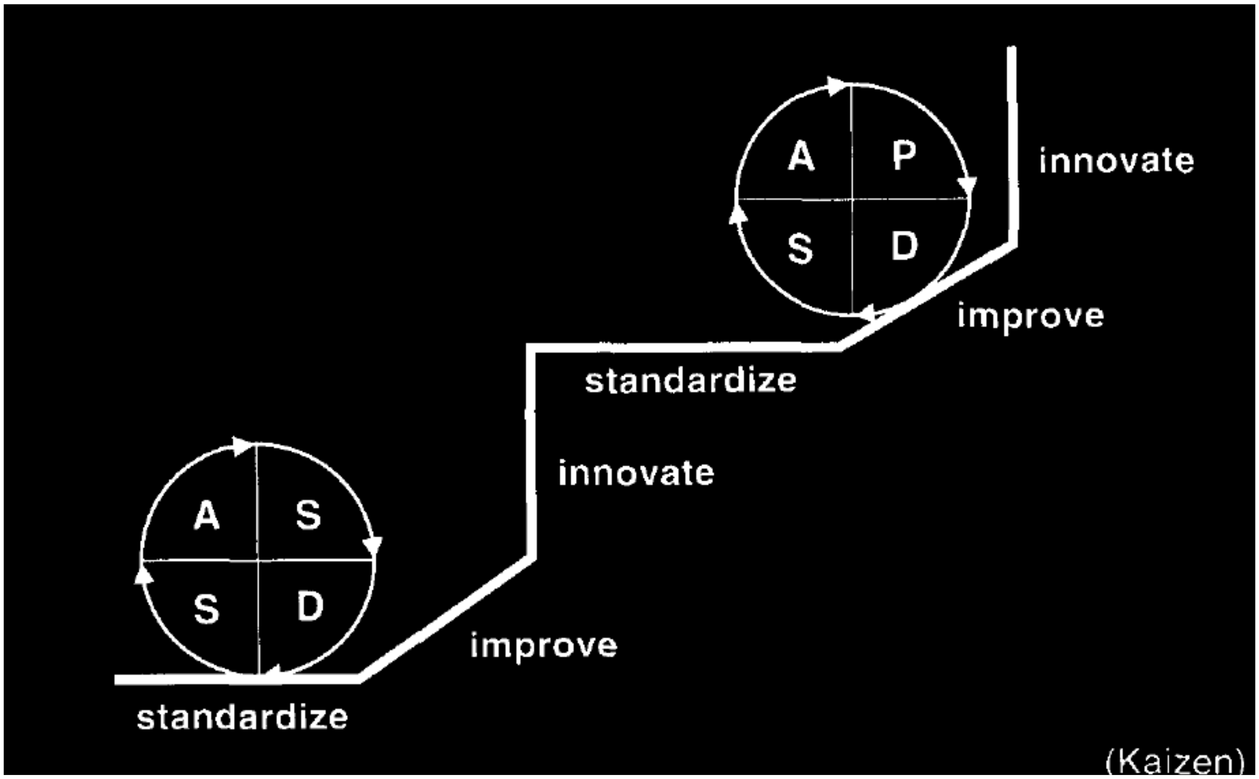


Figure 1- The TQM Cycle [Zultner93]

The improvement is the phase often identified by the Plan-Do-Act-Change (PDCA) –cycle (originally by W.A. Shewart, but often contributed to Deming). Once the variance has been reduced, it is possible to also raise the average quality. In archery it is much more difficult to make the arrows hit near each other (reduce the variance), than adjusting the sight to hit the bull's-eye (improve the average). In the quality literature and often also in practice, the easier path of defining the quality as an average is most often chosen. However, as described above, without taking the variance in to the account as tolerances, it is not possible to reach the highest levels of quality. If the quality is not built-in, the company risks costly rework or scrap [Vonderembse88, p.714], or even worse, product returns by the customer. The question thus is, what is the survival triplet of the company, and if the degree of CoPQ of 25-30% (3-4 σ) is competitively feasible or not.

Finally the innovation phase creates a discontinuity to the performance by radical improvement. One suggested method is to identify the bottlenecks and constraints of the total output, and to find solutions how they can be exploited. Zultner suggests that often these constraints are self-inflicted and can be broken by changing policies [Zultner93].

To discuss the radical improvement in detail, Goldratt interprets the purpose of an organization untraditionally, but according to the Lean principles: "*The goal is to reduce operational expense and reduce inventory while simultaneously increasing throughput*" [Goldratt84]. The throughput is the rate at which the system generates money through sales. The inventory is all the money that the system has invested in purchasing things which it intends to sell. Operational expense (OPEX) is all the money the system spends in order to turn inventory into throughput. In all systems, there are phases that constraint the total throughput of the whole system called bottlenecks. Goldratt claims, by optimizing the performance of any other part of the system than the bottleneck, no throughput improvement can be achieved. This is similar to the good software engineering practice familiar from the Extreme Programming [Beck00] "Leave the optimization till last", which concurs by accepting the fact that there is always a part of the code that constraints the total performance. A typical bottleneck is for example a slow or uncached SQL query that slows the execution by an order of several decades compared to the non-bottleneck code. The bottleneck can't be found by reviewing the code, but it has to be located by using profiler tools and measurement.

Goldratt suggest a Drum-Buffer-Rope (DBR)–production system to optimize the organization's performance. A drum sets the pace, cycle time, the Taktzeit (i.e. iteration length) for the whole organization. If any other activity produces faster than the drum, it generates excess inventory, which by definition is additional operational expense (OPEX). However, having a long cycle time implies also large inventory in work-in-progress (WIP) introducing also OPEX inventory waste. In software engineering this means, for example if the coders are the slowest part, the designers should not produce designs any faster than what is needed next by the coders, suggesting a small batch size or a short iteration should be used. The buffer is the backlog queue of tasks that should be done. This should be theoretically in length as close to zero as possible, but in practice it has to have some items, so that there is no risk that the total throughput would be affected by having the coders nothing to do. When the buffer reaches the predetermined alert level, the rope is pulled to signal the preceding phase that new work should be done (a demand managed pull system, where nothing is produced in advance into the inventory). In essence, the rope is equal to the Kanban –card signals, that notify the preceding phase either a permission to move material (C-Kanban) or to produce new material (P-Kanban) [Vonderembse88, p.441]. The DBR –model is useful in understanding the new lean and agile software engineering methodologies. [Poppendieck03] describes how to use the Extreme Programming [Beck00] Class-Responsobility-Collaboration (CRC) and the Scrum index cards as Kanban signals to implement a lean production system with theoretically maximum efficiency. The Kanban signals provide also a facility for making the work flow visible for the employees and the management enabling the identification, which practices or other things are acting as the bottleneck for further quality and performance improvement. From the QUPER point of view this bottleneck can be understood as a cost barrier.

The DBR -model is closely equal to the Just-in-Time (JIT) [Ford22] and Toyota Production System (TPS), with additional view of the constraints. A typical challenge in JIT is the setup time to readjust the production to produce typically a small batch of new material (or code). Traditionally the industry has used to produce large batches of similar items, but later on JIT and TPS have been able to reduce the setup times from orders of 3-4 hours to systems like SMED (Single Minute Exchange of Die) where setup time is in order of 1-3 minutes, for example by configuring the next setup while the machine is working on previous batch. In software engineering this could for example mean that the bottleneck resource (coders) should not be used in activities that are not contributing in generating throughput, such as design or planning sessions, but which can be provided Just-in-Time for the coders when they complete the previous job. The second problem in JIT is caused by the reduced inventory, which causes the defects (bugs) to cause serious hiccups in all other operations. Thus approaches such as SPC, Six Sigma and Kaizen (continuous improvement to eliminate waste) have been introduces to reach the rigorously the quality goal of zero defects. Thus the emphasis is placed on minimizing the process variance. This is very different to the traditional view of Good Enough Quality (GEQ), where the economical reasoning is based on avoiding the production of expensive over-quality not needed by the customers and knowing which bugs can be shipped, while they are not critical to quality. The Six Sigma also agrees with the idea by separating the non-critical-to-quality factors from the measurement used to define the quality level. However, by assuming Drum-Buffer-Rope –style system that optimizes the profit by minimizing the inventory where possible, the quality issues are much more severe than on traditional, lower throughput production systems. Due to the inherit total efficiency caused by the higher throughput, the JIT/TPS -like companies are more likely to out-compete their less-efficient craft and mass-production rivals especially in the long run [Cooper95].

Vonderembse divides the cost of quality (waste) into three general categories: costs of preventing defects, costs of appraising quality, and costs of the production defects [Vonderembse88, p. 717]. Each of the categories includes the specific costs described in Table 1

Table 1 - Costs of Quality [Vonderembse88, p.717-718]

Type	Activity	Cost type
Preventing defects	Quality planning	Time that is spent planning
Preventing defects	Quality training	Developing and operations programs to train employees in quality control and test procedures
Preventing defects	Design of quality systems	Sstudying and analyzing production systems, designing a means of control, or suggesting ways to improve existing processes
Preventing defects	Quality reporting	Preparing and distributing reports about quality to middle and upper management
Appraising quality	Testing and inspection	Actually measuring and testing parts and materials
Appraising quality	Quality audits	Measuring the level of quality and evaluating the systems and procedures used to monitor and maintain the quality
Production defects	Internal costs	Defects found before being shipped to the customer
Production defects	Scrap	Material and labor that were put into an item that now must be discarded or sold for scrap
Production defects	Retest	Reinspecting parts or items that have been reworked
Production defects	Scrap / Rework	Correcting defects in an item that can be salvaged through reworking
Production defects	Retest / Downtime	(Equipment) that must sit idle due to defects (Author's note! Only bottleneck downtime matters)
Production defects	Yield losses	Material caused by faulty processes
Production defects	Disposition	Determining whether defects can be corrected or must be scrapped
Production defects	External costs	Defects found after delivery to the customer
Production defects	Complaint adjustment	Monitoring and responding to complaints
Production defects	Returned material	Correcting or replacing and returning defective products
Production defects	Warranty charges	Correcting defects that have occurred while the product was under warranty
Production defects	Allowances	Providing an allowance to the customer if that product fails within a stated period
Production defects	Loss of goodwill	Associated with customers who reduce their purchases or take their business to a competitor because of dissatisfaction. This may be one of the most difficult costs to measure, but also one of the greatest.

Traditionally the optimal cost of quality level has been viewed to be related to the linearly increasing costs of defects and exponentially increasing costs to prevent defects, placing thus the optimum to be nonzero (Figure 2).

However, the Crosby (represented by methods like TQM and Six Sigma) states that the optimum is equal to zero (Figure 3) [Voderembse88, p.718].

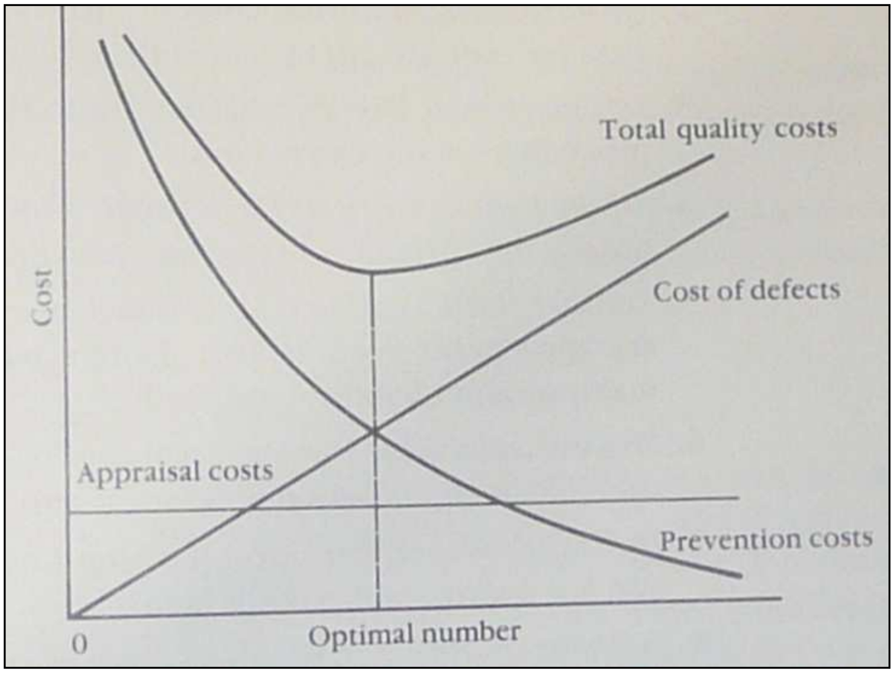


Figure 2 - The Optimum Good Enough Quality - Number of Defects and Cost

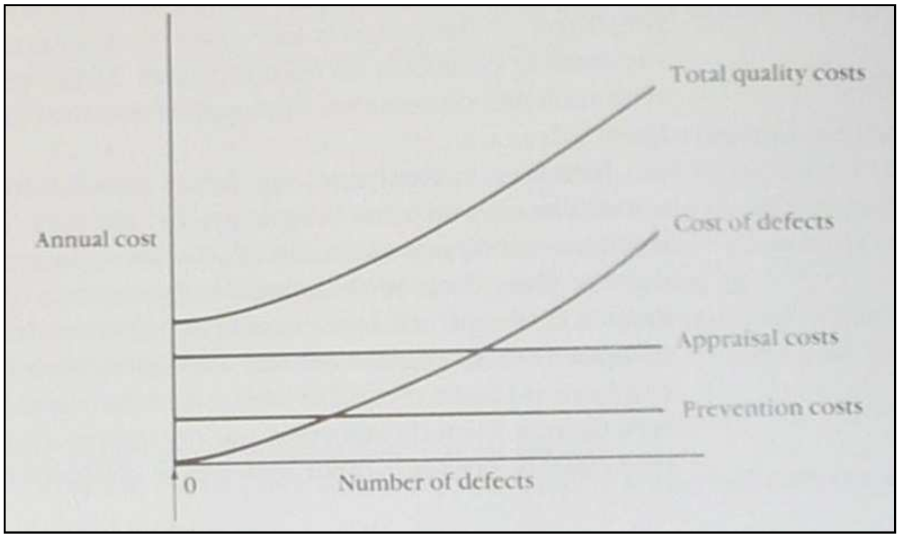


Figure 3 - The Optimum Quality - TQM

For the purposes of this thesis, however, the existence of the axiomatic concept of quality is questionable. The Goldratt's definition for the goal of a company can be linked back to the axioms of the microeconomics, where the <basic production equation> is defined as the <sales> equals <price> and <capital goods> (S=P+C). We can thus link the <sales> as throughput of the company, P as OPEX and C as the inventory. In the author's previous thesis the author defined the usability as "the cost to achieve utility" [Hätinen06], which is in essence, related only to the cost to use and the features of the product. The assumption used in this work is that the quality is not an axiomatic property, but can be derived from the process capability to achieve the throughput that satisfies the customer demand.

2.2 Experience Factory

One approach developed for the intellectual capital organization is the Experience Factory (EF) by [Basili92]. The idea of the EF is to divide the organization into a development organization which is improved by a separate EF –organization. This is similar model familiar to the Japanese manufacturing companies in the 70's, where a separate engineering organization issued TQM -support for the floor staff [Cooper95]. The original Japanese practice was derived from the inability of the traditional accounting systems to manage the efficiency of the knowledge work resulting into bloated support organizations and focusing the performance improvement and downsizing actions to the more easily monitorable manufacturing staff.

Despite the dysfunctional origins of the EF traceable back to the TQM shoring into the USA in the 80's, it has sustained some ongoing research activity especially in Fraunhofer Institute of Software Engineering, Germany [Althoff01]. The EF organization gathers empirical data, tools and lessons learned from the project increments, generalizes them, stores the lessons learned in an *experience base,* and gives direct feedback to the development organization. The objective of the EF is to provide a facility for the reuse of the collected learning. Thus, the EF acts as a logically or physically compartmentalized repository for structural capital enabling also the measurement of the structural capital assets (Figure 4).

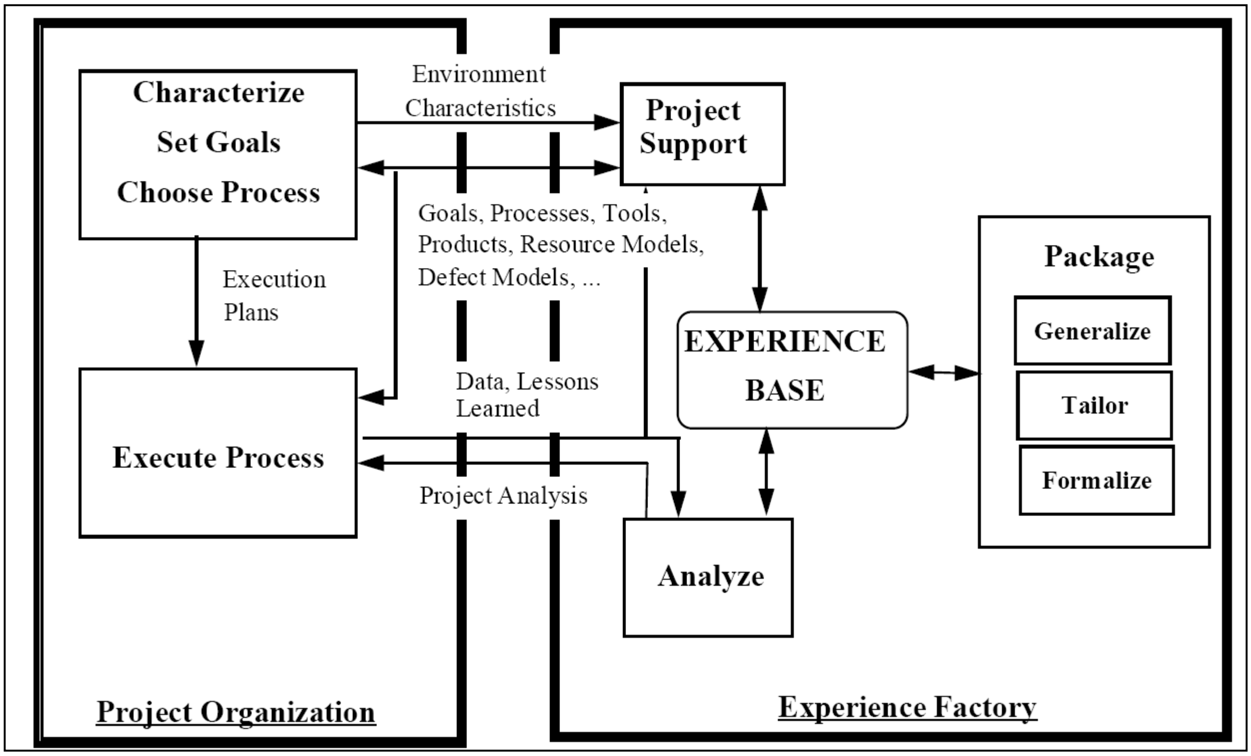


Figure 4 - Experience Factory [Basili94a]

2.3 Quality Goals

"Forgetting our objectives is the most frequent stupidity in which we indulge ourselves." (Nietzsche 1879, p. 642)

By the beginning of the last century in the manufacturing organizations, the craft (job) shops dominated the industry [Cooper95]. They were natural differentiators, who competed by producing products at high prices with high quality and functionality. Henry Ford introduced a new mass-production concept approximately from 1915 to 1925, which changed the competition field by making it possible to produce products at low cost and thus price, and forced the craft producers to move to toward upper-class customers [Ford22]. Topics such as cost leadership and product differentiation were regarded as the key strategic concepts to achieve sustainable competitive advantage. By definition, the best companies achieving towards either of the strategic ends will in practice create a "category killing" zones of no-competition, such as the monopoly of Microsoft in the Desktop OS market. On the 60's a new production philosophy of the lean production emerged, changing the competition once more by being able simultaneously provide high quality and functionality at low price. The origins of the lean can be identified to be first stated by Henry Ford [Ford22], but the original ideas have been conceptualized only as late as the 60's and 80's. Thus after decades of new kind of confrontational competition, in many industries, there are only lean companies remaining. The lean producers don't have anymore any sustainable competitive advantage, but rely on quickly adapting and exploiting emerging opportunities, thus competing continuously head-on by creating temporary advantages. The new requirement is the meta-ability to learn giving raise to concepts such as the systems thinking and learning organization [Senge90]. Cooper presents the Survival Triplet model to illustrate the survivable zones for lean companies at each point of time (Figure 5). The maximum feasible price, and the minimal feasible functionality, and quality are the lower levels that the customer. The other ends are the feasible production levels. The means for competition are for example reducing the cycle-time to introduce new functionality, and evaluating the rate of development vs. customer preferences in the each three areas..

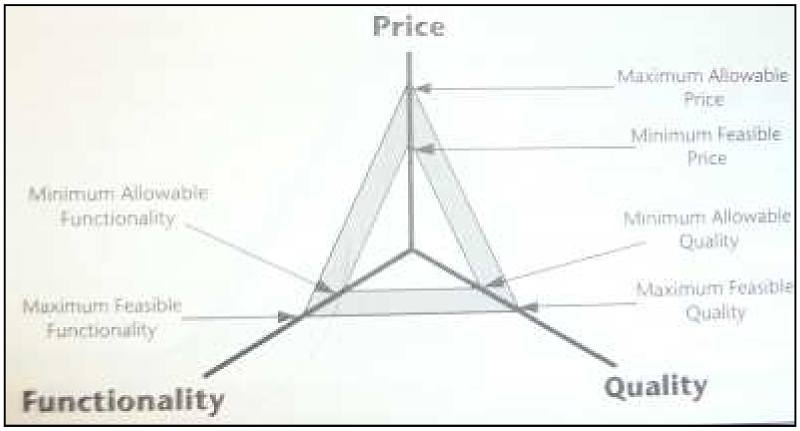


Figure 5 - Survival Triplet [Cooper95]

The software engineering methods have traditionally focused heavily on the management of scope (or functionality). This has unfortunately also resulted many times in bloated products that have far too many features for an ordinary user to utilize (such as Microsoft Excel). In the meantime the budgets of the projects are often overrun and the quality remains poor. The Target Costing is a traditional engineering method that has been used for at least a century in the construction industry [Haahtela07]. The idea of the Target Costing is to iteratively manage the cost and profit of a project by starting the planning with a coarse and fast estimate (e.g. the typical construction cost of a block house per m^2 in the capital area), and then phase by phase increasing the level of detail.

The Japanese extended the Target Costing method in the 60's by adding a similar process for managing also the quality related attributes. The Quality Function Deployment (QFD) process starts by evaluating the customer preferences and the competitor performance by asking the prospective customers (or by sensory "smell tests") to evaluate their current experience with the product using a two-column questionnaire [Akao90, p.39]. The questionnaire results are consolidated and used to fill out the demanded *quality deployment chart* (Figure 6). The most important quality goals are identified as sales points, such as "easy to hold". For these sales points, a product quality plan targets are set to differentiate the product from its competitors by setting improvement target. The required improvement is calculated by dividing the targeted by current goal level.

$$\rho = \frac{target}{current}$$

Finally the quality goals are weighted by calculating their proportional ratio of importance as a percentage of the complete product. [Akao90, p.149] suggests also using *Bottleneck Engineering* for analyzing whether the quality targets can be reached using the current technology. The Analytical Hierarchy Process (AHP) is suggested to be used for prioritizing the quality requirements. The QFD can be seen as a product design method specifically suited for lean organizations as it tries to constantly reposition the company in relation to the competition. Further, the improvement target setting process is quite closely related to yet one more Japanese extension to the Target Costing, i.e. the Target Pricing –outsourcing and product design method [Cooper95], where the component supplier is regularly (e.g. every year) given a new lowered price target. The supplier chain is managed by the client (typically a large company, e.g. Toyota) transparently helping the suppliers to reduce their costs by for example sending the client company's engineers to help the supplier to re-engineer their processes.

Akao emphasizes the systematic view of deconstruction of a larger whole into smaller sub-systems. The *quality function* means, according to Juran, functions that form or contribute to quality, such as sequential or logical planning and design activities. The *deployment of quality function* is a step-by-step objective process to develop the targeted quality level into the product. Akao argues that without using a systematic approach the customer requirements are often analyzed through a group communication and individual mental process and the factors such as the loudest vocalization gets often more weight than the actual customer requirements. Further, Akao introduces the concept of the Voice of the Customer into the method to represent the customer requirements for quality.

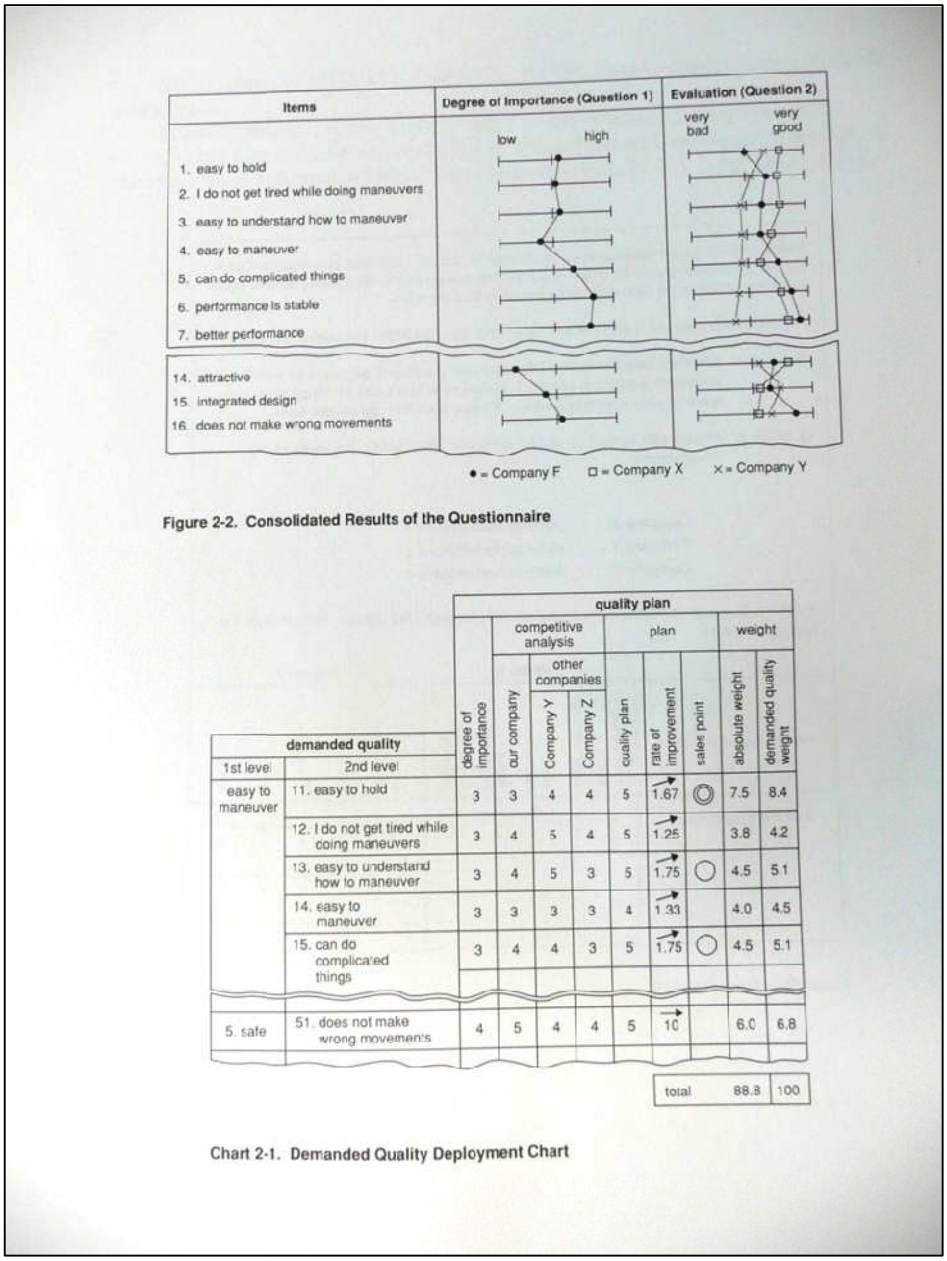

Items
Degree of Importance (Question 1)
Evaluation (Question 2)
low
high
very bad
very good
1. easy to hold
2. I do not get tired while doing maneuvers
3. easy to understand how to maneuver
4. easy to maneuver
5. can do complicated things
6. performance is stable
7. better performance
14. attractive
15. integrated design
16. does not make wrong movements
● = Company F
□ = Company X
× = Company Y

Figure 2-2. Consolidated Results of the Questionnaire

demanded quality		quality plan									
			competitive analysis			plan			weight	
				other companies						
1st level	2nd level	degree of importance	our company	Company Y	Company Z	quality plan	rate of improvement	sales point	absolute weight	demanded quality weight
easy to maneuver	11. easy to hold	3	3	4	4	5	↗ 1.67	◎	7.5	8.4
	12. I do not get tired while coing maneuvers	3	4	5	4	5	↗ 1.25		3.8	4.2
	13. easy to understand how to maneuver	3	4	5	3	5	↗ 1.75	○	4.5	5.1
	14. easy to maneuver	3	3	3	3	4	↗ 1.33		4.0	4.5
	15. can do complicated things	3	4	4	3	5	↗ 1.75	○	4.5	5.1
5. safe	51. does not make wrong movements	4	5	4	4	5	→ 10		6.0	6.8
							total		88.8	100

Chart 2-1. Demanded Quality Deployment Chart

Figure 6 - Consolidated Questionaire Results and Demanded Quality Deployment Chart [Akao90]

The TMap is a similar method intended for construction of a *test strategy* by choosing which testing practices should be used for capturing different kinds of quality defects [Pol02]. The participants are asked to evaluate how each testing method affects each quality character, producing a similar dataset which is used in this work. The novelty of this approach is construction of overview of the effects of the practices, which are not usually reported with similar wide breadth in the scientific literature. However, the relationships are based on the opinions of the participants and thus might not represent the reality accurately. TMap can be understood as a specific instance of the QFD –family of methods applied in the testing context.

The main promise of the TMap -method is to provide balance between the cost and benefits of testing by performing risk assessment. However, the method fails to document how the risk assessment step should be performed and starts from the evaluation of relative importance of the quality characteristics based on information that should be collected by the test manager. Pinkster provides the method called Risk & Requirements Based Testing (RRBT)–method that offers a detailed account of this missing step [Pinkster04].

The Quality Performance (QUPER) is a yet one more model to support road mapping of quality requirements [Regnell08]. The main difference to the QFD and TMap is that QUPER understands the relationship between the quality goals and the practices as nonlinear. To set the quality goals for the milestones, the idea of the QUPER -model is to first identify the utility, competitive and excessive breakpoints. The utility breakpoint is the minimum quality level that is acceptable for the customers. Below it the product has no value for the customer. For example if the battery of a mobile phone would have enough charge only for 5 minutes, it would render the whole device to be of little value for the user. The competitive breakpoint is compared against the best solutions available on the market, how to make the product to differentiate against its competitors. Third, the excessive breakpoint is the level of quality where the additional improvement in the quality does not yield any value for the user. For example it is rather the same for the user, whether a computer boots up in 0.1s or 0.01 seconds. Thus Regnell acknowledges the relationship between the quality and the benefit to be non-linear in nature, and provides the QUPER as a rough model to map the possibility space of the software engineering companies.

The second key concept of the QUPER is the barrier view, where the cost of the quality improvement is distinguished between the plateaus and barriers. A barrier for improvement is for example architectural spaghetti that prevents addition of new functionality before it is refactored with a substantial cost. Thus, once identified, it makes sense to set the quality goals by milestone in front of the barriers, instead of behind the barriers, enabling taking of full advantage of the current plateau of lower cost.

A method related to QUPER is the Cost/Worth Analysis by [Tanaka89]. The mapping of the barriers can be performed by estimating the relative worth and cost to develop the improvement on a chart (Figure 7). The components above the 45° diagonal can be understood as barriers or parts requiring cost-reduction, and the high worth/low cost items as plateaus that should be exploited further until they migrate closer to the diagonal.

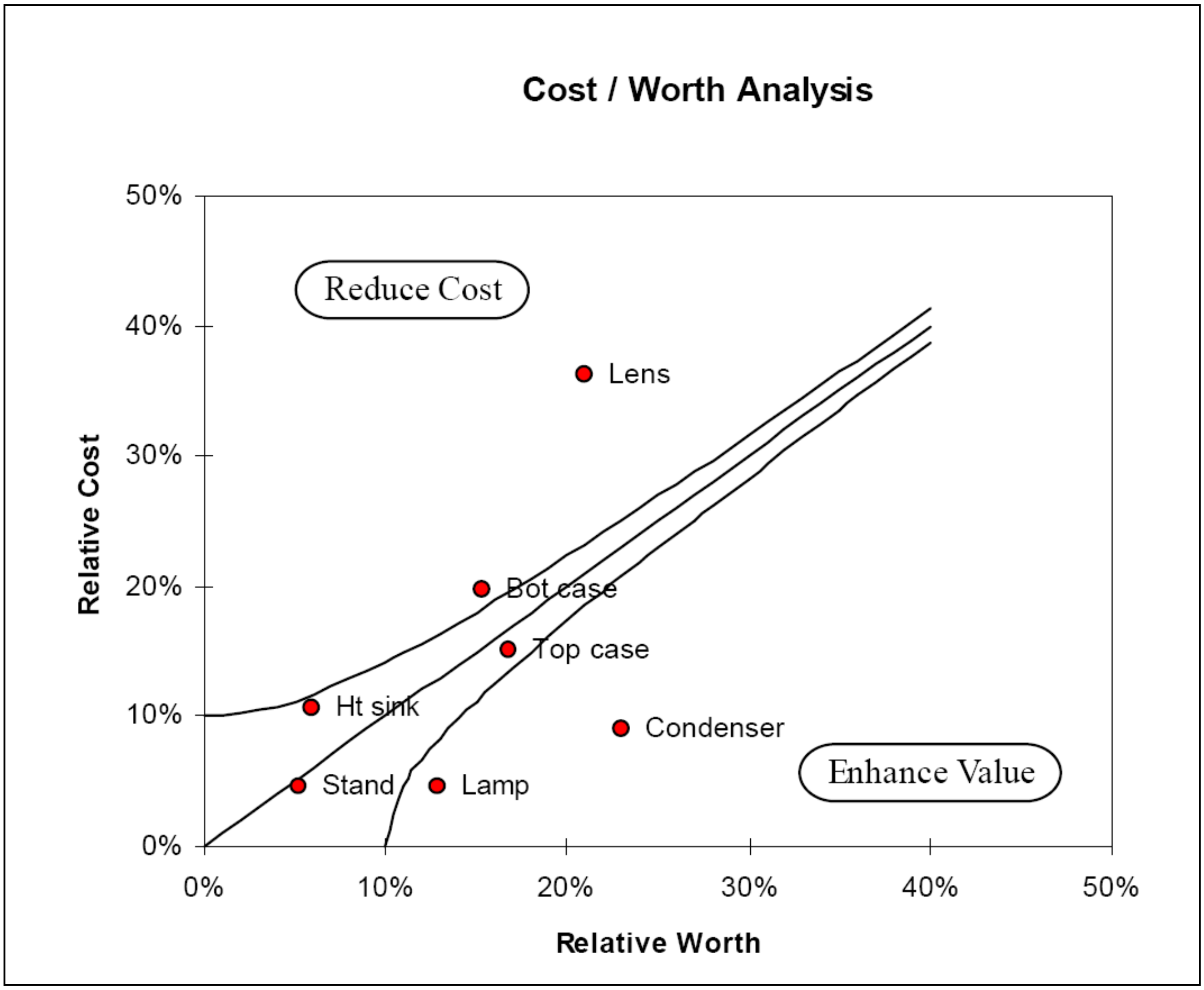


Figure 7 - Cost / Worth Analysis [Martin98]

Further, the relationship between the quality and customer satisfaction can be divided into two factors: attractive quality and expected quality [Kano84]. When the attractive quality is increased, the customer satisfaction increases, but decreasing it doesn't increase dissatisfaction (Figure 8). The opposite is true for the expected quality: increasing it doesn't improve satisfaction, but removes dissatisfaction. Near the origin exists so called neutral zone, where the changes in quality are indifferent.

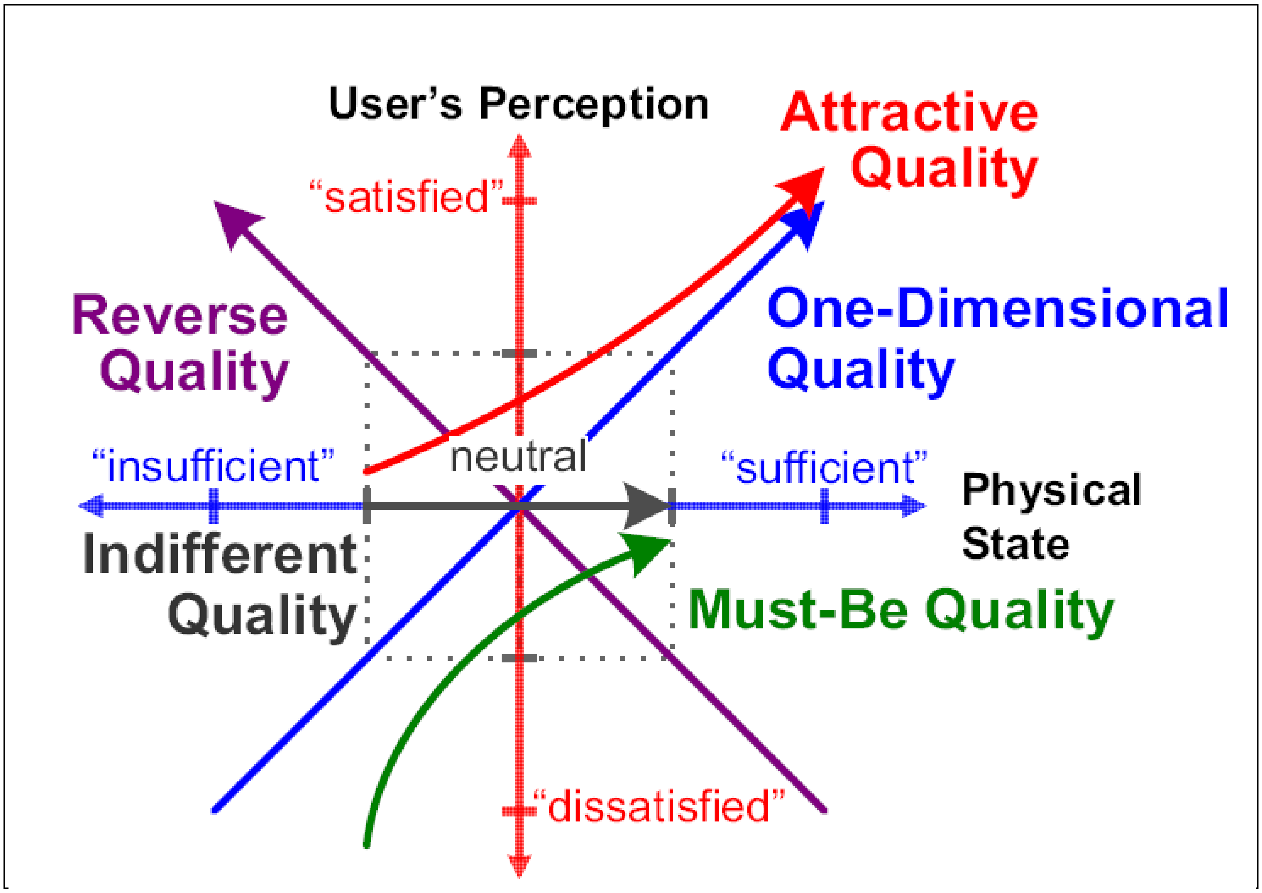


Figure 8 - The Kano -model for categorizing quality requirements

Additionally several standardized *ontologies* (explicitly defined conceptual categorizations of the world) such as the [ISO9162] and exist for categorizing quality goals. However, the standards need to be customized for specific contexts, and they don't provide guideline which metric should be particularly used by each organization. The author and the colleagues used the ISO9162 standard as an explicitly defined conceptual ontology to generalize the result metrics to make them comparable with each other.

2.4 Practices and Patterns

Some ambiguity exists in the nomenclature of software engineering between the exact definition and meaning of terms practice and pattern. [Alexander79] defines a design pattern as "*a rule which describes what you have to do to generate the entity which it defines*". Another definition for pattern is "*a solution to a problem in a context*" [Berczuk02]. The design patterns are an often applied concept in the field of software design, but also unfortunately the terms “organizational pattern” [Coplien95] and “management pattern” [Berczuk02] have been used with same meaning with practice in other fields.

However, the author feels the separate concepts useful to distinguish a technical and a social solution from apart. The design patterns have a strong etymology and association within the area of technical software engineering design, and thus should be used only for the field of design and architecture.

The concept of “best practice” refers to a technique, method, process, activity, incentive or reward that is more effective at delivering a particular outcome than some other. The practices are “benchmarked” by measuring the cycle times, cost, productivity and quality of a specific processes or methods. This notation is a closer to the social aspect of the solutions compared to the term “design pattern”. Further, in contrast to a “strategy”, which is a long plan to achieve a goal, the practices can be understood as components of the social (and to some extent also the technical) structure of an organization. Some practices might require or benefit from the utilization of technical tools, but these are defined by the social practice rather than vice versa.

The practices (organizational patterns) aren’t typically invented or created, but discovered (extracted) by empirical observation. In the empirical software engineering research, a vast number of studies exist describing the application of hundreds of variations of different practices, such as “formal software inspection” [Fagan76], “management code reviews” [AIII/Baker97] and “task-directed inspection” [AIII/Kelly00] as variations of the generic practice of “code review”.

2.5 Change Management

“Information is a difference, that makes a difference” – Gregory Bateson

While the temporal scope of this work does not allow study of the actual change within the subject companies, some findings from the literature are anyway reviewed. [Ford22] states that the efficiency of production is not in the interest of the worker, but improving the efficiency and quality of the production is the main task of the management. The management commitment is the key enabler for any change initiative. However, Ford regards the concept of management commitment as an abomination – if one is not interested in improving the production *“it indicates that the next jolt of the wheel of progress is going to fling him off”*.

Managing change in an organization is a difficult task. In the classical Lewinian change model, a system is unfrozen, then changed and finally re-frozen to a new improved state [Lewin51]. The Lewin’s model applies to the large changes of mechanistic industrial organizations. A more recent view of the organizational change is the autopoietic view, where the organization is viewed as a self-referential and self-preserving organism [Battram96, p.255]. As a metaphor the autopoietic theory suggests that a successful change can be achieved the best by a series of small steps with quick feedback, envisioned to lead in closer and closer to an *attractor*, the goal or end-state of the direction a system is moving to. In the autopoietic model of communication in comparison to the classical Shannon's transmission model of communication [Shannon49a] of senders and receivers, the players can be seen as constantly scanning the environment for anything that might of interest, such as the threats, opportunities, food or stimulation, and simultaneously looking for *resonance* in the other party of communication. As an example Battram mentions a child destroying a stereo very actively listening to the parents, if there is a possibility of something unpleasant to happen to them, but not actually listening to the words. Immediately when he hears "but this time I'll forgive you", the child immediately switches off. The communication happens in reference to the mental model of the parties, ignoring all of the communication received, except for the signals that might be of self-interest and aligned to the internal goals of the receiver.

The Senge's learning organization suggests *dialogue* (Table 2), a special kind of conversation with rules to allow all team members equal opportunity to bring forth their reality and forcing the others to actively listen and to understand the different perspectives without possibility to challenge the conflicting views, as one effective practice to search the possibility space [Battram99]. The role of the manager is to facilitate the dialogue by providing the means, space and training to perform a successful dialogue.

Table 2 - Communication with Autopoiesis in Mind [Battram99, p. 248]

1. Start by listening – identify the interests of your target 'systems'
2. Realize that communication isn't just sending a message, it's a process – establish resonance over a series of interactions
3. Tune the communication to suite the self-interests of the systems – it must be a 'difference that makes a difference' to them
4. Develop a consistent alternative model - another way of describing the reality

In comparison for example to the brainstorming [Osborn63], the objective of the dialogue is not to innovate or compromise, but to understand the goals of the other participants and how it affects their motivation, mental model and behavior. When used in the context of the model of autopoietic communication, the dialogue can be understood as a pre-requisite for a successful change initiative. The autopoietic model of communication fits also the so called Shannon's second theorem of information transmission , stating that a properly encoded message can be transmitted over a transmission channel no matter how much noise it is subject to, if the transmission channel is not overloaded [Shannon49b; Ward84, p. 18]. Thus, the reflection and resonance can be understood as the way to encode the message so that the distortions can be recognized and corrected and pass from one brain to another penetrating the filters of self-interest and the prevailing mental model.

Weisbord suggests the *future search* methodology as a structured approach to develop new ideas and possibilities with large groups [Weisbord00]. The idea is to draw in a variety of stakeholders who would normally never meet and to work together by reviewing the past, focus on the presence and mind-map everyone's perspectives publicly on to flipcharts. The idea of the *future search* is that nobody is required to change their mind, or give up their beliefs, values or commitments. Rather the session is expected to end up in 'confusion' and disagreement, but also outputting a wider view of the possibility space.

In comparison Schein suggests a successful cultural change can be achieved by destroying the artifacts (the techno structure) of the old culture and replacing them by a new [Schein99]. The key idea is to overcome the fear of change in respect to the anxiety of learning new. The both strategies can be applied simultaneously by highlighting the futility of staying with the old culture, shutting down the information systems supporting the old behavior, and by lowering the anxiety by offering for example training courses of the new computer systems. Of course, the change will fail if the change is not managed in a timely manner, i.e. not having the new infrastructure in place before the old is destroyed. Methods such as dialogue, brainstorming and future search facilitate mitigating the fear for change, as the mapping of the opportunity space is performed by the participants themselves.

In the context of this work, the interesting aspect of change management is how an organization is able to map the possibility space objectively and adapt the structural fitness to conform the competitive environment. The work evaluates two distinct approaches: the construction of an experience factory for benchmarking the quality capabilities of the best practices available in the industry and the usage of social search of the possibility space by using group workshops.

2.6 SPI Summary

The generic process of improving the efficiency of the structural capital of an organization starts according to Deming by improving the quality. The TQM –school can be divided into two major paradigms, namely the lean, which is closely related to the agile software engineering and the Just-In-Time/pull management systems, and the Statistical Process Control, where the Six Sigma presents the most widely accepted family of methodologies. The Lean school believes in modeling the production as a pull queue network, where nothing is produced in advance. The process improvement is performed by bottleneck analysis, Kaizen continuous improvement and the employee empowerment principles. In this work the first class of SPI methods or the social search methods are experimented according to the lean ideology. In contrast, the SPC/Six Sigma – school gives more trust in the statistical and scientific methods by collecting a large amounts of accurate data samples to be superior to the subjective opinions of the humans. The experience factory is an organization and a database wherein this data and analyses can be stored in and called upon on demand.

There is no consensus available which school presents the more effective SPI approach, although the Lean (Agile) has lately gained more acceptance amongst the smaller and young software engineering organizations as it gives power back to the knowledge workers to self-define the working environment and regards the SPC/Six Sigma as an unfunny remnant of the industrial age. However, the author believes that as the two sub schools of the overall TQM ideology, the both approaches can be applied in the future together, when the SW industry matures.

3. Research Method

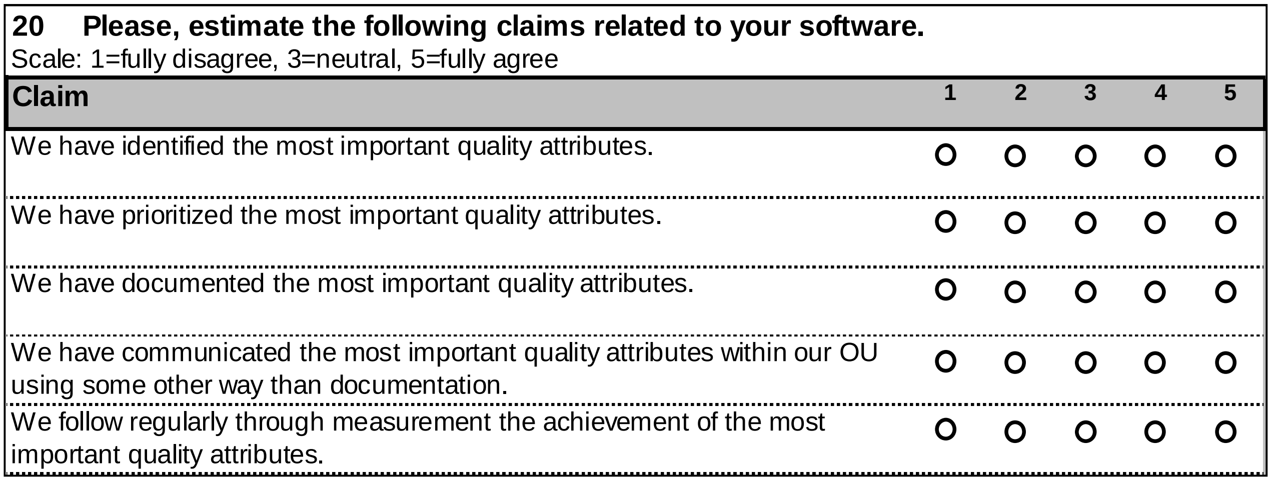
While the ESPA WP1 promises to deliver a concise SPI -method, many parts of the overall process such as the identification of goals, the prioritization of the development initiatives and the actual change management process to install the change initiatives in the organization were scoped out of the thesis. The main interest of this thesis is the selection of the quality practices based on the pre-given vector of prioritized quality goals.

The research method consists of three parts. First it is important to know what are the preferences of the subject companies towards SPI. Secondly the exact research questions are stated to study the problem at hand. Finally, the research method is discussed how the research questions should be studied.

3.1 Industry Requirements

The studied method is planned to be useful for the industry practioners. To understand the requirements of the subject companies a survey omnibus was installed on MASTO survey by the Lappeenranta University of Technology co-research group interviewing N=25 Finnish software companies (see Table 3). The results of the survey are presented in Table 4 [Luomansuu09]. In addition to the overall average result of all survey questions, the author calculated a comparative mean of the answers of the four subject companies. The column $P(H_0)$ states the asymptotic confidence level (probability) that the subset mean is statistically lower than total sample set mean using two-sided Chi-Square –test.

Table 3 - MASTO Survey Omnibus Questions



20 Please, estimate the following claims related to your software.
Scale: 1=fully disagree, 3=neutral, 5=fully agree

Claim	1	2	3	4	5
We have identified the most important quality attributes.	○	○	○	○	○
We have prioritized the most important quality attributes.	○	○	○	○	○
We have documented the most important quality attributes.	○	○	○	○	○
We have communicated the most important quality attributes within our OU using some other way than documentation.	○	○	○	○	○
We follow regularly through measurement the achievement of the most important quality attributes.	○	○	○	○	○

Table 4 - MASTO Survey Results on Prioritization of Requirements

Claim	Mean (All)	Mean (Subject)	σ (All)	σ (Subject)	$P(H_0)$
20.1 We have identified the most important quality attributes	3.73	3.50	1.11	1.29	0.37
20.2 We have prioritizes the most important quality attributes.	3.33	3.00	1.30	1.41	0.51
20.3 We have documented the most important quality attributes.	3.13	3.00	1.48	1.83	0.17
20.4 We have communicated the most important quality attributes within our organizational unit using some other way than documentation.	3.37	2.75	1.16	0.96	0.79
20.5. We follow regularly through measurement the achievement of the most important quality attributes.	2.97	3.00	1.35	1.41	0.07

The subject companies represent a more homogenous subset than the overall data, since the company A has roughly 60, companies B and C 120-130 employees and the Company D 500 employees. The amount of employees in the survey ranges from 4 to 350 000, representing a more heterogeneous sample. All four subject companies have international customers or operations, while the subject companies include also companies without exports.

It seems that except for the question 20.4, the subject companies seem to be less homogenous than the other companies, while the variance is somewhat larger. Also for the same question the hypothesis significance level is close to the 80% fractile indicating that it could be possible that the subject companies do not communicate the quality goals in other ways than by documentation.

For the motivation for answering the stated research questions, one must first understand the requirements of the practioners. The first applied method (QPA) was constructed without explicit prioritization of requirements by the colleagues. For purposes of this work and the development of new versions of the method the author extracted the colleague's implicit perception of the topic ()[3]. This information was used to perform the first three constructive interventions. Additionally the small steps and the employee participation were highlighted by the colleagues during the reflection sessions.

3.2 Research Questions

There are a very high number of goal setting methods and specific practices that claim to produce increased level of quality. However, a more interesting problem is to ask how reaching of a certain quality level can been guaranteed. As discussed earlier, the SPC regards a lower variance process as a high capability one, potential in using lower tolerances, benefitting from the smaller need for inventories and rework, and thus yielding lower OPEX. Thus the main research question in this work is the following:

> RQ: How can a software engineering company ensure reaching of a certain quality goal level?

In addition, specific sub-evaluation problems were stated during the research, which are described in the next chapter. One could call these problems as sub-research questions, but due to the selected explorative research method the author has chosen not to elevate them to the same conceptual category as the main research question. It should be noted that the progression in the action research cycle simultaneously improved also the research question. In particular the research question did not originally consider guaranteeing reaching of a certain quality level or a process capability. After the social search interventions (namely the QPA, IA and NMA) and the literature study it became evident that the original research question was not interesting enough for further study and a more interesting question should be provided. Thus a new more advanced research question (as represented above) was adopted and a further constructive intervention to develop the Semantic Web Experience Database was performed.

3.3 Research Method

Despite the refinement, providing an answer for the given question will remain difficult due to the lack of exact evaluation criteria what does the reaching of a quality goal exactly mean. This study has also

[3] ESPA Workshop 2.12.2008 Lappeenranta University of Technology

inconclusive research span to sample the effects of the proposed methods, which should be tracked over a time of several years, a clearly larger scope than what is possible within the resources allocated for this study. However, based on the literature study and the requirement surveys it seems that the starting point of the improvement process is the mapping of the possibility space. Thus the amount of generated SPI ideas is considered as the primary evaluation criteria for the developed methods. The second important consideration for the companies is the cost of application of the method. Many methods can be used, but some methods are more efficient and provide a better yield per invested man hour. It is possible to measure also several other metrics from the proposed methods, such as, how well they record the state of the current operations and the subjective opinions by the practioners. However, these metrics can be regarded secondary as they will only support the generation of the ideas, not providing new ones or evaluating the effectiveness of the possibilities.

The author has chosen two-fold approach of action research and constructive research for answering the questions. The basic paradigm underlying the both approaches is post-modernism, which tries to reveal and disprove the hidden assumptions of the research by pursuing to disprove authority by presenting rigorous critique. The objective of the study is to reveal the core of the hidden truth and to abolish the subjective myths surrounding the subject.

First, an iterative explorative research method was chosen inspired by Action Research [Lewin46]. However, due to the time constraints the author is unable to study the effect of the intervention in the subject companies objectively. Thus the topic of the research is the actual process improvement method and the research group rather than the target companies. Some subjective data was nevertheless gathered on the feelings of the subject companies. The research was performed in seven steps (see Table 5). Each step involved an exploratory constructive intervention step, where the evaluated method was modified to explore the possibilities for better performance. Since the optimal method was unknown, the explorative search strategy was chosen to map the possible solutions in an orthogonal manner based on the results of the literature study. Since it would be un-worthwhile to restrain one-self merely in repeating the research on already previously studied methods, the author chose to construct new methods based on the well known principles of prior study. The intervention in the action research means the researchers actively performing some change in the subject organization. In this work the interventions included both detailed and fundamental changes to the applied practice workshop method and were applied on the participatory workshops with the subject companies, but focusing more on the SPI method than on the subject companies themselves.

Table 5 - Action Research Process

1. Identifying the problems in the current method
2. Defining the research question and a hypothesis for the problem
3. Prioritizing the research questions by relevance
4. Designing a research plan to answer the 4-5 highest ranking research questions, selecting the research method, what data is collected, and how the data is analyzed.
5. Performing the research by the plan. Gathering data.
6. Reflective analysis to review the results and the validity of the study. Answering to the research question.
7. Constructive Intervention – Revise the method by the lessons learned. Study the literature or innovate new solutions to solve the problems.
8. Go to step 1.

The current problems, taxonomy and the research plan were constructed by a mind-mapping software enabling easy addition of research problems, questions, hypothesis and the answers. The mind-map items were prioritized by relevance and filtered to plan the next 4-5 studies. The iterative research questions are presented in Table 6. The method –column refers to the evaluation method of the problem. The objective of the evaluation is to provide a boolean answer with a rationale. The evaluation methods include a workshop experiment, where a new method construction in applied in a realistic company SPI workshop with cross-functional representatives of different roles of the development organization. Before the experiment the author and the colleagues formulated an evaluation criterion that was assessed after the experiment on a reflection session. A workshop control study is an evaluation, where the results of a normal workshop with a different primary objective are evaluated against a predefined evaluation criterion. The data is produced as a side product of the main agenda, such as performing a QPA WS. A post-analysis refers to analyzing the transcripts, recordings and the artifacts of one or more workshops against a preselected evaluation criterion.

The criteria is an evaluation statement that returns a boolean true or false. For example the functional – criteria is formulated at the reflection by consensus of the colleagues. At the control study the amount of new recorded practices provided by a control person (for example an employee of a role that has not participated to all workshops) are evaluated against a predetermined criterion.

Table 6 - Iterative Evalution Problems

Evaluation Problem	Subject	Method	Criteria
1.1 Should the "Indicator Analysis" be used for practice selection?	Company C	Workshop Experiment	Is functional?
1.2 Should the "New Method A" be used for practice selection?	Company D	Workshop Experiment	Is functional?
1.3 Does the absence of the employees of primary role affect results?	Company A	Workshop Control Study	New practices < 6
1.4 Should the QPA be used for practice selection?	Company A Company B Company C	Workshop Experiment	Is functional?
1.5 How the goals should be documented to support QPWS?	Company C	Workshop Experiment	Is functional?
1.6 How the practices should be categorized for SPI?	Company A Company B	Post Analysis	Is functional?

The research methods included interviews, observations and surveys. However, due to the fact of small sample set, the reliability of any statistical analysis will remain poor. Thus the emphasis is placed on more on case study and exploration.

The action research cycle culminated in the final constructive intervention of developing an evidence based software engineering database (EBSE DB) prototype based on the latest *Semantic Web* technology (described in Chapter 4.2). The constructive research method is perhaps the one most often used in software engineering, trying to solve the problem by constructing a system, algorithm or a theory, and validating it against practical or epistemic evidence (Figure 9). As a side product the EBSE DB contributes also as the first constructive step for advancing the ESPA WP1.3 of building a quality measurement and information utilization framework.

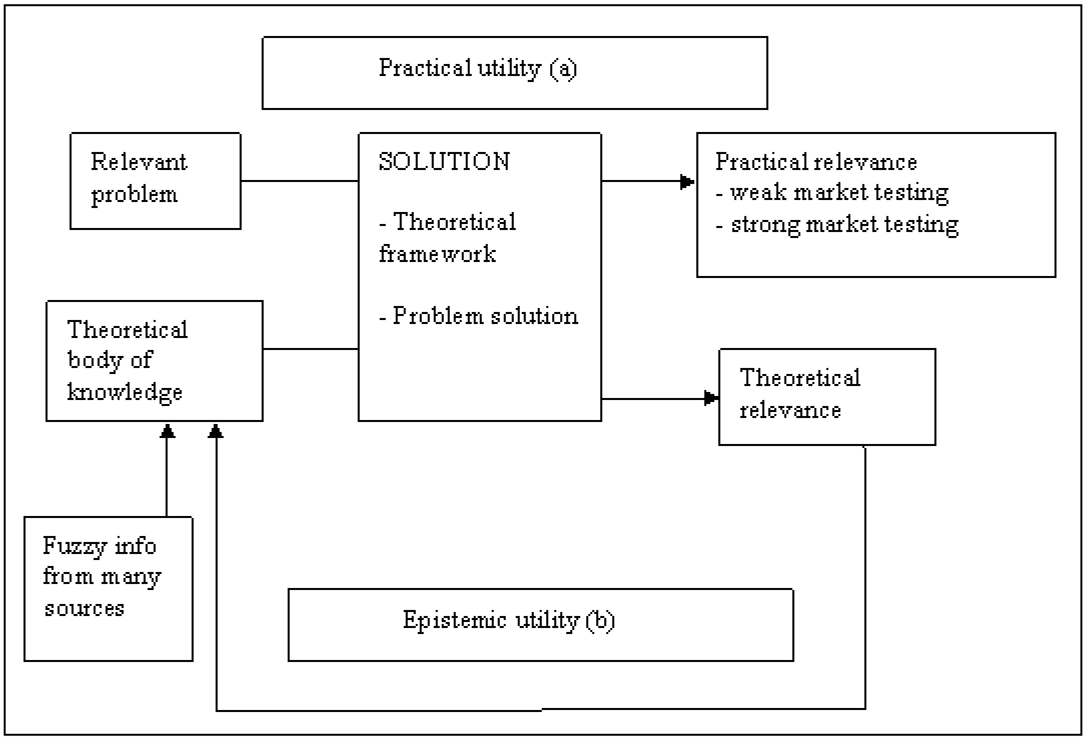


Figure 9 - Constructive Research

The validation of the constructions is finally performed by comparing the Improvement (I)–matrix provided by the social search techniques to the ones given by the semantic web database. The resulting intersection of the comparison is evaluated to choose the construction that provides the best alternative method for answering the main research question. The comparison is performed by the quality goals. Finally the comparison result is discussed to evaluate which approach provides the best solution.

A survey was planned to validate the assumed prioritization of requirements and conducted on the ESPA Experience Sharing Workshop (EESWS) by sampling participants of the all four subject companies by issuing a questionnaire for the participants (see Appendix I). The research team had initially listed the self-produced priority assumptions at the beginning of the QPWS project (). The EESWS validation survey questions were randomized to minimize the error caused by the order of the questions. A pilot survey was conducted on the ATMAN research group at the SoberIT, who is developing the Agilefant.org project management software. The results of the survey (N=7) are presented in Table 8, excluding the pilot survey.

Table 7 – Initial Assumed Prioritization of Requirements by 12/2008

1. Effective Quality Improvement Results
2. Software Process Improvement (SPI) step is small rather than large
3. SPI ideas can be implemented easily
4. SPI ideas have low risk for negative side effects
5. SPI method is lightweight to use
6. The users of the SPI method are pleased to the results
7. Method uses existing quality information as input

Table 8 - EESWS Results on Prioritization of Requirements

Requirement	Sum (Weight)
1. Most Effective Quality Improvement Results	35
2. SPI step is rather small than large	18
3. The SPI method is lightweight to use	18
4. Implementability – the SPI ideas have support amongst the employees	12
5. SPI ideas have low risk for negative side effects	12
6. The method uses existing quality information as input	9

All participants chose the effectiveness of the SPI initiatives as the most important characteristic of the method. Thus even though the N was small, the result has some significance, since it's unlikely that all participants would answer uniformly on a randomized question battery. The validity of the questions is however questionable, since it is unlikely that they represent a complete set of indicators that would cover even a substantial part of the problem at hand.

To discuss the results, it seems that the initial assumptions by the researchers reflect well also the actual opinions of the subject companies. An additional insight is the gained interval data of the relative importance of the EESWS result items. It seems that the utmost important factor is the effective improvement of the quality, while the other factors have much lower impact. To reach the desired quality level, it would be nice that the steps involved are small and the method is not costly to use, but these items don't matter if the quality can be reached by some method. During the constructive interventions a significant emphasis was placed on the implementability, meaning gaining of the support of the employees using participatory methods. However, it seems clearly that this factor is of even less importance from the subject companies' point of view, as their main objective seems to unanimously be to increase quality. The other factors seem to be only obstructions in the path of reaching of this ultimate goal.

3.4 Related Work

A multitude of methods exist for software process improvement (SPI) from the quality perspective. The first evaluated method the Quality Palette Analysis is conceptually very closely related to the Japanese *Quality Function Deployment* (QFD) –approach from the 60's [Akao90, Zultner93] and it's western adaptation of the *House of Quality* [Hauser88]. An even more closely related work is the TMap test planning method that provides a logical process for selecting testing practices and promises improved efficiency by utilization of risk assessment and effort comparison. It seems that TMap has also many ideas that have inspired the colleagues in the construction of the QPA-method.

Other often quoted SPI methods include the Goal-Question-Metric (GQM) [Basili94b] that is an adaptation of the more widely known management practice Analytical Hierarchy Process (AHP) [Saaty80] (although Basili never refers to this connection). AHP compares pair-wise the decision points and assigns a numerical weight for the different problem hierarchy items. AHP has been extensively studied and used as a part of the other methods such as Six Sigma and QFD for quality management and other purposes. The difference between the GQM to the AHP is that Basili uses predicated software metrics instead of the pair-wise comparison of alternatives following a numerical analysis. Although a significant overlap exists between the two methods, some sources [Kontio96] regard the QGM to be useful in deriving the initial high-level AHP decision tree.

Yet another approach to SPI is the maturity model such as CMMI[4] and SPICE [Dorling93]. However, the maturity models have been criticized from not matching the actual critical success factors of the companies that drive the profitability [Fitzgerald99] making the heavy weight maturity model –paradigm to lose ground against the more efficient TQM -based lean and agile approaches since the 90's. The reason is grounded in operations management theory as described earlier – the good enough quality is not enough when the lean organization outperforms the traditional mass producers in the pace of quality improvement and the efficiency.

A more interesting and less researched point of view is the construction of databases for empirical evidence on practices. At least two software engineering practice knowledge databases have been suggested and implemented by various international authors [Shaw07[5], Janzen09[6]]. However, these databases lack semantic

[4] http://www.sei.cmu.edu/cmmi

[5] https://bpch.dau.mil

properties and functionality, and are thus more like wikis or article indices rather than industrially useful tools. Fraunhofer has implemented a case based experience factory featuring case based artificial reasoning [Althoff01]. It seems that the Fraunhofer database involves entry of primary case data compared to the secondary data used to populate the two other databases. Yet two more databases (ERR[7] and CeBASE[8] [Boehm01]) have been constructed and referred in literature, but seem to be inaccessible at the time of writing of this thesis and are probably discontinued. Many similar semantic web based approaches have been adopted on other fields such as in evidence based medicine [Gao05].

3.5 Research Method Summary

According to the survey performed by the author, the subject SW companies unanimously regard the effectiveness of the SPI method as the most important property, before for example the efficiency. The temporal footprint of the method is of secondary importance, but sets a budgetary constraint for the application of the SPI method. Thus the method should provide a substantial increase in the quality with a low cost to use. The author has selected a constructive-action research as the research method by iteratively stating new research questions, designing a research setup by performing a constructive intervention, and evaluating the results in the context of the subject companies.

According to the related work a substantial amount of TQM SPI methods have been suggested and studied earlier. The closest ones include the QFD, Goal-Question-Metric, the maturity models and the EBSE databases such as the BPCH and the SEEDS. The Author has chosen to experiment on one QFD-style method (the Quality Palette Analysis/QPA), a management-by-objectives/GQM –style analytical process (the indicator analysis), a lean brainstorming method (NMA), and finally constructing an EBSE DB using the semantic web technology, which seems to have been not used prior in the context of the EBSE.

[6] http://evidencebasedse.com
[7] http://www.dur.ac.uk/ebse/repository/evidencebasedlitreviews.php#whatlearnt (inaccessible)
[8] http://fc-md.umd.edu/cebase-faq/answer.asp?questionid=77 , http://www.cebase.org (inaccessible)

4. Results

Next the research process and the findings are described in detail. Prior to this work the colleaguesconducted the Quality Goal Workshop (QGWS) at each of the four subject companies to find out the prioritized goals to be used as givens for the Quality Practice Workshops (QPWS). For non-disclosure purposes, the author performed a coding for the subject companies and informants. The companies are referred with capital letters from A to D and the informants of each company with a trailing number (for example informant A1).

4.1 Quality Practice Workshop

While the QGWS was based on a literature study, the first evaluated QPWS -method was loosely based on the Quality Palette Analysis (QPA) [Itkonen07] unpublished workbook and the experience of the colleagues. However, it was not applied exactly as described in the original source, but incorporated many new characteristics from the earlier Quality Goals Workshop (QGWS) phase such as the pre-assignment and the workshop. The QPA -method utilizes a similar goal prioritization scheme to the TMap, but both methods failed to take account the inherently hierarchical structure of the goals. It seems that neither TMap nor QPA have paid full attention to Akao's original work of the QFD. On the TMap's behalf the sanity of the linear prioritization choice can be somehow justified by the reference to the risk assessment, but as discussed earlier, the TMap -book fails to describe this step in detail.

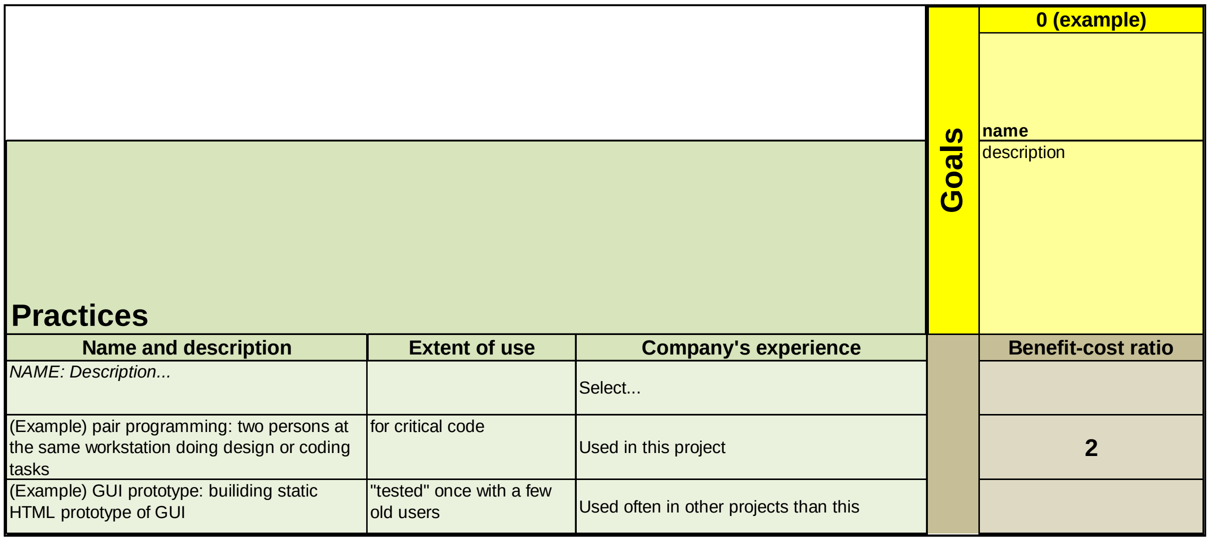
The original hypothesis of the QPA was to perform a graphical matrix analysis by looking for "anomalies" in the matrix for goals that lack proper practices. The unpublished QPA workbook does not specify how the matrix data should be collected, but provides an analysis method. The hypothesized problems are listed in Table 9. The analysis was, however, never performed as described in the original source, as after the first workshop it became evident to the colleagues that the matrix will not have "holes" in it, but the companies already stated to have at least half a dozen practices for any given goal. For the record, the early hypothesis is described.

Table 9 - QPA Analysis Anomalies

- Quality goals that have no good practices
- Practices that do not contribute to any quality goal
- Practices that contribute to many quality goals
- Practices that are not used
- Frequently used practices that are perceived helpful or auxiliary only
- Good practices that are not used or only occasionally used

When an anomaly is found, the practioners are suggested to find new practices, adjust the existing ones, or to enforce and improve the usage of previously chosen practices. Further, the application of short-cycle iterative and incremental development is preferred by evaluating whether the practices are “in-sync” or “out-sync” with the daily, iteration and release cycles. The practioners are suggested to model the quality practices by three different perspectives; the purpose, attitude and roles. The purpose is a description whether the practice is focused on preventing or detecting the defects. The attitude describes whether the practice is constructive or destructive in nature. The attitude -classification seems to have substantial overlap with the purpose, since the constructive practice is defined as a quality builder and the destructive as defect detection. Finally the role defines who is performing the practice and whether this person is for example in-house, shared, dedicated or outsourced. In contradiction to the original QPA, data on the anomalies, synchronization and the three perspectives were not collected during the QPWSs.

Based the experience of the previous phase, the colleagues planned a workshop method to be used for the construction of the Excel-based matrix. This step introduced a range of developments to the original description. Instead of using the plus and minus signs for indicating the strength of relationship between the goals and practices, the colleagues chose to use a range of 0-3 representing the combined benefit/cost -ratio instead of the contribution effect. A pre-assignment was adopted from the previous QGWS. As during the earlier phase, the workshop was justified by incorporation of a participatory process, thus increasing the acceptability of the resulting SPI initiatives and reducing the resistance for change. The companies were asked to choose one person from each personnel group to act as a representative to the workshop. The prioritized goals were taken as a given input from the preceding QGWS. Figure 10 shows the initial pre-assignment -sheet that was sent to the participants of the Company A and which was used as the main artifact for the whole process with gradual modifications.



Practices

Goals

0 (example)

name

description

Name and description	Extent of use	Company's experience		Benefit-cost ratio
NAME: Description...		Select...		
(Example) pair programming: two persons at the same workstation doing design or coding tasks	for critical code	Used in this project		2
(Example) GUI prototype: builiding static HTML prototype of GUI	"tested" once with a few old users	Used often in other projects than this		

Figure 10 - QPA Pre-assignment Template for Company A

4.1.1 QPWS Company A

The Company A is a software company delivering ERP software for electrical power plants and the energy business. The Company A has roughly 110 employees in three countries. The initial QPWS had a total of 8 participants including the CEO. Two subsequent workshops were held mostly with the informant A1 (Senior Manager for Software Development Projects) and A2 (Lead Test Engineer), while informant A3 (Project Manager for Export Deliveries) participated occasionally. The project scope was defined as the export delivery projects, which actually mostly consists of product configuration, consultation and delivery, and actually does not include direct R&D efforts, except for the purpose of product adaptation necessary to fit to the local market. An additional difficulty is the layer of consulting partners on the export markets, who act as proxies in communication between the Company A and the customers, causing new challenges for the projects and the R&D that cannot be solved using the traditional direct development practices proven successful on the domestic markets.

The first QPWS was held on the Company A in Nov'08. Before the workshop, a pre-assignment was sent to the participants. The task description of the pre-assignment was to list the practices affecting the previously found 10 quality goals by listing the name &short description, the extent of use, and the company's experience in using the particular practice. However, for the Company A, the description, extend and experience fields were initially hidden on the provided sheet. They were later changed to be displayed by default for the Company B. Also for Company B the name & description field was expanded into two separate fields. The researches received before the starting of the workshop in total of 39 practices or ideas filled on the submitted Excel-sheet matrix and the estimates of the contribution of the practice towards the quality goal on a scale of 0-3 (legend 3 =large, 2=moderate, 1=small, empty = no effect). At the beginning of the QPWS, the result of the previous goal workshop was presented and a few clarifications were made by the colleagues to modify the so called "false goals" to fit the categories of the ISO9162 -quality model. For example the previous goal title "the throughput time of a version order" was relabeled as "installability", and the original goal was added as an indicator under the higher level goal category title by the colleagues. The workshop continued by reviewing the results of the pre-assignment. The colleagues had listed and grouped the pre-assignment answers on an Excel-sheet, and asked, if the grouping and the contribution effect was correct. For example, the practices "code reviews for critical functionality" by A1 and "code reviews" by A2 were merged during the workshop. The information A1 had stated the practice is not used by the company, but the informant A2 claimed it was used sometimes. The merged practice retained the status not used with the specification for critical functionality. It was later migrated on the idea backlog by the author as it was not used in the company currently. Already on the pre-assignment a high number of development ideas were listed by the participants, causing the colleagues to check the extent of use on the workshop one-by-one, whether the practice was used often in this project, other projects, sometimes, or not at all. The walkthrough of the practices was performed by the results of the pre-assignment by filling the missing fields and re-evaluating the contribution value to each goal. The pre-assignment caused further need for clarification and merging, while different participants had listed the effect on a different number, for example A2 had estimated the contribution towards the goal "installability and updateability" of the previous goal as "2", while A4 had estimated her version as "3". After discussion and merging on the workshop, the merged practice "knowing of customer options" got impact of "3".

The colleagues had also identified three "false-practices" listed on the pre-assignment and relabeled them as goals, for example the "updating the software should be easy and instructed". They were simply filtered out and not discussed later as new indicators or elaborations for previous goals. Also the practice "project contracts" was filtered out by the colleagues although it had been marked as contributing by the maximum factor to the 3rd highest ranking goal of the project scope as this was regarded as a non-quality practice by the colleagues. As a sub-research question the colleagues studied if there would be a suitable categorization for the practices, and a few different ones were attempted, such as the "XP categories" by the author and the "SW engineering category" by the colleagues (JI). However, after the Company C QPWS these classification ontologies were deemed to be useful only as internal aids for the researchers, but not for the participants. On the contrary, the increase in the amount of fields and instructions (e.g. the categorization) only caused confusion and difficulty to understand how the method should be used[9].

During the reflection of the workshop, the colleagues noted that the ideas and the practices were mangled on the QPA-matrix[10]. The time ran out in the workshop, because the matrix walk-through temporal consumption rate of the matrix grew exponentially O(C•P•G) ~= O(N^2) by the addition of each new column C, practice P or goal G (capped originally to the top 10). The colleagues decided to change the walkthrough order orthogonally from the "all goals by practices" to the "practices by the top 3 goals" for the next QPWS limiting the consumption rate to 3•C•P ~= O(N^2). The objective of the workshop was also to select which practices would be selected for process improvement, but due to the schedule problems this phase was performed rather superfluously due to the schedule problems caused by the still remaining exponential footprint.

Three more QPWSs were held at the Company A, where the practices, contributions and indicators were further elaborated and clarified. After filling out the QPA-matrix, the participants were asked to evaluate the current state of the practices in use for the top 3 goals based on their expertise, for example to the question *"is this practice palette sufficient for reaching the target indicator level, if the indicator is throughput time per update?"* On the final QPWS a further problem was identified; on what (kind of goal-practice relationship) is the estimate based on and who has defined it? However, this concern didn't lead into any direct changes to the method or further investigation.

After the workshops held with the colleagues, the Company A had used the QPA –matrices in an internal SPI meeting[11]. The topics included for example aiming for the full-automation of the functional test suite, increasing the usage of unit testing and collection of data on change requests after project deliveries. The participants noted that they had used and updated only the indicators and the practice list. The SPI idea backlog, i.e. the list of ideas for new practices, was not used.

4.1.2 QPWS Company B

The Company B is a Microsoft certified consulting and Internet-software hosting company with 50 employees and roughly 5M EUR in revenue targeting business to business markets. The company was

[9] QPWS Company C - The author and JI concluded that showing the categorization caused confusion amongst the participants.
[10] Mika Mäntylä 20.1.2009
[11] Company A Status Beat Meeting Diary 24.2.2009

founded in 2000. The main participants of the QPWS were the R&D Manager (B1) and the Quality Manager (B2). Additionally the author communicated with the UI Designer B3 by email.

On the Company B QPWS the pre-assignment was performed independently by the company in a novel way surprising the colleagues. Instead of filling out the results personally, the company held an internal meeting to list the currently used practices, and after this 4 participants returned their personal estimates of the contribution towards each goal. The positive effect of this was that there was no need for merging of the practices. However, a new difficulty was the conflicting goal-practice contribution estimates. The researchers listed the pre-assignment estimates from the four informants and discussed the value of each during the workshops with the quality manager and the R&D manager. The estimate had large variation between the informants. The colleagues reflected also that there was no clarity whether the contribution values were representing the current state or estimates about the future after the improvement ideas would be implemented. Instead of for example calculating the mean, the managers re-estimated the final value based on the pre-assignment and their personal experience, by the request of the colleagues.

The meaning of the evaluation (0-3) integer value changed from the initial workshop's meaning of "benefit/cost ratio" to "effect towards a goal". It seems that it was cognitively too complicated to try to force the participants to evaluate a complex evaluation ratio of several factors, and thus a semantically simpler criterion was selected. The following new columns were added to the QPA matrix –sheet after the pre-assignment; "needs improvement", "is being improved", "needs help from the researchers". The idea of the new columns was partly to cope with the semantic division, but it seems that they failed to capture the original cost-dimension in any meaningful manner. However, the amount of evaluation work to be performed per data point increased from asking three to six questions. This might be further contributed to the fact that also this time the time was insufficient for filling out the complete quality palette and two more workshops were required to reach the end of the phase, although some time was consumed also by the Company B managers re-explaining their processes to the author. The colleagues noted that the participants listed more improvement ideas than current practices, while the objective of the workshop was to list mainly current practices for the quality palette analysis. It seems, however, that the change in walk-through order from "*practices by goals*" to "*goals by practices*" increased the efficiency somewhat, despite the new modification was even heavier than the original at Company A.

4.1.3 QPWS Company C

The Company C is the industry leading provider of a maritime design and operations software. Founded in 1989, the Company C has 115 employees with 12.7 M EUR revenue (2007) and offices in 6 countries in Europe and Asia. The scope of the project was the next 6 month release of the 3D structural design software.

As a difference to the previous workshops, the goals set by the Company C were more related to success factors of the company than the current problems, which were interpreted by the colleagues to have been caused by the switch of context from project to product scope [Vanhanen09]. The author and JI found out on the Company C QPWS that displaying of the practice categorization –instruction confused the participants, and it was later discarded. A new approach to the pre-assignment was to not use it as the inputfor the workshop, but the participants were asked to report the pre-assignment first on a blank sheet. Also this time the time ran out, but only one further workshop of 1h was required to fill out the contributions towards all 9 goals. The author performed the first process intervention and the colleagues a second one by the request of the subject, which are described next.

On the second QPWS at the Company C the author contributed to the method for the first time by inventing a novel method for the practice selection by name Indicator Analysis (IA). Based on the literature study and the dominant organization culture at the Helsinki University of Technology, the author suggested Management by Objectives [Drucker54] –paradigm (MbO) as a solution to solve the problems of the QPA –approach. The MbO and the Scientific Management –approaches have strong research evidence to link the high white collar employee productivity with the management practices [White81]. The IA has considerable conceptual affinity with the concept of the "quality function", despite this relation was unknown at the time of the construction of the method. The IA in essence is a method to build a linear model of characteristics that contributes towards and against a quality goal. In contrast to the Akao's QFD -method, the IA tries to capture also the negative relationships. The deliverable of the method was envisioned to be a breakdown of the largest affectors towards the quality goal, or a quality function similar to the QFD -model.

During the one hour session, which was recorded and later transcipted by the author, the participants were asked first to choose the best indicator of one of the top goals for the practice evaluation, instead of the general ISO9162 category. By the suggestion of the author, the participants chose the indicator "hours to design and to provide material for class approval of a standard product using the design software". The participants estimated the current level to be several dozens of man months and the target level to be reduced by roughly 1600 man hours, based on the customer and competitor benchmarks. The participants noted that this indicator is one of the main sales point for the company, on which the Company C has held traditionally an edge over its competitors, but further improvement would be constantly required. The session produced in total 13 estimations (5 positive, 8 neutral) on the previously identified quality practices.

After this the participants were asked to first estimate the positive/negative direction of each practice towards the goal. The second phase was to estimate the contribution in units of the indicator (hours) toward the indicator (goal). For example, the participants were asked to evaluate how many hours the practice "competitor benchmark" contributes per release towards the number of hours it takes to design the standard product. However, the result of the session was discouraging; the participants were unable to give even a rough estimate. The only value the author was able to collect was the direction of the effect. However, while the author noted the participants discussing one practice "adding new features" to contribute negatively towards the selected quality indicator, when asked explicitly the participants denied the existence of the negative direction.

> *"To speak completely honestly, this is a that kind of thing, which should be followed, while the new features are implemented, since it causes in average, if not put enough emphasis on real and big models and while developing new features, our performance to keep slowing down. This has happened many times."* – Developer C5 / QPWS Indicator Analysis transcript from recording, original in Finnish.

During the reflection, the author speculated that the cause of the failure was probably related to the lack of data available for the participants, while they were unable to estimate the effect without doing a proper research on the subject. However, the colleagues noted that the participants could be able to do such a study, if they would use a small amount of resources to dig the historical records of the benchmark values. This was also suggested as an action point for the company during the workshop.

By the suggestion of by the informant C1 (product quality manager), the researchers were asked to apply the QPA method to a more specific iteration scope, resulting in a new method called "QPA for Features" (QFF). The idea of the QFF is to create a new quality model for each feature, while it is unlikely that different parts (subsystems) within the product will have similar quality requirements [Kitchenham97]. However, it is uncertain whether the quality models should be constructed by features (as suggested by the Company C) or by sub-systems (as suggested by the colleagues).

Informant C1 commented that using the QFF at the concrete level with concrete goals and features enabled the participants to identify new practices to meet the quality goals (EESWS). The participating colleagues

(JI/EESWS) interpreted that traditional task planning (on for example MS Project) does not facilitate the invention of new practices to meet the quality goals, but QFF resulted in a number of new SPI ideas such as the "screenshotCaptureVideo", or recording a screen capture video of the user test. The method can be understood to reflect the MbO –approach with a new perspective of focusing on product/project/release/iteration –scope instead of the traditional business unit/department/team/person –division.

4.1.4 QPWS Company D

The Company D is a publicly listed software company founded in 1966 targeting utilities, energy and construction companies. The Company D employs 450 people in 12 countries producing 60M EUR in revenue. The project scope was the electricity network planning and management software product.

To solve the problems of the QPA and Indicator Analysis, a new approach by the working title "New Method A" (NMA) was performed at the Company D. Based on the literature study, the author identified brainstorming [Osborn63] as a participative method to generate new process improvement ideas. The mapping of current practices was discarded as it was seen by the author as an unnecessary step for the company, which normally knows which practices it performs, and mapping of the practices also causes unnecessary weight to the method. Further, four helping questions were formulated based on the quality improvement methods including the Management by Objectives [Drucker54], Six Sigma [Harry00], the Theory of Constraints [Goldratt84], and the Mythical Man Month [Brooks95], see Table 10.

Table 10 - New Method A – The Four Helping Questions

1. How could the current situation be improved the most?
2. Which current practices can't be obeyed?
3. What prevents/limits reaching the goal?
4. What could be done earlier?

The effect of the helping questions remains unclear, since the colleagues observed the participants to be more likely to list ideas that had been previously presented in the company, rather than to invent new ones. The contribution of the helping questions towards invention remains inconclusive.

The participants were asked to first select an indicator for improvement, and then list improvement ideas on post-it -notes. Thereafter the ideas were presented for the group and a new set of ideas were generated. The rule of the session was that no idea can be criticized, to allow the forthcoming of a larger amount of ideas. The method was applied on two goals and resulted in total of 34 ideas.

Although it was not planned by the author, the co-participating colleagues added spontaneously some extra low-level practices from the previous QGWS –phase, that caused a small amount of extra time spent on non-essential activities such as affinity mapping of practices on the wall and collection of a rough list of current practices on a flap board. However, these unrelated practices didn't cause significant increase in the temporal footprint or result in any bias or new results according to the author's observation.

The colleagues were able to extract on the flap board 8 current practices of two goals[12]. In addition at least 6 technical tools or other ambiguous titles such as the "database", "interfaces" and "experience of the coders" was mentioned on the flap board, but author didn't consider these as social quality practices, but rather part of the technostructure or the human capital. The participants commented the method to be beneficial and something that they would like to use again also in the future.

4.1.5 EESWS

The ESPA Experience Exchange Sharing Workshop (EESWS) was held in Jan'09. Only a small subset of all method users from the four companies participated the EESWS to discuss their perceptions about the QGWS, the QPWS, process improvement and other topics. The colleagues had composed an overview of the differences in the method between the companies, and moderated the discussion by providing topics for the discussion, partly based on the requests by the companies.

The participants of the Company C commented that the method had been useful, but also they had heard on the lobby people commenting it was also very heavy to use[13]. The coders didn't perceive the workshop as high-value enough to motivate them to participate any further workshops after the first one at the Company C, although it was agreed that participation of the coders would be an important supporting factor to the implementation phase of the development initiatives of the new practices. The participants agreed that the upper limit for the usage cost of the development method is roughly 2% (D: 20h/1000h, C: 1h/50h). The

[12] Company D QPWS pictures of the flap boards.

[13] Informants C1&C2 at EESWS

current footprint of the combined QGWS and QPWS workshops were estimated at 10% of the total project effort was considered too high by the opinion of the informant D1.

The Company D had used the results of the QGWS as an input for their yearly strategy workshop in December 2008, and commented that it was helpful to remember clearly the prioritization of the goals and to communicate them to the other units. The Company C had requested the QFF to be used on a specific scope of an iteration, which was not originally planned by the colleagues. The Company C had also applied the method without the help of the researchers, but commented it was more difficult to apply the method and shape the results compared to the previous workshops when the researchers were present (C3/EESWS).

The informant B2 added an insight to the communication practice by stating that one important factor to the success of the process improvement is that the representative of the personnel group has enough credibility amongst the peer group. Only the top-people rated by the peers should be involved. The informant D4 commented that in their company a working solution might be, if the personnel groups (such as the coders) would make first an internal meeting deciding about the common agenda before the cross-functional development meeting.

4.2 EBSE Semantic Database

After performing several practice selection method interventions, the author decided to perform a final constructive intervention by experimenting on the utility of an external tool compared to the internal knowledge of the subject companies. The author chose to copy the model of the Semantic Web -technology from the field of Evidence Based Medicine [Gao05] to the given problem, due to its potential ability in modeling complex ontological problems and the possibility to use artificial inference for data analysis. After the action research cycle and evaluating the EESWS -survey results (Table 8) it became also evident that none of the previously proposed social search methods were able to answer in any way the main research question, i.e. how the companies can **ensure** reaching of a certain quality goal. Only the numerical analysis of sampled evidence gives any hope for even trying to solve this industrially relevant question and to give insight on the comparative performance ratios of the practices.

Thus author constructed a Semantic Web [Barners-Lee01] –based RDF/OWL [W3C04] –data model for conceptual modeling of the evidence based software engineering –domain. The basic model has 5 classes (Practice, Result, Goal, Context and Reference) that are sufficient to represent an individual research result found from the literature, such as pair programming consumes in total 42% more man hours compared to solo programming [Nosek98]. The idea of the database is to offer possibility for companies and researchers to perform semantic queries by entering a prioritized and weighted goal vector and receiving a suggestion of quality practices that would the best match the defined goals. Based on the known empirical evidence of whether the goals can or cannot be reached, the knowledge database provides an answer, how close the goals can be met, what are the limitations, and how the current performance could be improved. The knowledge database can be used both by the companies to enter their own context specific data and by the researchers to enter and study the cross-contextual empirical evidence on how well a specific result is generalizable to nearby domains, such as from the students to the industry. The knowledge base is an instance of a semantic experience factory that collects, abstracts, and applies information about the best software engineering quality practices.

The author constructed an OWL -schema for modeling the problem domain (Appendix IV). While entering data to the model, the author noticed that the database produces also an ontology of the practices, quality goals and the context as a side product. The context is an important dimension to understand how generalizable the particular findings are across domains, such as the small web-site coding companies, the space industry and the students. The initial ontology version contains descriptions of about 100 software engineering practices and 30 quality goals. In contrast to the traditional wiki-based knowledge databases the most significant difference of the OWL-database is the possibility to perform automated inferring over the data to produce novel knowledge without need for explicit analysis by humans.

The data model accepts most easily well structured research results of the relationship between quality goals and quality practices such as the results provided in [Hätinen06]. However, the data gathered in the QPA-matrices is not well formed to be directly useful as an evidence for empirical software engineering, for several reasons. Firstly the goals have been split into 0-4 quality indicators, but the colleagues have performed the QFD -analysis over a super goal that categorizes the several quality indicators under one ISO9162 –style goal. This causes the effect of the QFD -practice weight vector to lose meaning since no direct effect can anymore be tracked between the QFD -vector and a specific quality indicator. The same effect is present with the QFD -vector in an even larger scale, since the contribution towards a super goal is calculated over a vector of 30-50 practices. Third, the increase in the error marginal is caused by the choice of the weighting method that produces both quantification noise (by using discrete values from 0 to 3) and blurring of the relative contributions of the practices towards the goals since the quantifier values are evenly spaced. Akao suggests using an exponentially increasing intervals (0,1,4,9) [Akao90], but this does not remove the error source. In a summary one can't but note that the QPA -matrices are ill-equipped to provide much contribution towards evidence on empirical software engineering. Perhaps the only significant contributions are the mapping of the used practices and the especially the so called 0-results, where the subjects have noted that a particular practice has no effect towards a goal. However in retrospect, the researchers have failed to consistently gather this information from all cases, although it would have been valuable from the Evidence Based –approach point of view.

To analyze the data available the author nevertheless constructed sub-ontology for modeling the QPA -matrices and inference rules to convert the data available in the matrices to a form that is comparable with the literature based empirical evidence. While the typical research articles typically report only one single result, the QPA -matrices are linear combinations of many Goals, weights and Practices. The weight that reports the relationship is an integer ranging from 0 to 3 (or alternatively undefined) in function of the set of 0 to 4 goals. It is obvious that the failure to represent the QPA -results clearly in functions of a single goal or practice greatly diminishes the reliability of the results, while the model clearly can't pinpoint directly any single relationship from the data. However, the data still contains some information, although of poor quality and relevance. The QPA-ontology assumes the phenomenon represented by the data to be linear in nature, and simply assigns equally distributed values between the goals and practices unless a specific weight is assigned. However, the current data is insufficient to calculate the relevance multiplier of the QPA-matrices, or how much the reliability of the results should be downgraded compared to the more high quality results reported in the literature. Despite these shortcomings, the QPA-information levels down the goal-practice relationship map indicating the hot spot relationships that are of interest for further research.

The initial version of the knowledge engine calculates two values: the evidence level that a certain practice reaches a specific goal based on the evidence, and the reliability of the result. The user may set constraints such as the desired goal level. The prototype system assumes that all goals have their metric optimum in the origin (i.e. zero bugs, zero clicks) according to the assumed quality definition.

The author used KAON2[15] inference framework to implement the inference rules using the Java API[16]. The development was done on Protégé[17] and Eclipse[18]. Author also included some third party ontologies such as

[15] http://kaon2.semanticweb.org/
[16] http://java.sun.com
[17] http://protege.stanford.edu
[18] http://www.eclipse.org

the SWEET/NASA unit conversion ontology[19]. Two SPARQL[20] query interfaces were built – a web based interface using PHP[21], ARC[22] and MySQL-based triple storage, and a command line version using the Java/KAON2 tools. The main KAON2 query for answering the question "*which practices should be used to reach a goal with an indicator value X (qgp:goalResultValue)*" is presented in Table 11. Due to the limitations of the KAON2 SPARQL support the result evidence has to be grouped, summed and sorted by the Java code, which is unnecessary with the PHP ARC library that supports SPARQL+ with aggregations and summing functions.

Table 11 - KAON2 SPARQL -query for providing empirical evidence on the quality practices

```
SELECT ?resultEvidence ?goalResultValue ?practice
WHERE {
       ?goalResult
                    qgp:isSubGoalOf     qgp:installability ;
                                        qgp:isGoalResult    ?result ;
                    rdf:type            qgp:GoalResult      .
       ?result
                    qgp:resultEvidence  ?resultEvidence ;
                    qgp:isResultPractice ?practice .
OPTIONAL {
       ?goalResult  qgp:goalResultValue ?goalResultValue .
}
FILTER        (
       ?goalResultValue                 < 30
) .
}
ORDER BY DESC(?resultEvidence)
```

The author codified the goals and practices in the QPA -matrices into a QPA-ontology (with a namespace prefix "qgp"). However, this was not as straight forward process as one would hope for. On the companies A and C the researchers had omitted entering the 0-values making the results arbitrary. On the Company B -matrix, the researchers had collected an aggregate of values from 4 separate subjects, but there was exact data available only for the top 3 goals. For the remaining 6 goals the author had to calculate averages by using the provided value ranges. A further hindrance in the codification was a mixed use of 0-3 and binary-positive (X) values for the Company B. However, the binary negative values were not marked during the data collection. The author solved the problem by quantifying the X –values as 3. Despite these shortcomings, the author feels that for the Company B omitting the aggregation increases the validity of the results and reduces the

[19] http://sweet.jpl.nasa.gov/1.1/

[20] http://www.w3.org/TR/rdf-sparql-query/

[21] http://www.php.net

[22] http://arc.semsol.org/

gatekeeper bias. The standard deviations for the top 3 goals for the Company B between the pre-assignment and the workshop gatekeeper aggregates were (σ = 0.79, 0.94 and 0.70) by average σ = 0.81. Another error source is the quantification noise, which contributes further

$$\sigma = \sqrt{VAR(x)} = \sqrt{\frac{1^2}{12}} = 0.28,$$

if the values are assumed to be evenly distributed[23]. It seems that the gatekeeper error is much larger than the quantification noise. Thus, it seems that the workshop -phase and the gatekeeper either has more correct information of the importance of each practice than the other employees, or the results filtered by the gatekeepers (both the colleagues and the manager C1) differ from the sample averages. In the opinion of the author the results should have been collected by using statistical and empirical methods instead of by political or subjective process. Third error source is the codification and classification of the goals and practices, which was to some extent inexact due to the imperfect knowledge transfer between the company and the researchers. The author compensated this by transcription the workshop recordings and extracting a large number of practices and ideas that were omitted by the colleagues during the sessions (see and). Further, the researchers had also failed to extract any data points on the current level of the quality goals from the Company B, meaning that no exact empirical evidence can be extracted from this company. The only extractable information is the claim that a practice contributes with a certain quantified weight towards a generic ISO9162 quality goal, which already in the definition might contain half a dozen subcategories, each of which might have dozens of possible metrics. When taking all the error sources into account, it seems that from the reliability of the QPA-method's results is low. Due to the vagueness of the classifications the method would need at least thousands of repeated measurements before it would produce any scientifically significant results for any question, making the application of the QPA in its current form a futile attempt for any single organization or even for the academia and industry combined. However, the contribution of the QPA -matrix is to give a rough overview on how the 30+ quality goals affect on the 100+ practices, which is not previously reported at least according to the author's knowledge.

On the Company C -matrix, the author noticed the difficulty in codifying the described practices into the ontology, since amongst the normal practices such as pair programming and short iterations, the author found practice descriptions "realistic scope and schedule" and "unrealistic scope and schedule". Both practices were given the QPA-weights, so it is impossible to know is one of the annotations describing the

[23] http://www.comlab.hut.fi/studies/1110/luennot/S-72.1110_Luento10.pdf

current situation and the other improvement, or something else. However the case, the researchers had failed to distinguish the current practices from the process development ideas.

Author constructed two inference rules for the QPA -matrix analysis add-on. The first inference rule calculates the evidence level of a single matrix data point as a Result, that can be read by the web query -front end. The resulting evidence value is calculated by dividing the weight (0-3) by the number of practices times the number of indicators in the column (goal) at hand. The rationale behind this formula is to ensure that sufficient precautious is used to prevent too optimistic conclusions of the available evidence in the matrix. As discussed earlier, the validity of the matrix results are diminished by the failure to pinpoint the relationship between the goals and practices exactly, the quantification error, the error caused by the choice of the quantification levels, and the failure to explicitly specify evaluated the quality indicator. Thus the formula cuts the maximum evidence by the number of other practices and indicators. However, the formula does not currently take into the account the further evidence reducing effects, such as what is the error range of the results compared to a rigorous study, where only one goal and practice is studied with a sufficient N? Although it would be important to study this property of the Goal-Practice -relationship, the author has scoped it out due to the insufficient N. A further application of the database would definitely require a more exact evaluation of the relationship by acquiring a sufficient dataset and applying statistical methods.

The second problem detected by the author was the sparseness of the goal ontology. The author failed to detect any overlapping individual quality indicators that would contain data on the current state of the metric. This is mostly due to the insufficient N of the study. Despite the fact that the author had scoped out the envisioned implementations of the Bayesian -network analysis of the goal and context ontologies from the study, the author decided to implement a transitive goal closure –inference rule that generalizes the results produced by the first rule to their parent instances (instances of the super classes). While the author also lacked the schedule to implement a functioning unit conversion, the inference rule currently omits the goal value for the parent instances. The result evidence value is diminished by dividing the parent result evidence geometrically by the level of ancestry. A much more scientifically valid (or the only sound) solution would be to use Bayesian overlay network, but the current practical solution at least ensures some reduction of the evidence level preventing drawing of too optimistic conclusions based on the data. After entering the available research data and the company goal vectors, the system gave the following suggestions for organizing the software engineering functions (see Table 12).

While analyzing the EBSE DB -results, the author noticed that the database suggested a number of practices that however, were already used by the companies, but were not evaluated on the quality function. For example for the Company A goal "testability" the database suggested with the highest evidence level practice "nightly builds", which was already used by the company. Yet another practice with a high evidence level, but unevaluated for the Company A, was the "coding convention". For the subsequent goals testability and accuracy the EBSE DB suggested practices that were already all used by Company A, but not evaluated as the highest ranking ones. Thus the suggestion might be to focus on strengthening the use of these practices, but in this work the author did not find support from the dataset to recommend any improvements for Company A. More data points should be inserted into the database. The second possibility is that no other companies provided evidence towards these goals, as it is unlikely that Company A has reached the optimal modus operandi.

These inconsistencies might be due to several reasons. First the QPA -evaluation was performed incompletely, since the researchers were unable to record a large number of practices during the actual workshops. The remaining data was extracted from the transcripts by the author. The second possibility is that the informants were unable to recognize the relationship of a practice and a goal during the workshop. Third possibility is that the colleagues failed to identify a correct quality indicator or the author coded the indicators to a wrong goal super class. Fourth question is whether it is possible to reliably generalize the effects of a practice – quality indicator relationship to an ISO9241 goal category reliably, or if the informants are able to perform the evaluation reliably. Nevertheless, a new verification study would be necessary for filling out the missing estimates and especially rechecking the contribution of those practices that seem to contribute a large portion of evidence in other companies.

For the Company D the data collection of the usage of the current practices was omitted intentionally by the author. However, the colleagues managed to collect some data on a flap board that was recorded using a digital camera. By reviewing the recordings, the author noticed that of the 11 practices recorded, the ones displayed in italics in the Table 12 are probably already in used by the Company D. Further, as already noted, due to lack of data, the database was unable to produce any information on which goal level can be reached by following these suggestions. If a sufficient volume of proper data could be entered, the database would be able to provide also this information and to offer a substantially larger level of practical utility.

Table 12While analyzing the EBSE DB -results, the author noticed that the database suggested a number of practices that however, were already used by the companies, but were not evaluated on the quality function. For example for the Company A goal “testability” the database suggested with the highest evidence level practice “nightly builds”, which was already used by the company. Yet another practice with a high evidence level, but unevaluated for the Company A, was the “coding convention”. For the subsequent goals testability and accuracy the EBSE DB suggested practices that were already all used by Company A, but not evaluated as the highest ranking ones. Thus the suggestion might be to focus on strengthening the use of these practices, but in this work the author did not find support from the dataset to recommend any improvements for Company A. More data points should be inserted into the database. The second possibility is that no other companies provided evidence towards these goals, as it is unlikely that Company A has reached the optimal modus operandi.

These inconsistencies might be due to several reasons. First the QPA -evaluation was performed incompletely, since the researchers were unable to record a large number of practices during the actual workshops. The remaining data was extracted from the transcripts by the author. The second possibility is that the informants were unable to recognize the relationship of a practice and a goal during the workshop. Third possibility is that the colleagues failed to identify a correct quality indicator or the author coded the indicators to a wrong goal super class. Fourth question is whether it is possible to reliably generalize the effects of a practice – quality indicator relationship to an ISO9241 goal category reliably, or if the informants are able to perform the evaluation reliably. Nevertheless, a new verification study would be necessary for filling out the missing estimates and especially rechecking the contribution of those practices that seem to contribute a large portion of evidence in other companies.

For the Company D the data collection of the usage of the current practices was omitted intentionally by the author. However, the colleagues managed to collect some data on a flap board that was recorded using a digital camera. By reviewing the recordings, the author noticed that of the 11 practices recorded, the ones displayed in italics in the Table 12 are probably already in used by the Company D. Further, as already noted, due to lack of data, the database was unable to produce any information on which goal level can be reached by following these suggestions. If a sufficient volume of proper data could be entered, the database would be able to provide also this information and to offer a substantially larger level of practical utility.

Table 12 - SPI Suggestions by the EBSE Semantic DB

Company	Goal	Suggested Practices	Evidence
Company A	Updateability	qgp:configurationManagementTools qgp:codeReviews	0.2541 0.1363
	Testability	n/a	
Company B	Familiarity	n/a	
	Usability	qgp:competitorBenchmark qgp:measureAndImproveDifficultFeatures qgp:functionalTestScenarios	0.3247 0.2727 0.2505
Company C	Functional Suitability	qgp:designMeeting qgp:changeManagementDocumentation	0.2308 0.2308
	Maturity	n/a	
	Usability	qgp:outsourcedFunctionalTesting qgp:functionalTesting qgp:userDocumentation	0.6071 0.5254 0.5124
Company D[24]	Reliability	*qgp:regressionTesting* *qgp:acceptanceTestingByProductOwner* *qgp:alphaBetaTesting* *qgp:manualTestingOfCustomerProcesses*	0.3595 0.3595 0.2730 0.2530
	Updateability	qgp:configurationValidationTools *qgp:codingConvention* *qgp:codeReviews*	0.2541 0.1363 0.1363
	Performance	*qgp:optimizeWhenProblemFoundInTesting* *qgp:functionalSpecificationAcceptance* *qgp:realisticTestingEnvironment* *qgp:manualTestingOfCustomerProcesses* *qgp:regressionTesting* *qgp:acceptanceTestingByProductOwner*	0.2222 0.1481 0.1481 0.1481 0.1481 0.1481

4.3 Summary of Workshop Results

Next the numerical findings of the QPWS were compared. The contains comparison of number of collected current quality practices by the companies and the workshop phases. The pre-assignment –column shows how many individual answers were given at each subject company. The pre-assignment was omitted for the Company C, and while initiated at Company D, no results were asked or returned. The WS –columns present how many practices were reported at the end of each workshop. The original idea was to held one half day workshop, but this was often forced to be extended to up to 4 workshops. Additionally, the author performed a post-analysis –phase by analyzing the transcripts of the workshops. The author divided the findings into current practices and SPI ideas (see Table 13). As a note should be mentioned that for example for the Company A, the number of findings doubled in the post-analysis yielding a total of 102 practices that

[24] For Company D the researchers collected no data on their current practices, thus the full overlap is not known.

were discussed. Only 38% of these were practices actually used at the Company A, the rest were new ideas. The total -column presents the final number of practices. For some companies the result is presented two-part separated by a slash. The prior part is the number of evaluated practices and the posterior number is the total number of practices entered into the EBSE DB based on the both QPA evaluation and the post-analysis. Respectfully the yield per man hour is presented in two parts. The man hours –column presents the total number of hours that the company personnel invested to participate the workshops (number of people times workshop duration), not including the pre-assignment.

Table 13 - Current Practices

Method	**Pre-ass.**	**WS1**	**WS2**	**WS3+**	**Post-anal.**	**Total**	**Man Hours**	**Yield/Mh**
Company A QPA	42	39	38	35	39	18/46	40	0.45/1.15
Company B QPA	19	20	23	31	40	25/40	12	2.08/3.33
Company C QPA	-	17	25	-	-	22/26	14	1.57/1.86
Company C IA	-	0	-	-	-	0	4	0
Company C QFF	-	0+2	0+14	-	-	0+14	8+?	0
Company D NMA	0	8	-	-	-	8	6	1.33[25]

A significant finding is the variance of the number of practices. For the Company A all pre-assignment results were included as input for the first workshop as the researchers were unable to group or filter them independently. The first workshop seemed to function rather as filtering and grouping session than as an idea generation workshop since the total number of practices (and also ideas) fell by three. The same pattern can be seen on the subsequent workshops; no new practices are recorded by the researchers, but were rather filtered out. However, the post-analysis by the author suddenly experienced a boom of new practices and ideas surfacing. It seems that the subjects constantly provided new information about the current practices and ideas, but the researcher-gatekeepers filling out the matrix were unable or unwilling to include them on the QPA-sheet. Nevertheless, the QPA yield of results per total hours spent was very poor. It seems that the QPA -workshop at the Company A yielded by average one current practice and one idea per invested man hour. However, almost a half of these findings were already collected and evaluated during the pre-assignment, making the worth of holding the heavy workshops dubious.

For the Company B a little bit different pattern can be seen. This time the number of practices seems to be increasing from workshop to the next. However, again, the post-analysis doubles the number of findings by adding 9 previously unconsidered current practices and 20 new SPI ideas. At the Company B the increase of

[25] Marked in parenthesis, since producing information on the current practices was not the intended.

the current practices can be partly contributed to the email exchange with the Company B usability expert B3, who provided a much more detailed account for the usability practices compared to what was provided during the workshops, yielding roughly 5 new practices and altering a few reported earlier. Again, roughly 50% of the yield of the process was gained before the workshops during the pre-assignments.

For the two QFF -workshops, the results are presented two-fold separated by a plus. The prior sign is the number of new unique practices or ideas found and the latter part is the number of overlapping refinements for existing practices or ideas. For the idea-matrix the QFF produced mainly refinements for test planning, or test cases how the new feature should be tested. Thus, it is uncertain whether the yield of the workshop was more as a test planning meeting or a SPI workshop. However, total of 6 new previously unreported practices were suggested. It remains unclear whether these practices such as "beta testing" have been used sometimes in the past or are they authentic new SPI -ideas. The second workshop seems to have yielded just one sheet to use 6 previously reported practices for a new feature. The author was unable to extract the duration and the number of participants of the second QFF -session, since it was held independently by the subject company without researcher participation. The both SPI idea generation and the new practice extraction yield of the QFF were both low.

The separation of findings to the current practices and the SPI ideas was performed during the post-analysis and thus the exact distribution of yield per workshop phase is unavailable. The total number of SPI -ideas per company and method is presented in the Table 14 The subjective quality (i.e. whether they were "good" or "bad") of the ideas is not evaluated in this work, as the approach of the work is objective and empirical. The QFF -yield is marked as a maximum due to the unavailable duration and number of participants for the second workshop. The workshop yields are presented separated by a plus. The prior part is the number of unique SPI -ideas and the posterior the number of further variations such as different test cases or ways to perform "a standard test". It seems that the method NMA yielded by a factor of decade more ideas per hour compared to the other methods.

Table 14 - SPI Ideas

Method	Pre-assignm.	WS1	WS2	WS3+	Post-anal.	Total	Ideas/Mh
Company A QPA	-	-	-	-	-	63	1.58
Company B QPA	-	-	-	-	-	21	1.75
Company C QPA	-	-	-	-	-	13	0.93
Company C IA	-	2	-	-	-	2	0.50
Company C QFF	-	6+2	6+2	-	-	6+2	<0.75
Company D NMA	0	34	-	-	0	34	11.3

The Table 15 contains the number of participants from the subject companies per workshop. Additionally 1-4 researchers participated the sessions. The pre-assignment of the Company B and the second QFF - workshop were performed independently by the subject companies without participation of the researchers.

Table 15 - Participants per Phase

Method	Pre-assignment	WS1	WS2	WS3+
Company A QPA	4	6	2	2
Company B QPA	(4)	2	2	2
Company C QPA	3	5	4	-
Company C IA	-	4	-	-
Company C QFF	-	4	N/A	-
Company D NMA	0	3	-	-

5. Analysis

Next the results of the workshops and the database are discussed to analyze the evidence towards answering the research questions at hand. First, the Quality Practice Workshops are analyzed, followed by the prioritization method, current practices and the discussion on the innovativeness of the methods. The preceding phase of the quality goal workshop is excluded from this work.

5.1 QPWS

After the several subsequent workshops at the companies A and B author noticed that it seemed the companies already do have a plenty of practices towards each unmet goal. Thus one of the starting assumptions of the QPA seemed to prove quickly false; it was not possible to point any "holes" in the Quality Palette for process improvement, since all goals had already from 3-5 most highly ranked (contribution=3) practices and even more lower ranking practices. The further steps to formally analyze the data as specified in the original QPA –article was silently discarded as a futile attempt, although author had interest to complete the analysis. The overview of answers for the iterative evaluation problems is presented in Table 16.

The colleagues claimed the research space to be empty, while they could not find any relevant articles related to SPI and matching of goals to practices. In contradiction, the author had initially a divergent view by being able to list couple dozen of different management methods, such as Management by Objectives, Theory of Constraints, Total Quality Management and Six Sigma, starting from the 50's overlapping the same research area. However, these methods were discarded by the colleagues by questioning their applicability to the context of software engineering, although there is a number of evidence and text books available describing the usage of such methods in the context of SW engineering [Biehl04, Zultner93 etc.]. Compared to the two closest related methods the QFD and TMap, QPA differs mainly by trying to construct the deliverables subjectively during a workshop instead by the analytically by the managers. This might be a reflection of the organizational culture of the research group that was in charge of designing the original method. To comply with this hidden assumption, the author suggested a non-analytical, high-participation brainstorming session as a top candidate for a possible future practice selection method (NMA). This method is compatible on the execution level with the QGWS –method, since it shares the same basic low level execution practices of two-phased brainstorming and post-it -notes. However, as seen from Table 8, the assumption that the participation is of the utmost importance is a false one. Thus, the process should have been originally designed to focus first on providing effective quality improvement instead. For this purpose, the proposed knowledge engine seems to be the best solution, as it can at least theoretically provide empirical evidence surpassing the capability of a single manager or a team in knowing the efficiencies of the different practices. However, the findings need to be confirmed by using a larger sample.

Table 16 - Iteative Evaluation Problem Answers

Evaluation Problem	Answer	Rationale
1. Should the "Indicator Analysis" be used for practice selection?	No	It can't be performed ad hoc on a workshop, but it might work as a separate analysis task.
2. Should the "New Method A" be used for practice selection?	Yes	It produced a lot of process improvement ideas.
3. Does the absence of a key employee role affect results?	No	The R&D Manager was able to provide overview of the used practices with significant completeness.The absence of the target employee representation did not increase the amount of reported methods or new ideas significantly (project manager involvement produced 5 new practices or ideas out of 117).
4. Should the QPA be used for practice selection?	No	It is too work intensive compared to the outcome.
5. How the goals should be documented to support QPWS?	No	Show only the description, current value and target value, and no other data.
6. How the practices should be categorized for SPI?	No	The different categorizations provided no benefit for the SPI process, except for the researchers to conceptually group the results. Instead, the grouping caused a process loss in filtering out some ideas and current practices.

By comparing the QGP-matrices produced at the QPA -sessions and by reviewing the field-notes by the author, it seems that the colleagues acted as gatekeepers for the data collection and had a number of biases. Firstly, author was able to extract from the field notes almost twice the amount of SPI ideas and current practices compared to what was recorded on the matrices during the sessions (&). It seems that the colleagues had difficulty in formulating the practices and ideas based on the discussion, since they didn't have any familiarity with the daily work practices of the company and some of the stated practices were ones not commonly referred in the literature. In the future this bias might be mitigated by having a participant from the company acting as a secretary filling out the matrices and lists. Second, the gatekeepers seemed to systematically omit listing the "bad practices", or practices that have negative impact on the quality goal. It seems that both the colleagues and the subjects had the same bias. For example the colleagues failed to record on the IA –session the negative impact on the usability caused by the practice of introducing new modular features. Similarly the participants denied the existence of such a negative effect when explicitly asked, despite they had just a few minutes earlier claimed a contradicting statement. Third bias by the colleagues was to scope out so called "process practices" and "non-quality practices", but failing to explicitly define the selection criteria. For example at the Company A the colleagues omitted the legal practice of "project contracts" despite its reported effect on one of the top3 quality goals. Also at the Company C omitted the "release scheduling" as effecting to the process quality, despite the participants explicitly stated it as one of the main causes for the quality problems (i.e. a bad practice) referring to it as "*a farce, a comedy drama*". The author questioned this arbitrary criterion later by comparing it to "unit testing". It seems that unit testing is clearly a process practice, since it produces no visible deliverables to the customer and should be omitted. However, this practice was always included by the colleagues. It seems that the gatekeeper limits to a significantly large extent the search space of the opportunities to a single perspective, which seems to be theoretically and practically suboptimal. Due to the multitude of biases, it seems that using a gatekeeper should be avoided when possible.

Based on the data (), it seems that the temporal footprint and efficiency of the methods varied wildly. The NMA at the Company D seemed to be the most efficient idea generator and the least time consuming. However, since the phase to record the current practices was omitted, the drawback was that the colleagues were unable to get comparative data what practices are actually used at the Company D. The author had deducted this to be unnecessary by assuming the company already knows its own practices. In retrospect this information could have been useful for the EBSE DB, but this was not known during the workshops. Explicit extraction is only necessary for the external parties such as the researchers. However, the informant D4 commented at EESWS, that it anyway could be useful to map the current practices explicitly, since it is likely that the people at different departments can have a very different view what kind of practices are used at the other department. This problem was stated to be smaller at the Company B with 50 employees, since the small size ensured that the departmentalization problem had not yet emerged as strongly as at Company D with 450 employees. It should be noted that none of the suggested methods take account the situation, if the organization should be different from one function to another, but a monolithic organization was assumed. Only the QFF –method takes a step into this direction, but fails to provide a clear overview by whom, when and how the different practices are to be selected per feature, or if the new suggestions are cumulatively added to an overall process. The quantity is also alone a bad indicator, since the codification revealed the quality of the ideas to be poor and difficult to define exactly based only on the ambiguous post-it -notes. Anyhow, the author deducted that the workshop was one of the main reasons for the temporal footprint overflow witnessed at the companies A, B and C while applying the QPA –method, and omitting it would be a plausible solution. The NMA seems to solve the gatekeeper biases by recording all SPI ideas individually on the post-it -notes. The second alternative would be to develop a pre-assignment tool for the knowledge database to collect results from the individual subjects. Despite the colleagues introduced affinity mapping as a possible procedural gatekeeper process, the author was able to circumvent this potential bias by discarding the grouping at post-analysis and listing all SPI ideas on the SPI backlog individually.

The research group had stated an assumption that the temporal footprint will decrease when the method is installed in the company and used in several consequent releases. However, based on the length of the QFF – session of two hours that used the previous product level data as input, this assumption has no practical evidence to support it, and the opposite might be more true. The QPA and QFF –methods seem to experience rapidly diminishing returns to the invested effort.

5.2 Prioritization

To discuss the sub-evaluation problem "*how the goals should be documented to support QPWS*" further, Akao points out that the prioritization is the most important activity in QFD [Akao90, p.10]. The research on this topic remains inconclusive, but the author criticizes strongly the usage of voting as a tool for prioritization in a hierarchical company. The colleagues also lack literature support to rationalize the choice for the usage of voting, while the Quality Attribute Workshop (QAW) [Barcacci03], from which the voting was copied, does not rationalize its usage. Due to the lacking information on the relationships and levels between the goals, the voters had difficulty to discern the true priority of the goals and the result is likely to contain a substantial bias due to misinterpretation of the meaning of the voting targets by the voters.

Using voting as prioritization method remains questionable also when applied to QFD, since as described in the original source, only pure analytical methods such as AHP the Pareto-chart should be used [Akao90]. As discussed earlier, the author can't see any other outcome, but the top management to overrule the results of the voting, when real process improvement is started, which will lead only dissatisfaction of the employees when a charade of voting is performed in corporate context. The first requirement step of change management by [Folan05] "support from the top management" agrees with this conclusion; no change initiative can be successful without the consent of the top management regardless of how popular it might be amongst the employees.

By negating the hierarchy-assumption, the result of the vote can be also interpreted as the ability of the management to communicate the strategic intent to the employees. At the Company B it seems that the employees did not uniformly comprehend the prevailing strategy, which gives further motivation for utilization of the QGWS method for communication of the strategy and prioritization in larger extent, if the voting is replaced with a more enlightened decision making practice.

However, the data to formulate a conclusion to this research question remains insufficient. The only lesson that can be drawn from the data is that voting is not a rational method to prioritize the goals, as it is unsuitable for hierarchical command-structures such as companies. The only practical finding is that a contradiction between the voting result (if such a method is for some reason used) and the strategic intent indicates insufficient amount and skill of communication by the management.

The open question remains how to handle the interrelations between the goals in prioritization. Based on the industry experience and the literature study [Andersin01, Ittner03, Folan05] the author suggests formulation of a causal quality model such as the Strategy Maps [Kaplan04] to solve this exact problem. For successful strategy implementation it is an imperative to derive the Critical Success Factors (CSF) from the organizational strategy, and to further transform them into performance scorecards. Based on the literature review [Ittner03] it seems likely, that by omitting the CSFs the company will focus its efforts on non-performance driving activities causing sub-optimal performance compared to its rivals. The applicability of this method in context of the QPWS was not studied in the scope of this thesis, although it is preferred by the author.

5.3 Current Practices

According to the data on it seems the QPA method was slightly more efficient (i.e. higher output/h) than the others to extract information on the current practices of the companies. However, the difference to the methods such as the NMA where the extraction was not intentional is not very large. Thus it raises disbelief whether QPA is really as efficient as it should be on the task of current practice extraction. The Indicator Analysis can be seen clearly as the most inferior method, since it failed to record any current practices, although it was the objective of the method.

It seems that the method QFF was unable to extract much new information of the current practices. This might be due to the fact that the intention of the method was to use the previous data as input for practice selection per new feature and release. During the follow-up, the Company C confirmed that they had actually implemented at least partly the QFF -plan. For example the practice "screen capture videos" was implemented and the practice "specification in co-operation with the customer" was enhanced.

An alarming finding was the dropping of the analysis phase of the QPA –method . While the colleagues did not find any way to analyze the data, the whole workshop phase remains unjustified. If no analysis can be performed on the data, the author recommends removing the practices collection phase from the method. However, as a compensatory construction the new EBSE DB seems to provide an analysis step of the relationship between the goals and practices and is also able to provide SPI recommendations.

Despite the decades of QFD software industry application, it remains unknown, why the well described full range of practices of the QFD were not found previously and used as part of the QPA –method, or why a new so closely related method was constructed without references to the original. It also remains unclear, why this method was considered better or chosen over any other more contemporary methods. The research group's professor Lassenius commented later on QFD that it is known to result into an ever-increasing number of matrices, which makes it very cumbersome to use, and it's practical popularity has diminished since the early 90's[26]. The method's author Akao agrees and suggests for example that using QFD for more than five goal levels leads into explosion of detail. In the construction engineering the target costing planning -method is performed typically for three iterations before execution of the plan [Haahtela07].

However, the QPA method offers a novel contribution to the QFD -method by constructing a quality goal and software engineering practice -matrix for the first time. Previously the process/quality goal -matrix has been used for process analysis [Hauser88], but not with a practice break-down. The lack of proper literature study by the colleagues is shown in the QPA method by re-inventing the wheel in several occasions and simultaneously failing to incorporate or evaluate the usefulness of the other well-known QFD -practices. For example the colleagues fail to recognize the top-down work-breakdown paradigm of the QFD -approach inherit also in the QPA -matrix resulting in difficulties, when applied on the leaf before performing the higher level analysis [Vanhanen09].

[26] Thesis Steering Meeting 20.1.2009, Casper Lassenius, Antti Hätinen, Mika Mäntylä, Jari Vanhanen.

The applicability of the QFD and QPA to the agile software engineering remains questionable due to the increase of the agile manifesto preferring working code over documentation [Beck01], although Akao recommends to use QFD in an iterative fashion. Akao suggests that several revisions to the quality chart are required and it is unlikely especially in the new product development that the first quality chart would produce even a satisfactory result [Akao90, p.10]. This is a similar approach to the traditional Target Costing –method in the construction engineering [Haahtela07], where the planning detail is iteratively elaborated by using detailed historical cost statistics breakdown of typical building projects. The major difference of the construction projects to the software is that the typical project cost variation is low (only 3-5% of the project sum is allocated to the risk reserves), while the budget of a software engineering project can overrun by several hundreds of percents. Thus this heavy planning oriented design method is unlikely to gain more popularity in the low documentation oriented agile organizations despite its systematic approach. While several studies indicate QFD's application for large scale non-iterative processes [e.g. Martin98], the application data of this research suggests even the simplified QFD/QPA -matrix to be too cumbersome for usage in relatively slow cycle (6-12 month) iterations. The administrative overhead would grow even higher when a full-scale QFD would be applied by using the method also for the methods original purpose of functional design. It is doubtful that even Akao's suggestion of constructing the QFD -matrices only for the product categories would ease the burden [Akao90]. He also suggests that

"an incomplete quality chart can do more harm than good",

questioning the validity of partial QFD -charts such as the QPA. This partly also demonstrates Akao's incomplete understanding of the method's roots in Target Costing and its main purpose of maximizing the profit for the contractor. However, he also suggests the companies to adapt the QFD -method to their own needs and recognizes that the standard QFD -chart is not suitable for all situations. While the QFD -method optimizes the non-cost related elements of the project, the Target Costing -method should be applied first to ensure reaching of the acceptable economical rationales.

If the organization is lacking an experience factory or other facility for performance analysis and best practice benchmarking, the collection of the metrics is unjustified by from both the practioner and researcher point of view. The social search methods were designed by the assumption that the participants possess the best knowledge for the most effective direction of SPI. However, the picture changed fundamentally by the introduction of the EBSE knowledge database that assumes holding the best available empirical evidence on software engineering processes. The EBSE paradigm regards the evidence provided by subjective opinions inferior to the evidence provided by actual measurement of well defined metrics and extraction of the current practice vector. However, the data collected in this work is insufficient in sample size to determine which paradigm is superior and this should be investigated further.

The EBSE DB flips the interpretation of the results upside down. The NMA -results are rendered to useless speculation, and the best (although weak) evidence of the compared method is produced by the QPA – method. The whole process of current practice extraction that was regarded useless for the social search – based approach is now the only rightful data collection method. However, a substantially higher rigor should be used to increase the reliability of the measurement and to lower the required sample size to produce more scientifically significant results. As a summary, in the absence of an operational EBSE DB, the NMA seems to be the best method of the compared ones for SPI. When a functional EBSE DB can be developed, the primary emphasis will remain in entering first all available secondary research data into the system. Second phase would involve development of a primary data collection tools that would introduce a substantially more precise and statistically valid method than what the current QPA facilitates.

5.4 Innovativeness

From it becomes very evident, that the NMA with the rate of 11.3 ideas per man hour is by far the most efficient method to produce new process improvement ideas. It seems that by removing the gatekeepers from filtering and grouping the ideas, the individually submitted post-it notes seems to facilitate the best participation to the idea production. The individual post-it -notes seem to also enlarge the number of perspectives involved in the search process of the possible SPI -idea space better than the methods utilizing a gatekeeper. The concrete indicators from the QGWS also clearly accelerated the idea generation. The two-phased brainstorming of having the participants to represent their ideas at the middle of the session produced some ideas, but nevertheless significantly smaller amount than on the first part. Thus, it might be possible to optimize the method by removing the second phase, keeping the NMA –sessions regularly, and by replacing the time used for the second phase by choosing a new goal indicator to be brainstormed.

The innovativeness of the QPA -method seems to produce roughly the same amount of SPI ideas, but the temporal footprint is up to 10 times larger and thus the QPA –method is more inefficient than the NMA. The IA is again inferior to the both methods by being able to produce only 0.5 ideas per man hour.

The QFF -method seems to provide a few new SPI -ideas, but the yield compared to the invested man hours seems to be poor. However, it also seems that the per-feature -analysis enables the participants to discuss the details of how each practice should be applied in detail. The yield seems to contradict the hypothesis by the colleagues that the efficiency of the QPA -method would increase on the subsequent workshops. The opposite seems to be more true; the QPA –method seems to experience diminishing returns in function of the effort invested.

The Company A used the QPA -method independently in Feb'09 without help of the researchers. They chose existing practices to be improved, such as focusing on increasing the automation percentage of the functional test suite. The SPI backlog was not used and no new ideas presented earlier were chosen to be improved. This can be interpreted to have been caused either by an implicit QPA -matrix analysis performed by the company trying to fill out the largest gaps in their practice palette, or the prevailing perception by the company that the increasing the automation degree would anyway be the most rewarding course of action. The prioritization criteria used by Company A should be further investigated for its general applicability versus the methods available on the literature.

5.5 Analysis Summary

To discuss which SPI method should be used by comparing the results provided by the different SPI methods, it seems that the QPA is too ineffective given the time constraint from being used in the industrial context. The QPA used roughly 10% of the subject project man hours, provided a low number of output ideas and practices per hour invested, while a maximum budget of 2% would have been acceptable. The Indicator Analysis failed to produce almost any results and it cannot be recommended. The NMA seems to provide clearly the highest number of SPI ideas per man hour, and is recommended.

For the sub-question how to document the quality goals to support the practices workshop it was found that only the description of the goal (i.e. no ISO9126 topic), current value and target value should be presented for the subjects. All other fields, such as the attempt to categorize the goals, caused confusion amongst the subjects. The author argues that in the industrial context the goals should not be voted upon, since the companies are hierarchical command structures and not democracies, unlike the research organization that has developed the method at hand. An alternative method for prioritization of goals is the hierarchical Target Costing –method found on QFD that starts from a rough overall target and iteratively creates a more detailed goal hierarchy. However, according to Akao the drawback of this is the exponential growth in work required to manage the increasing detail.

6. Validation

Next the reliability and correctness of the presented results are evaluated. This is performed by discussing the reliability, the SPI suggestions and the EBSE DB model.

6.1 Reliability

After the implementation of the EBSE database, the author analyzed what kind of information the collected data contains. The first observation was the incompleteness of the data; the Company B goal matrix did not hold any information about the current state of the goals failing to produce any evidence that a particular combination of practices would produce a certain goal level. This was due to the fact that the current state of the goals (i.e. usability) were either not presently known by the subject or were related to a new product not yet available for evaluation. The author tried to imagine what kind of worth the collected data provides, but was unable to come up with any solution, where the value of the incomplete data matches the effort that was invested in the collection (see). Thus the first suggestion by the author is to ensure sufficient rigor of data collection is emphasized to improve the quality in the future.

The second look on the analysis results provided the overview of the combined practices and goals. Evidently the conclusion was that the data is very sparse. The author was able to point only one goal and practice, where data is collected from two companies on a single goal; namely the user observation for learnability from the companies B and C. The result data, however, revealed also a new source of error, the classification of the input data. The combined data showed the highest evidence for learnability to be contributed by the practices “training feedback” (e=0.6000)[27] and “user observation” (e=0.5579). When the company goal matrices were evaluated against the results, this was the single data point suggesting (very vague) evidence for improvement of the practice set of the Company B by introducing the new practice “training feedback”. However, while validating the correctness of the calculation by the formula specification, the author noted that the original definitions for “training feedback” at the Company C was very similar to the definition of “user observation”, and not to the traditional definition available in the literature. The Company C was claiming to perform “user observation”, while in reality they were rather doing “training feedback” by observing users during the training sessions. The closeness of the mutual high evidence level compared to the other practices with similar N’s is some (rather vague) indicator that these two practices definitions probably

[27] As mentioned before e=w/Sum(w)/Count(G) in this case for Company C’s goal number III e= 3 / (3+2) / 1 = 0.6

would have high correlation. when a Bayesian classifier would be implemented, applied and sufficient amount of data would be provided.

Another explanation for this inconsistency could be a finding of a new previously unreported emergent industry practice "user observation during training". The companies often fail to apply the practices exactly as described in the original work, but adapt them to fit better their competitive environments, resources, skills and context. The unorthodox pattern of combining "user observation" [Hackos98] and "training" can be considered as blasphemy by the academia, but seem to yield better management acceptance with both commercial and usability related results compared to the original sources. Another emergent practice could be the Company D's "demo by customer", where the product owner instead of the coders presents the new functionality to the others. This new practice increases the awareness of the coders towards the quality, while they can't compensate the known bugs by avoiding parts of the software that is less stable during the acceptance demo session.

The second revelation of the analysis was that the input data classification done by the author was a significant error source. In future the classifications of the ontology should be at least analyzed by advanced algorithms such as self-organizing maps [Kohonen82] and Bayesian networks to find clusters of similar and divergent practice definitions. This would provide improvements for the conceptual ontology by giving more disjoint definitions of the software engineering practices, provide validation for correctness of such an ontology, and potentially discovering new previously unknown classifications that would advance software engineering as a science. An equally important, but currently actively dismissed research, should be conducted on the so called bad practices or anti-practices to find the common patterns that hinder the quality.

Third finding was on the second inference rule allowing the transitive generalization of the results from metrics to their parent goals all the way up to the root goal of the ontology; the "qgp:quality". As discussed earlier, while the validity of this transitive closure can be considered dubious, the inference produced more overlapping results similar to those that were presented by the colleagues to the companies on the EESWS[28]. The colleagues had made a different classification for the workshop material for example by combining "user observation" and "usability testing" in their ontology, while being separate practices on the author's EBSE DB. Also, on the original document only the occurrence of a practice in a company for a goal was noted, but the EBSE database calculates and sorts the practices also by evidence weights. For example for the parent

[28] Vanhanen J., EESWS 28.1.2009, Jari_Quality Goals Found in Companies.pdf

goal “usability” the EESWS material showed all companies using practice “functional testing”. The EBSE DB ranks this practice as third (e=0.5254) after “user observation” (e=0.6377) and “outsourced functional testing” (e=0.6071). Other high ranking practices include “user documentation” and “usability testing”, before the evidence level of the remaining practices drops substantially. Thus, if one would trust his decisions to the vague evidence provided by the database, all subject companies could at least consider including the two higher ranked practices in to their practice portfolios.

The second most highly ranked goals on the EESWS material are the installability and updateability, stating “smoke testing” as the practice that should be used in all companies. However, in contrast the EBSE DB calculates “smoke testing” to provide a very low level of evidence (e=0.0967) that this practice would be useful for reaching the goals of the companies. This can be explained by two factors. First, though semantically defined, the system does not have rule for transitive practice closure describing, which practice specializes on some other practice. The evidence matrices for updateability have two smoke testing practices, the vanilla “smoke testing” and the special case “smoke testing in a realistic environment”. The two practices are separated through reducing the individual evidence level. The second factor is the high number of practices (n=13) evaluated by Company A to contribute towards three indicators. Compared to the usability results where only a few practices were evaluated, the overall evidence level is substantially lower. However, in principle the evidence should be more trustworthy since the data is more detailed. On the other hand the high number of practices and the error caused by the choice of the quantification levels blurs the meaning of the weights close to a plain binary (has effect / doesn’t have effect) stratum. It seems that also those cases, where a smaller number of practices and goals have been evaluated (perhaps due to lower rigor) the formula produces comparatively higher evidence level, which should not be the case. Currently the EBSE DB suggests that the practice “configuration validation tools” has the highest evidence contributing towards the generic updateability goal. However, the data is inconclusive to point whether the contribution is negative or positive towards reaching a certain goal level.

Yet one more significant result provided by the database is which practices should not be used. However, the author noticed that the initial solution of annotating the unknown values as e=0.001 in comparison to the “no effect” -value of 0, results in round down of all values to 0.000, making the differentiation impossible. The “no effect” values would otherwise be very useful for the detection of practices or anti-patterns in the company methodology that should be removed as counter-productive. The practices with non-zero evidence are listed on Appendix II. This remains as a concern for further study how to process this data on an open world assumption –based system[29], and the current database is unable to provide unfortunately any information on which practices should not be used.

6.2 Evaluation of SPI Suggestions

All of the examined constructions produced a number of SPI suggestions. Next the results of the methods are qualitatively cross-tabulated to perform a comparative analysis. The quantitative comparison can be found earlier in the .

The author extracted the top SPI suggestions produced by each method in Table 17. The QPA -method for the Company A produced a vast number of ideas, of which only the top ranked ones are presented. For the Company B, the idea -backlog seemed to be poorly constructed, as it contained only the same two generic practices for both of the top goals. The Company C QPA SPI -matrix was also a rather vague one, and contained the same practices both in the current and the idea backlogs. The related goal to which the SPI idea belongs to was not explicitly marked unlike on the other companies. The IA -backlog was in contrast well-formulated, but contained low number of ideas as the product of the low innovativeness of the method. The QFF –method seemed to have produced a few more unique new ideas than IA and QPA. The NMA-method used at the Company D produced a vast quantity of ideas, but the author noticed difficulties in codifying the suggestions to ontologically valid practices. The idea backlog also lacked estimations of the importance of the idea. The author was forced to filter roughly 2/3 of the ideas due to their mutual affinity and also by the vagueness of the description not allowing direct recognition as a previously codified practice. This might be due to lack of contextual understanding by the author. In another study the yield of accepted brainstorming ideas for product development was measured to be 80% [Tyllinen09], so the authors codification might have been substantially biased. For example the author was unable to codify the idea "improvement of the architecture" as a practice, despite this SPI might have useful and specific semantics for the subjects.

[29] Compared to a closed world assumption (the relational DB), on the OWA logic failure to derive a fact does not imply the negation.

Table 17 - SPI Suggestions by Company and Method

Company	Goal	SPI Suggestions	Method
Company A	Updateability	*qgp:installerTools* *qgp:training* *qgp:automatedRegressionTests* qgp:configurationManagementTools qgp:codeReviews	QPA
	Testability	*qgp:automatedRegressionTests* *qgp:realisticTestingEnvironment* *qgp:automatedTestBuilds* *qgp:codeReviewsForCriticalFeatures* *qgp:tutoring* *qgp:performanceTesting* *qgp:automatedTestingOfCustomerProcesses* *qgp:codeCoverageMetrics* *qgp:staticCodeMetrics* *qgp:newTestWhenBugIsFound*	QPA
Company B	Familiarity	qgp:userDocumentationProcess qgp:testingProcess	QPA
	Usability	qgp:userDocumentationProcess qgp:testingProcess	QPA
Company C[30]	Accuracy	*qgp:shortIterations*	QPA
	Maturity	*qgp:shortIterations* *qgp:functionalTesting* *qgp:realisticScopeAndSchedule* *qgp:automatedUnitTests*	QPA
	Usability	qgp:usabilityTesting qgp:realisticTestEnvironment qgp:dedicatedTestingTeam qgp:screenCaptureVideo qgp:alphaBetaTesting	IA IA QFF QFF QFF
Company D	Reliability	qgp:regressionTestingForCriticalFeatures qgp:rootCauseAnalysis qgp:realisticTestEnvironment qgp:testCoverageAnalysis qgp:earlyCodeReview qgp:manualTestingOfCustomerProcesses	NMA
	Updateability	n/a	NMA
	Performance	qgp:architectureDocumentation qgp:performanceRequirementSpecification qgp:realisticTestEnvironment	NMA

[30] Practices in *italics* are already used by the companies, and were identified as improvement suggestions.

The author experimented also on a few SPI idea generation methods. The author envisioned the implementation of the I (idea) -matrix data analysis as an inference rule, but due to time constraints and the even lower potential (compared to the current C -practice matrix) evidence relevance, decided to drop implementation of the sub-analysis plugin. The potential of the I-matrix analysis would have been to utilize also Company D and the I-matrix data from the other companies, but the author acknowledges that while the C -matrix is based on the opinions of the subjects, the I-matrix contains only subjective projections based on the cognitive models of the respondents. While the empirical software engineering research should be based on objective observations on real phenomena and considering the already low validity of the C -matrix, drawing conclusions from the I-matrix would have been pure subjective speculation. Compared to entering real empirical research paper results the author sees little value on modeling the hopes and dreams of the subjects, while they are unlikely to be based on empirical research evidence [Shaw07].

Next the author compared the SPI suggestions provided by each method and the EBSE database (Table 12). The Company A suggestions provided by the participants seemed to be very similar to the current practices found at the Company D for the goal “updateability”. The EBSE database suggests in addition the usage of “configuration management tools” and “code review” for increasing the updateability. The Company A has identified the first EBSE DB suggestion for implementation, but has not ranked it as high as the other suggestions found during the QPA -session.

While analyzing the EBSE DB -results, the author noticed that the database suggested a number of practices that, however, were already used by the companies, but were not evaluated in the quality function. For example for the Company A goal “testability” the database suggested with the highest evidence level practice “nightly builds”, which was already used by the company. Yet another practice with high evidence level, but unevaluated for the Company A, was the “coding convention”. For the subsequent goals testability and accuracy the EBSE DB suggested practices that were already all used by the Company A, but not evaluated as the highest ranking ones. Thus the suggestion might be to focus on strengthening the use of these practices, but in this work the author chose to not recommend any changes. The second possibility is that no other companies provided evidence towards these goals and more data should be collected.

These inconsistencies might be due to several reasons. First the QPA -evaluation was performed incompletely, since a number of practices were unfortunately not recorded during the workshops. The remaining data was extracted from the transcripts by the author. The second possibility is that the informants were unable to recognize the relationship of a practice and a goal during the workshop. Third possibility is that the colleagues failed to identify a correct quality indicator or the author coded the indicators to a wrong goal super class. Fourth question is, whether it is possible to reliably generalize the effects of a practice – quality indicator relationship to an ISO9241 -goal category. Nevertheless, a new verification study would be necessary for filling out the missing estimates and especially rechecking the contribution of those practices that seem to contribute a large portion of evidence in other companies.

The EBSE DB fails to provide any new suggestions for the goal "testability". Thus, the QPA method clearly outperforms the cross-tabulation method by providing actually the largest number of high quality suggestions of all the applied methods. Only the NMA at the Company D produced more suggestions, but their quality was a rather poor at least from the practice codification point of view.

For the Company B the QPA resulted only a poor number of SPI -ideas. Thus, in this case it seems that the suggestions provided by the EBSE DB should be used instead. One possible explanation for the poor QPA performance for SPI suggestion generation at the Company B seems to be that the majority of the ideas were not recorded during the workshop, but were extracted later by the author from the transcripts. Thus the ideas remained unevaluated making it impossible to identify to which goal the ideas belong to and how efficient the idea could be to improve quality. Thus these ideas were excluded from the analysis, but if a re-evaluation would be performed on the idea contributions towards the goals, the backlog contents could be altered to favor the QPA -workshop results over the EBSE DB.

For the Company C the QPA also failed to produce a proper SPI matrix. The suggestions recorded were most often duplicates of practices already used by the company, but identified for further improvement. However, the Indicator Analysis produced the first ingenious SPI idea the "usabilityTesting". The number of ideas produced by the IA was unfortunately almost insignificant. The QFF -method was able to generate a few more new unique SPI ideas, when the assessment scope was narrowed down to a release level. However, the number of generated ideas was not impressive. The EBSE DB provided one matching result for the "usabilityTesting" agreeing with the IA. Two suggestions namely the "dedicatedTestingTeam" and the "alphaBetaTesting" were codified by the EBSE DB, but in the context of other quality goals, and provided thus no information on their effect on usability. Based on the combined data this relationship should be probably studied in more detail.

For the Company D and goal "reliability" the suggestions provided by the NMA and the EBSE DB probably matched the best. While lacking the full sample of the current practices, the NMA idea and the EBSE DB suggestions seem to overlap to a higher extent than for the other companies and methods. For example both methods suggest using the "regression testing" for the goal "reliability". The NMA provided a number of more detailed ideas how the application of this practice could be improved. The NMA produced also the idea "testing of customer processes" that the author matched with the EBSE DB suggestion "manual testing of customer processes". While NMA produced also an idea "improve test automation" this could be also be rather matched with the "automatic testing of customer processes", but the EBSE DB currently lacks codification of such a practice and the researchers should re-check ,which practice was exactly mentioned by the original NMA- idea. The NMA -analysis was not performed for the goal "updateability". For the goal "performance" an overlap exists again. The practice "realisticTestingEnvironment" is produced by both analyses. Additionally the practice "performanceRequirementSpecification" specializes on the "functionalSpecificationAcceptance". While the EBSE DB lacks any current practice or goal data from the Company D, the result can be interpreted as a promise that the new system holds some estimation power for forecasting the future organizational performance.

Of the four analyzed practice selection methods and the EBSE database, it seems that the NMA and EBSE DB provide the best mutual match. The other methods (QPA, IA, QFF) seem to provide poor support for generating SPI suggestions either due to internal process failure or the low support for innovativeness, except for the Company A's second goal. However, the evidence in all cases is rather weak and cannot be interpreted to provide any scientifically significant meaning due to the too small sample set.

6.3 Model Validation

The author faced a multitude of difficulties in finding ways to validate the research. The first problem is the low volume and sparseness of the result data, which does not enable any meaningful statistical analysis to be applicable for the validation process. Some related work of EBSE databases do exist, but are constructed in a significantly more simplistic manner making direct comparison difficult. Despite this, to demonstrate the improvement of utility compared to the existing databases, the author chose to duplicate the data available on the SEEDS and BPCH databases for the top code review practices (see Appendix III). The author performed a small scale literature review by entering duplicates of all relevant articles found from the two other EBSE databases. The protocol of the literature review was to first look for related articles given the topic (ie. code reviews), second to fetch the article metadata (i.e. authors, title), the practice related data (by mapping it on the practice ontology), the context (i.e. students, space industry) and the goals reported. Finally the paper was searched for the result on the relationship of the given goal and practice, the sample size and the reliability of the result. The data was entered into the EBSE data model (see Appendix IV). The objective of the validation is to check which of the three databases provides the most useful model for empirical software engineering result analysis. The evaluation of the EBSE DB is done by comparatively checking the consistency of the proposed data model to the related two databases and trying to find modeling failures in respect to the problem at hand.

First the author scanned the both databases looking for overlapping results on the top practices. On the SEEDS -database the author identified categories “Scrum” and “XP” to include the practice ranked high by EBSE the “short iterations”, but noticed that their classification entailed respectfully the whole methodologies and thus the relevance of the results for one practice would have been low. However, the author found the categories “inspection” from the SEEDS and “formal reviews” from the BPCH databases that looked like comparable with 3rd highest ranking practice “code reviews”. The author scoped the validation literature review down to this practice and the respectful categories. The SEEDS database contained 15 and the BPCH 11 reference articles (see the Appendix III). The code reviews are the method that has probably the most widely studied topic on the software engineering, which was reflected by the fact that both databases contained a high number of articles of the topic. However, to perform a valid review the author should have included an even higher number of articles. Thus the reliability of this validation attempt can be questioned.

Nevertheless, an unexpected discovery was made during review of the related databases. While looking for papers related to the “short iterations” the author noted papers describing the effects of whole methodologies towards certain sets of goals. However, the methodologies such as Scrum and XP are aggregates of individual practices. Entering this type of data in the EBSE database raises a problem that would need to be solved either by creating a new practice class “Methodology” and mapping the practices to it using a new aggregate relationship “isUsedBy”, or by splitting the methodology –results into arrays of practices already in the data entry phase. Also a new inference rule would be required to deduct the effect of a single practice towards a goal as a component of a methodology. Of course, the evidence provided by such analysis is much lower than an explicit result on a single practice, but these reports nevertheless contain some data that can be used to extract information by using a sufficiently large sample set. One surprising solution could be to use the QGP -matrix plugin for the analysis, since it already implements a similar model of calculating the effects of a practice vector towards a goal vector. A new application of the EBSE DB could be to construct new optimal methodologies based on the evidence data, or to validate and compare the existing methodologies. The downside of this type of analysis is the arbitrariness of the categorizations found in the articles lacking the exact breakdowns which practices are regarded as components of e.g. the Extreme Programming [Beck01], or to which extent the subjects have been able to use each component practice.

The SEEDS -database does not contain any codified numeric or relational goal data. Thus, the utility of the database was rather low as a mere index of articles. The author was forced to extract the verification data manually. In this sense, the suggested new system produces significantly higher utility. The comparability issues became evident while codifying the SEEDS data. The new system is lacking functional inference rules for example for the unit conversion. Thus a substantial care was necessary during the data entry to check whether the units for the metric "ratio of total defects found" were raw numbers 0.25 or as percentage 25%. Yet another finding was that some of the referred papers are of a rather low quality with hard to interpret results [AIII/Kelly00]. For example the "total number of defects found per method" was reported, but the "efficiency" (man hours/defect) was not reported explicitly, although it seems that it was tracked by the authors of the article.

The author was unable to enter the results of one SEEDS article on code reviewer personality suitability [AIII/daCunha07] into to the data model. While the paper reported clean results on the personality types of the programmers, the author did not find a clear way to enter the reported indicators for code review performance, as the schema can hold only classes for practices and context. It would be possible to enter the results of this paper as a normal practice result, but the main (and the most interesting) result of the paper would be excluded. The SEEDS resolves this problem by holding only a summary of each paper, but also lacking any codified information that would enable the users to understand the data without full text search of the complete database. In future it would be interesting to provide coding for the performance indicators, or estimates on the driver metrics for each practice. However, this is clearly out of scope of this work and there is only a limited subset of papers describing these topics. It would not be impossible to add modeling of this sub-field into the database, if it is found valuable from the research or commercial point of view, for example to provide psycho-analysis models for employee recruitment.

The BPCH -database contained also some information about the relations towards some goals. However, the author noted only two goals (quality and cost) being mentioned on the overview. The rating was also rather arbitrary (improved, reduced), while the reference point was not mentioned from which to where the quality metric was improved or reduced. For some articles also the "net impact" was extracted in a written format such as:

"p4: Barry Boehm reported that at TRW it was found that early prevention effort (via reviews, inspections, and analysis tools) had a 5:1 or 10:1 payoff." [Shull02],

but this information was not available for all references. The BPCH -database did not contain the links to the original references, making validating the claimed results difficult or impossible. The author was unable to find access to four referred articles; such as the NASA technical reports [AIII/Kelhorst & AIII/Eagan86], while the IEEE Digital Library is missing the volume 12, although does have all other volumes scanned from the previous and past years. Several papers [e.g. AIII/Rifkin94 & AIII/Basili96] contained results on specific practice variations such as the “inspection training” and the “perspective-based reading” rather than the main BPCH classification title “Software Formal Inspections”.

Another new modeling challenge was discovered while evaluating the Mars Climate Orbiter mission failure report [AIII/JPL99], reporting on which specified practices were **not** used. The BPCH -database fails to model this type of evidence and has omitted the net impact -field. The author also failed to include the evidence in the new system, as the model lacks annotation for a counter-practice or the negation of a goal (result as a failure instead of a success). Another hindrance was the open world assumption –paradigm, which does not allow inference on negations unlike for example on traditional relational databases. Due to the modeling failure, author excluded this paper from the review. Yet one issue requiring analysis is the type of result, where two or more practices are pair-wise compared and the result is reported as a ratio. None of the databases seem to have taken account how to model this type of a result. This can be circumvented in the proposed system by reporting the original results in raw format, if available. However, the new system does not feature inference rules to redraw the comparative conclusions from the raw data yet. Overall, the BPCH -database seems to hold the extractable goal-practice relationship information, but in a non-codified format, which can’t be utilized by the users, unlike on the proposed new system.

A significant discovery was made when the literature was compared to the QPA -data. It seems that the experimental literature reports focus almost solely on results that report only a few metrics, which are completely different to those found in this research. For example the SLR on the code reviews on the two related databases produced reports only of two metrics (efficiency and effectiveness) and their variations: “number of defects found per hour”. Economic related variations to this included “return on investment” and “ratio of rework hours saved per invested man hour”. The QPA, however, recorded the code reviews to be contributing to a multitude of other goals such as “maintainability”, “installability”, “updateability”, “scalability” and “performance”. None of these are previously reported in the papers available on the two related database.

Whether this verification was reliable is also questionable, since the two databases did not contain any overlapping references to the same articles. However, the verification demonstrates the advanced utility of the new proposed system over the existing ones for the practioners, since it contains codified goal-practice relationship data. Of the evaluated systems, the SEEDS -database contained the most evidence and was also the most easily verifiable. However, the usefulness of the SEEDS was low due to lack of goal-practice relationship data modeling. The BPCH contains this data, but in a format that is not usable for the end-users. The proposed prototype system lacks currently all reference features (descriptions, links, summaries), but provides substantially more practical utility for end users. Further, two new modeling issues were found that none of the three databases manage to model correctly. These design requirements can be included if the system is expanded to perform tasks outside of its original scope of QPA -matrix analysis.

6.4 Validation Summary

The amount of the collected data was found to be too sparse to perform any conclusive statistical analysis to validate the findings. However, the prototype system constructed by using the explorative interventions suggests that the new Semantic Web –based model seems to provide very useful and practical data for performing Software Process Improvement. The research should be continued by collecting a larger data set from both primary and secondary sources. This would enable the system to provide answers for a substantially larger quantity of questions by an increasing validity.

The all three evaluated EBSE DBs were found to contain several modeling errors that should be corrected in the future systems. Nevertheless, the new semantic web –based construction seems to have the predictive property of providing similar SPI -suggestions from the cross-company benchmarking data than the companies were able to provide themselves during the QPWS workshops. However, this finding needs to be verified further by collecting more data points.

7. Conclusions

Based on the collected data it seems evident that the QPA -method is time consuming and produces results of both low validity and low relevance. From the industry point of view the temporal footprint of the QPA is roughly 5 times too heavy to be practically acceptable for the subject companies. The maximal footprint would be maximum 2% of the project effort. From the academic and the constructed EBSE DB point of view it also fails to produce research data in a sufficient quantity to enable publication of valid enough research results on the relationship between quality goals and practices. However, the QPA -matrix produces intersection of practice portfolios that do contain unique information about the combinations of the practices not available from elsewhere in the literature and could be reported as case studies. In the Future this type of data could be used for primary research in a lighter form.

The findings for the main research question are two-fold. In the absence of an operational EBSE DB that is capable of providing definitive evidence that it is possible to reach a specific quality level using some practices, it is evident that the method NMA produces SPI -initiatives in an almost 10-fold higher quantity than the other methods. However, according to [Battram99] due to cultural constraints the company employees themselves are able to map only a smallish close-by subset of the possibilities that would be open. The problem with the social search methods such as the NMA is that they are unable to provide any proof that a certain quality level could be reached by implementing the suggestions. Thus another approach is required for SPI that would enable improvement beyond the subjective knowledge of the employees to lower the risk factor.

It seems that it is theoretically possible to provide an answer to some specific instances of the main research question by using the EBSE DB. Unfortunately the range of answerable questions depends on the population data, which is currently too sparse to provide an answer that could be considered reliable. The author was able to produce a proof-of-concept prototype -database providing a few answers to questions such as "*what practices should be used for reaching 8h effort per update*" with an answer of practices "smokeTesting" and "alphaBetaTesting" and the un-normalized realibility level of 0.0476 in the context of the given companies. Unfortunately the data is insufficient yet to calculate the degree, how the results can be generalized from one company (context) to another, which means the results might contain a substantial error. In the absence of data the database is also unable to answer, how for example an update effort of 1 hour less could be reached. Thus, the EBSE DB should be first populated by performing systematic literature reviews of all scientific data on software engineering practices available. The reliability level is a bare number related to the QPA –matrix analysis and should be normalized in the future by studying the relationship of the QPA –analysis and the literature review results. By mapping all pre-existing data on the relationship of an individual practice and one goal the researchers would be more able to produce research results that would have practical and scientific relevance than by performing primary research first. After the collection of all evidence from the secondary sources, a standard reporting format (e.g. a XML/OWL –based machine interpretable schema) could be formulated to facilitate the compability of the future primary studies to the EBSE DB. From the knowledge database point of view it is evident that an in-depth, high-N study on a single relationship has more value than a vague scan of a large practice portfolio. After this the EBSE DB can be used also for primary study, but a more light-weight and precise method than the QPA should be developed. One possibility is to construct a pre-assignment tool for data entry that would provide means for faster and higher volume data collection than the workshops do currently.

From the subject company point of view a scientific report on the quality goal effects on a single software engineering practice is of low interest. The companies are more concerned of the overall organization the whole software engineering departments. This could motivate the quality improvement -oriented companies to incline towards investing in construction of an experience factory. Probably the most cost effective implementation would be an industry or world-wide evidence database benchmarking results provided by the literature to the context specific measurements of individual software engineering teams.

The author envisions such a community driven system to be able to collect data on the current practices and the goals of the companies, and analyzing this information to make automated inference on how the companies should improve their processes. The same data could be used by the researchers to construct new inference rules and papers on the relationships between the goals, practices and the context. However, it's uncertain whether companies would agree to enter their sensitive goal and practice data on a publicly available database. It is also unknown whether this kind of system should be operated by a commercial or an academic organization.

Both the SEEDS and BPCH lack a sufficient ontological hierarchy to distinguish the practice variants from each other. Both databases classify results of for example the "scenario-based reading" and the "task-directed inspection" under label "code reviews", while the results can't be directly generalized to the parent category. Perhaps since the databases lack also the exact codification of the result information, sufficient rigor in result codification has not been followed and it is unrealistically assumed to be performed by the end-user. In comparison, the suggested new database does provide means for a proper result codification.

Compared to the other available EBSE -databases, the new system enables unique mapping of the performance relationship between specific practices and goal levels. The reference data in the system can be also converted into a web based reference index and description wiki, thus surpassing the functionality of the other systems significantly. However, the current prototype lacks the required XSLT transformations and the input data of the practices to convert the RDF/OWL -database to a web service, as this feature is trivial to implement in the future. A yet further use case of the EBSE DB is to provide educational information of the software engineering practices. The web-based database could be also used as a resource for online-courses in software engineering curriculum reducing the adaptation delay of the most recent research results to the industry.

8. Future Research

After the construction of the EBSE DB the author found a comparative study of different possibility space search methods, where the Method 635 [Rohrbach69] was found to produce 95% higher quantity and higher yield of 94% for acceptable ideas than the brainstorming and the other methods compared [Tyllinen09]. Thus, a further efficiency could be achieved by experimenting with a "New Method B" (NMB), where the idea generation method would be replaced to a brain-writing method such as the Method 635. The name of the method comes from the six participants writing three ideas in five minutes and then passing the ideas to the next person. Otherwise this process is similar to the brainstorming.

The EBSE -knowledge database offers substantial opportunities for advancing the research on software engineering by providing a single medium to report and analyze the results world-wide. However, it is not in the scope of this thesis to implement such a full-scale system or in the interest of the author to maintain such a system for a pure academic interest. The side contribution of this work is to evaluate the usefulness of the semantic web -based approach for advancing the scientific knowledge and the industrial practice.

The new system requires a range of further improvements. Firstly a substantially larger dataset would be required to provide meaningful information of practical utility. This can be most easily achieved by utilizing the database as a platform for composition of systematic literature reviews [Kitchenham04]. While the semantic database is based on the first order predicate logic, any analysis performed on the dataset has less possibility for error than a review done by a human and is though more reliable. The remaining task is to search and enter all available literature data on the goal-practice relationships and to evaluate the validity and reliability of the sources. It is unclear whether an automatic data retrieval system could be effectively used for this kind of task, although many such systems have been proposed [Cruzes07].

Secondly, the system is capable of providing novel information about context and goal interdependencies. This would require the Goal and Context -classes to be implemented as Bayesian -networks [Jordan99]. The author studied opportunities to utilize the BayesOWL [Ding04] and the PR-OWL [daCosta05] Bayesian Network extensions, but was unsuccessful to produce any meaningful inferences due to project time constraints.

One further interesting information extractable from the data is the classification of the practices. For the ontological purposes it would be valuable to extract information on how the practices can be categorized orthogonally to create a conceptually minimized language for software engineering practice nomination.

Different practice ontologies, classifications and even complete methodologies like Extreme Programming could be compared against the orthogonal practice ontology to test their expressivity and ambiguousness. This can be done by grouping which practices contribute towards which quality goals. Such a classification could be constructed for example by using self-organizing maps [Kohonen82]. The objective would be to create new methodologies and classifications that would enable faster adaptation of the latest advances in EBSE in the industry and also lessen the extent of the classification error for the forthcoming research.

At least two types of articles could be written on the results of this thesis. First, the description of the database could be published in a similar fashion to [Janzen09] and [Shaw07]. The approach in this paper is novel and superior to the previous state of art, while it combines results from several other top research fields such as the Semantic Web and Evidence Based Software Engineering. When the database can be developed to the full scale, further papers can be written on the topics of application of the Bayesian networks to the EBSE, generalizability of the results across practices, goals and contexts; and using the database to enhance industrial and educational learning performance. An even wider range of papers could be written of the results on the results calculated by the database. When a full scale systematic literature review is performed by entering all available research data in to the database, a new paper of each practice (or category of practices) can be written at regular intervals. Based on the early data, there are at least a hundred of different practice category topics, even when the practice sub-flavors are combined under a single classification. It seems that the topic and the solution of this thesis are not only sufficient to facilitate research for a dissertation, but also to employ a large research group for a better part of a decade. For those willing to produce the primary research results, the database offers a common machine-readable reporting format and interface, which would enable acceleration in both production and SLR-based analysis of the results to advance the research in the field faster compared to the contemporary progress by pointing out the most demanded research topics. The author would suggest publishing an OWL-based reporting format for machine interpretable analysis of the research data as a conference paper, or even as a RFC or other standard. The database also opens new opportunities for a previously weakly studied research field on software engineering methodologies, or how the collections of practices affect quality goals. The organization willing to apply the approach suggested in this thesis is likely to gain the status of the world's leading author in the field of empirical software engineering due to the sheer volume of previously unknown high quality results and conclusions that the database is able to produce.

9. Summary

Four software process improvement methods and three knowledge databases were compared in this work. The methods can be firstly categorized by the user requirements they are designed to solve. The Quality Palette Analysis seems to have been designed primarily to collect data on the current practices the companies are using and their effects on the quality goals. The two new methods produced by the constructive interventions by the author (NMA & IA) were designed by the primary purpose of corporate SPI, and do not take account the needs of the researchers in any fashion. The fourth method (QPA for Features) that was the one suggested by a subject company seems to be a trade-off between the corporate and the researcher requirements. The final construction; the EBSE Semantic Web –based experience factory knowledge base, was found potentially usable for the industry, the academia as well as for the education. Four mid-sized Finnish software product companies were used as subjects for data collection and method application.

The main research question was "*how the software engineering companies can ensure reaching of a specific quality goal?*" The author recognizes two basic approaches in selecting the SPI -initiatives: either the subjects map the possibility space based on their collective personal knowledge (such as in NMA & IA), or the decisions are based on external evidence provided either by consultants, literature (adopting a complete methodology such as the Extreme Programming [Beck00]), an experience factory or an EBSE -database. Of the two the author has to prefer the second alternative, with the precaution that the external data source containing sufficiently advanced, recent, complete and reliable information on how the software engineering teams should be organized is actually available. Secondly, the practices suggested by the sources should be implementable by the target organization. Unfortunately, none of the external related knowledge bases or other data sources available currently meets these requirements, so the companies have to rely on the personal knowledge, which is often limited and speculative. Of the evaluated methods, it seems that the NMA –style brainstorming is the most light-weighted and effective method for SPI idea generation. However, due to sparseness and low volume of collected data, this study does not provide any conclusive answer, whether the EBSE -database or the NMA –style brainstorming produces higher quality performance improvements in practice, and should be further studied. The QPA-method seems to consume the resources of the companies in an unnecessary quantity, while simultaneously producing low volume of SPI suggestions. The inefficiencies can be linked with the implicit purpose of the method to gather research data rather than from producing effective SPI initiatives for the companies.

However, if a fully functional EBSE experience factory could be developed and populated by the systematic literature reviews, it could provide a more efficient and optimal method for the SPI. By cross-tabulating the QPA -data across the companies the EBSE DB provided a few SPI suggestions for the subject companies based on the practices the other companies are using. Unfortunately the data remains insufficient to confirm that the statements claimed in this work would be reliable or generizable between different contexts, such as even between the subject companies.

For validation of the results it seems that the suggestions invented during the NMA –brainstorming session at the Company D overlaps to some extent with the suggestions calculated by the EBSE DB -based on the QPA –matrices from the three other companies, giving a promise of the potential predictive power of the knowledge database. However this might be coincidental. A substantial work would be required to develop an industrially or research applicable system and to populate it with all available secondary research data. The value of the suggested system seems to be very high in optimizing the organization of both the software engineering companies and advancing the research of the empirical software engineering.

References

Akao90 Akao Y., Quality Function Deployment. Integrating Customer Requirements into Product Design., Productivity Press, USA, 1990.

Althoff01 Althoff K-D., Decker B., Hartkopf S., Jedlitschka A., Nick M., Rech J., Experience Management: The Fraunhofer IESE Experience Factory. Proceedings of Industrial Conference on Data Mining, Leipzig, July 24-25, 2001. Fraunhofer IESE report no. 0.35.01/E, Germany, 2001.

Bach97 Bach J., Good Enough Quality: Beyond the Buzzword, IEEE Computer Society, p. 96-98, August 1997, USA.

Barcacci03 Barcacci M.R., Ellison R.J., Weinstock C.B., Wood W.G., Quality Attribute Workshops (QAWs) Third Edition, Technical Report CMU/SEI-2003-TR-016, ESC-TR-2003-016, SEI, USA, 2003.
http://www.sei.cmu.edu/reports/03tr016.pdf

Barners-Lee01 Barners-Lee T., Hendler J., Lassila O., The Semantic Web. Scientific American, May 2001, USA, 2001.

Battram99 Battram A., Navigating Complexity. The Essential Guide to Complexity Theory in Business and Management. The Industrial Society, USA, 1999.

Basili92 Basili V.R., The Experience Factory – Can it Make You a 5?, Seventeenth Software Engineering Workshop, NASA/Goddard Space Flight Center, College Park, MD, Dec 1992.
http://www.cs.umd.edu/~basili/publications/technical/T78.pdf

Basili94a Basili V.R., Caldiera G., Rombach H.D., The Experience Factory, Encyclopedia of Software Engineering, p. 469-476, John Wiley & Sons, USA, 1994.
ftp://ftp.cs.umd.edu/pub/sel/papers/fact.pdf

Basili94b Basili B.R., Cadiera G., Rombach H.D., The goal question metric approach. Encyclopedia of Software Engineering, John Wiley & Sons, USA, 1994.

Beck00 Beck K., Extreme Programming Explained: Embrace Change, Addison-Wesley, USA, 2000.

Beck01 Beck K., et.al., Agile Manifesto. http://agilemanifesto.org/, USA, 2001.

Berczuk02 Berczuk S.P., Software Configuration Management Patterns. Effective Teamwork, Practical Integration, Addison-Wesley, USA, 2002.

Biehl04 Biehl R.E., Six Sigma for Software, IEEE Software, Vol.21 (2), p.68-70, USA, 2004.

Boehm01 Boehm B., Basili V., The CeBASE Framework for Strategic Software Development and Evolution. EDSER-3 position paper, USA, 2001.

Bontis98 Bontis N., Intellectual capital: an exploratory study that develops measures and models, Management Decision, 36/2, p. 63-76, USA, 1998, http://www.business.mcmaster.ca/mktg/nbontis/ic/publications/MDBontis.pdf .

Brooks95 Brooks F.P.Jr., The Mythical Man-Month, Addison-Wesley, 1995.

Cooper95 Cooper R., When Lean Enterprises Collide, Harvard Business School Press, Boston, USA, 1995.

Coplien95 Coplien J., Organizational Patterns, in “Coplien J., Schmidt D., Pattern Languages of Program Design”, p. 183-237, Addison-Wesley, USA, 1995.

Crosby79 Crosby P., Quality is free, McGraw-Hill, New York, USA, 1979.

Cruzes07 Cruzes D., Mendonça M., Basili V., Shull F., Jino M., Automated Information Extraction from Empirical Software Engineering Literature: Is that possible? First International Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, 2007.

daCosta05 da Costa P.C.G., Bayesian Semantics for the Semantic Web, Ph.D. Dissertation, George Mason University, Fairfax, VA, USA, 2005.

Deming86 Deming W.E., Out of the Crisis, MIT Press, USA, 1986.

Ding04 Ding Z., Peng Y., A Probabilistic Extension to Ontology Language OWL, Proceedings of the 37th Hawaii International Conference on System Sciences, USA, 2004.

Dorling93 Dorling A., SPICE: Software Process Improvement and Capability dEtermination. Software Quality Journal, Vol. 2, p. 209-224, USA, 1993.

Drucker54 Drucker P., The Practice of Management, USA, 1954.

Drucker85 Drucker P., Innovation and Entrepreneurship, Harper & Row, USA, 1985.

Fagan76 Fagan M.E., Design and Code Inspection to reduce errors in program development, IBM Systems Journal, Vol. 15 (3), p. 182-211, USA, 1976.

Feigenbaum51 Feigenbaum A.V., Quality Control: Principles, Practice, and Administration. McGraw-Hill, USA, 1951.

Fitzgerald99 Fitzgerald B., O'Kane T., A Longitudinal Study of Software Process Improvement. IEEE Software, May/June 1999, p.37-45, USA, 1999.

Folan05 Folan P., Browne J., A review of performance measurement: Towards performance management, Computers in Industry, Vol. 56, p.663-680, 2005.

Ford22 Ford H., My Life and Work, USA, 1922.
http://www.gutenberg.org/etext/7213

Gao06 Gao Y., Kinoshita J., Wu E., Miller E., Lee R., Seaborne A., Cayzer S., Clark T., SWAN: A distributed knowledge infrastructure for Alzheimer disease research. Journal of Web Semantics, Web Semantics: Science, Services and Agents on the World Wide Web, (4) 2006, p. 222-228, USA, 2006.

Goldratt84 Goldratt E.M., Goal: Process of Ongoing Improvement, 2nd Edition, Gower Publishing, UK, 1984.

Hauser88 Hauser J.R., Clausing D., The House of Quality, Harward Business Review, Vol 66 (3), May-Jun, p. 63-73, USA, 1988.

Hackos98 Hackos J.T., Redish J.C., User and task analysis for interface design. John Wiley & Sons, USA, 1998.

Harry00 Harry M., Schroeder R., Six Sigma, Doubleday, USA, 2000.

Hätinen06 Hätinen A.J., Kahden käyttöliittymäsuunnittelumenetelmän vertailu – virtuaali-ikkunat ja skenaariopohjainen suunnittelu, M.Sc. Thesis, University of Helsinki, Department of Computer Science, Finland, 2006.

ISO9000 TC 176/SC (2005), ISO 9000:2005, Quality management systems – Fundamentals and vocabulary, International Organization for Standardization, 2005.

ISO9162 ISO / IEC, ISO Std 9162 FDIS version 10.9., 2.10.2008.

Itkonen07 Itkonen J., Quality Practices in Time-Paced Software Development, unpublished SHAPE – project workbook, Chapter 4, SoberIT, Helsinki University of Technology, Finland, 2007.

Ittner03 Ittner C.D., Coming up Short on Nonfinancial Performance Measurement, Harward Business Review, Nov. 2003, p. 88-95, USA, 2003.

Janzen09 Janzen D.S., Ryoo J., Seeds of Evidence: Integrating Evidence-Based Software Engineering. 21st Conference on Software Engineering Education and Training, IEEE Computer Society, USA, 2008.

Juran74 Juran J.M., Gryna F.M., Bingham R.S., Quality Control Handbook, McGraw-Hill, New York, USA, 1974.

Jordan99 Jordan M.I., Learning in Graphical Models, MIT Press, USA, 1999.

JPL99 JPL D-18441, Report on the Loss of the Mars Climate Orbiter Mission, JPL Special Review Board, Jet Propulsion Laboratory, California Institute of Technology, USA, 1999.

Kano84 Kano N., Seraku N., Takahashi F., Tsuji S., Attractive quality and must-be quality. Quality: The Journal of Japanese Society for Quality Control, 14 (April), p. 39-48, Japan, 1984.

Kaplan92 Kaplan R., Norton D.P., The Balanced Scorecard – The Measures That Drive Performance, Harward Business Review, Jan/Feb, USA, 1992.

Kaplan04 Kaplan R., Norton D.P., The Strategy Maps: Converting intangible assets into tangible outcomes, Harward Business School Press, Boston, USA, 2004.

Kay93 Kay J., Foundations of Corporate Success, Oxford University Press, UK, 1993.

Kitchenham97 Kitchenham B., Linkman S., Pasquini A., Nanni V., The SQUID approach to defining a quality model. Software Quality Journal, Vol 6., p. 211-233, USA, 1997.

Kitchenham04 Kitchenham B.A., Dybå T., Joergensen M., Evidence-based Software Engineering, Proceedings of the 26th International Conference on Software Engineering, ICSE'04, 2004.

Kohonen82 Kohonen T., Self-organized formation of topologically correct feature maps. Biological Cybernetics, 43:59-69, 1982.

Kontio96 Kontio J., Caldiera G., Basili V.R., Defining Factors, Goals and Criteria for Reusable Component Evaluation. CASCON'96, Canada, 1996.

Lewin46 Lewin K., Action Research and Minority Problems, Journal of Social Issues, vol. 2, p. 34-46, USA, 1946.

Lewin51 Lewin K., Field Theory in Social Science, University of Chicago, Chicago, USA, 1953

Luomansuu09 Luomansuu R., Tilastollinen tutkimus ohjelmiston laatuun vaikuttavista tekijöistä, M.Sc.(Eng) Thesis, Lappeenrannan teknillinen yliopisto, Tuotantotalouden osasto, Finland, 2009.

Martin98 Martin M.V., Kmenta S., Ishii K., QFD and the Designer: Lessons from 200+ Houses of Quality. World Innovation and Strategy Conference, Stanford University, USA, 1998.

Napier08 Napier N.P., Kim J., Mathiassen L., Software Process Re-engineering: A Model and Its Application to an Industrial Case, SPIPFL Sep-Oct 2008; 13 (5): p.451-471, Wiley 2008.

Nosek98 Nosek J.T., The case for collaborative programming. Communications of the ACM, 41 (3), p. 105-108., USA, 1998.

NIST02 NIST: The economic impacts of inadequate infrastructure for software quality, 2002.

Osborn63 Osborn A.F., Applied Imagination: Principles and procedures of creative problem solving. 3rd revised edition, Charles Scribner's Sons, New York, USA, 1963.

Pinkster04 Pinkster I., van de Burgt B., Janssen D., van Veenendaal E., Succesful Test Management – An Integral Approach, LogicaCMG, Springer, The Netherlands, 2004.

Pol02 Pol M., Teunissen R., van Veenendaal E., Software Testing. A Guide to the TMap Approach, Addison-Wesley, USA, 2002.

Poppendieck03 Poppendieck M., Poppendieck T., Lean Software Development: An Agile Toolkit. Addison-Wesley, USA, 2003.

Regnell08 Regnell B., Svensson R.B, Olsson T., Supporting Roadmapping of Quality Requirements. IEEE Software, 25 (2), p. 42-47, 2008.

Rohrbach69 Rohrbach B., Creative nach Regeln: Methode 635, eine neue Technik um Losen von Problemen. Absatwirtschaft, Vol. 12, Bundesrepublik Deutschland, 1969.

Royce70 Royce W.W., Managing The Development of Large Software Systems, Proceedings, IEEE WESCON, Aug 1970, p. 1-9.

Saaty80 Saaty T.L., The Analytical Hierarchy Process. McGraw-Hill, USA, NY, 1980.

Schein99 Schein E., Corporate Culture survival guide: Sense and nonsense about cultural change, Jossey-Bass Publishers, San Francisco, USA, 1999.

Senge90 Senge P.M., The Fifth Discipline. Century Business, London, UK, 1990.

Sommerville01 Sommerville I., Software Engineering, 6th edition, Addison-Wesley, USA, 2001.

Shaw07 Shaw M.A., Feldmann R.L., Shull F., Decision Support with EMPEROR. Proceedings of the First International Symposium on Empirical Software Engineering and Measurement, ESEM, p.495, USA, 2007.

SQUID08 Itkonen J., Mäntylä M.V., Vanhanen J., Lassenius C., SQUID – Goal and Evidence-Based Quality Practice Selection and Improement – Sub-project of ESPA – Research Plan, Software Business and Engineering Institute, Helsinki University of Technology, 29.1.2008.

Tanaka89 Tanaka M., "Cost Planning in the Design Phase of a New Product", in (ed.) Mondem Y., Sakurai M., Japanese Management Accounting, p. 49-71, Productivity Press, Cambridge, MA, USA, 1989.

Taguchi92 Taguchi G., Taguchi on Robust Technology Development, ASME Press, 1992.

Tyllinen09 Tyllinen M., Käyttäjien osallistaminen ideointiin osana toiminnanohjausjärjestelmän konseptikehitystä. M.Sc.(eng.) Thesis, TKK, Finland, 2009. http://www.soberit.hut.fi/T-121/shared/thesis/di-Mari-Tyllinen.pdf

Vanhanen09 Vanhanen J., Itkonen J., Mäntylä M.V., A method for Setting Software Quality Goals And Initial Experiences of Its Use, unpublished work in progress, SoberIT, Helsinki University of Technology, Finland, 2009.

Vonderembse88 Vonderembse M.A., White G.P., Operations Management: Concepts, Methods, Strategies, Wiley 1988.

W3C04 Editors: McGuiness D.L., van Harmelen F., OWL Web Ontology Language, W3C Recommendation 10 February 2004, http://www.w3.org/TR/2004/REC-owl-features-20040210/.

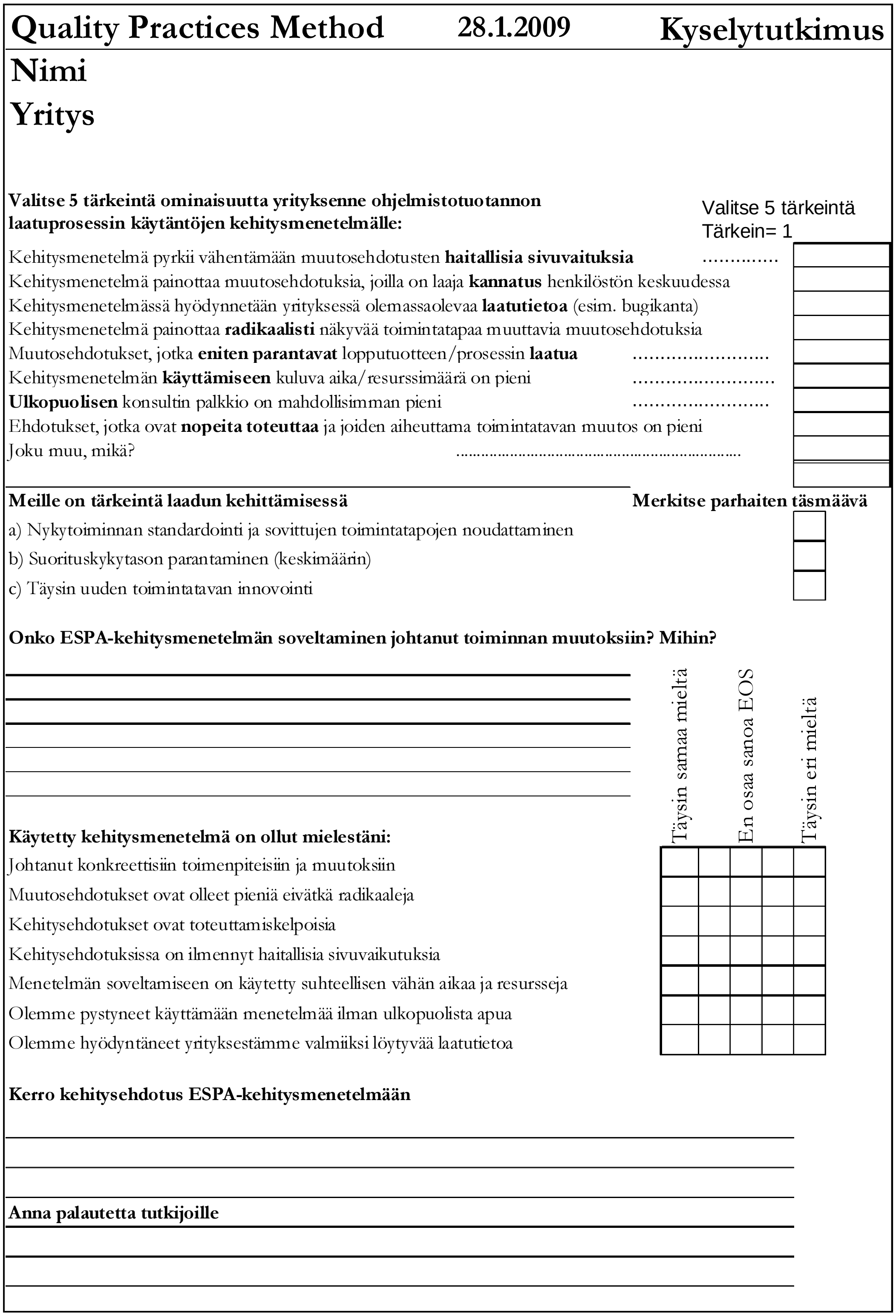
Weisbord00 Weisbord M., Janoff S., Future Search – An Action Guide to Finding Common Ground in Organizations & Communities. Berrett-Koehler, USA, 2000.

White81 White F.M., Locke E.A., Perceived Determinants of High and Low Productivity in Three Occupational Groups: A Critical Incident Study, Journal of Management Studies, vol. 18 (4), p.375-387, USA, 1981.

Xia08 Xia L., Conitzer V., Lang J., Voting on Multiattribute Domains with Cyclic Preferential Dependencies, Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, p. 202-207, Association of Advanced Artificial Intelligence, USA, 2008.

Zultner93 Zultner R.E., TQM for Technical Teams, Communications of the ACM, Vol. 36, No. 10., p. 79-91, October 1993.

Appendix I – Survey Questionnaire



Quality Practices Method **28.1.2009** **Kyselytutkimus**

Nimi

Yritys

Valitse 5 tärkeintä ominaisuutta yrityksenne ohjelmistotuotannon laatuprosessin käytäntöjen kehitysmenetelmälle:

Valitse 5 tärkeintä
Tärkein= 1

Kehitysmenetelmä pyrkii vähentämään muutosehdotusten **haitallisia sivuvaituksia** ………….. □
Kehitysmenetelmä painottaa muutosehdotuksia, joilla on laaja **kannatus** henkilöstön keskuudessa □
Kehitysmenetelmässä hyödynnetään yrityksessä olemassaolevaa **laatutietoa** (esim. bugikanta) □
Kehitysmenetelmä painottaa **radikaalisti** näkyvää toimintatapaa muuttavia muutosehdotuksia □
Muutosehdotukset, jotka **eniten parantavat** lopputuotteen/prosessin **laatua** …………………….. □
Kehitysmenetelmän **käyttämiseen** kuluva aika/resurssimäärä on pieni …………………….. □
Ulkopuolisen konsultin palkkio on mahdollisimman pieni …………………….. □
Ehdotukset, jotka ovat **nopeita toteuttaa** ja joiden aiheuttama toimintatavan muutos on pieni □
Joku muu, mikä? ………………………………………………………… □
□

Meille on tärkeintä laadun kehittämisessä **Merkitse parhaiten täsmäävä**

a) Nykytoiminnan standardointi ja sovittujen toimintatapojen noudattaminen □
b) Suorituskykytason parantaminen (keskimäärin) □
c) Täysin uuden toimintatavan innovointi □

Onko ESPA-kehitysmenetelmän soveltaminen johtanut toiminnan muutoksiin? Mihin?

____
____
____
____
____
____

Käytetty kehitysmenetelmä on ollut mielestäni:	Täysin samaa mieltä		En osaa sanoa EOS		Täysin eri mieltä
Johtanut konkreettisiin toimenpiteisiin ja muutoksiin	□	□	□	□	□
Muutosehdotukset ovat olleet pieniä eivätkä radikaaleja	□	□	□	□	□
Kehitysehdotukset ovat toteuttamiskelpoisia	□	□	□	□	□
Kehitysehdotuksissa on ilmennyt haitallisia sivuvaikutuksia	□	□	□	□	□
Menetelmän soveltamiseen on käytetty suhteellisen vähän aikaa ja resursseja	□	□	□	□	□
Olemme pystyneet käyttämään menetelmää ilman ulkopuolista apua	□	□	□	□	□
Olemme hyödyntäneet yrityksestämme valmiiksi löytyvää laatutietoa	□	□	□	□	□

Kerro kehitysehdotus ESPA-kehitysmenetelmään

____
____
____

Anna palautetta tutkijoille

____
____
____

Appendix II – Quality Practices and Evidence

Goal	Practice	Evidence Goal Value Unit
qgp:quality	shortIterations	1.0485
qgp:quality	userObservation	0.9516
qgp:quality	codeReviews	0.7592
qgp:quality	codingConvention	0.6825
qgp:quality	designMeeting	0.6796
qgp:quality	functionalSpecificationReview	0.6649
qgp:quality	pairProgramming	0.6610
qgp:quality	technicalDesignReview	0.6500
qgp:quality	functionalSpecificationAcceptance	0.6292
qgp:quality	acceptanceTestingByProductOwner	0.6290
qgp:quality	regressionTesting	0.6290
qgp:quality	unitTesting	0.6240
qgp:quality	planningDocumentation	0.6213
qgp:quality	functionalTesting	0.6137
qgp:quality	trainingFeedback	0.6000
qgp:quality	outsourcedFunctionalTesting	0.5791
qgp:quality	usabilityAnalysis	0.5195
qgp:quality	userDocumentation	0.4874
qgp:quality	competitorBenchmark	0.4766
qgp:quality	changeManagementDocumentation	0.4608
qgp:quality	selfStudy	0.4343
qgp:quality	implementationDocumentation	0.4187
qgp:quality	alphaBetaTesting	0.4154
qgp:quality	manualTestingOfCustomerProcesses	0.4032
qgp:quality	newRequirementManagementProcess	0.3847
qgp:quality	nightlyBuild	0.3736
qgp:quality	acceptanceReview	0.3648
qgp:quality	designGuideline	0.3642
qgp:quality	demoByCustomer	0.3538
qgp:quality	functionalTestScenarios	0.3396
qgp:quality	manualFunctionalTesting	0.3239
qgp:quality	configurationValidationTools	0.3239
qgp:quality	usabilityTesting	0.3177
qgp:quality	performanceTesting	0.3069
qgp:quality	customerBenchmark	0.3000
qgp:quality	functionalSpecificationWithCustomer	0.3000
qgp:quality	iterativeAndIncrementalDevelopment	0.2999
qgp:quality	tenderReviews	0.2616

qgp:quality performanceMonitoring	0.2409
qgp:quality realisticTestingEnvironment	0.2406
qgp:quality realisticProjectScopeAndSchedule	0.2308
qgp:quality automatedRegressionTests	0.2276
qgp:quality groupWalkthroughUIDesignEvaluation	0.2204
qgp:quality recursiveUIDesign	0.2204
qgp:quality guiPrototyping	0.2113
qgp:quality changeRequestManagementProcess	0.1873
qgp:quality validatorTools	0.1839
qgp:quality optimizationWhenProblemFoundInTesting	0.1667
qgp:quality automatedUnitTesting	0.1647
qgp:quality manualTestResultVerification	0.1397
qgp:quality measureAndImproveDifficultFeatures	0.1364
qgp:quality smokeTesting	0.1156
qgp:quality userInterfaceTextConfigurationTool	0.1151
qgp:quality refactoring	0.1098
qgp:quality versionControlGuideline	0.1094
qgp:quality automaticCodeMetrics	0.1000
qgp:quality customerSurvey	0.0455
qgp:quality releaseScheduling	0.0440
qgp:quality smokeTestingInRealisticEnvironment	0.0312

Appendix III – Validation References

Database	Reference	Status
BPCH	Shull F., Basili V., Boehm B., Brown A.W., Costa P., Lindvall M., Port D., Rus I., Tesoriero R., Zelkowitz M., What We Have Learned About Fighting Defects, Proceedings of the Eight IEEE Symposium on Software Metrics (METRICS'02), IEEE, USA, 2002.	OK
BPCH	JPL D-18441, Report on the Loss of the Mars Climate Orbiter Mission, JPL Special Review Board, Jet Propulsion Laboratory, California Institute of Technology, USA, 1999.	Failed
BPCH	Rifkin S., Deimel L., Applying Program Comprehension Techniques to Improve Software Inspections, 19th Annual NASA Software Engineering Laboratory Workshop, USA, 1994.	OK
BPCH	Madachy R.J., Measuring Inspections at Litton, Software Quality, 2 (4), USA, 1996.	Missing
BPCH	Eagan M.E., Advances in Software Inspections, IEEE Transactions on Software Engineering, Vol. 12, Issue 7, USA, 1986.	Missing
BPCH	Basili V.R., Selby R.W., Comparing the Effectiveness of Software Testing Strategies, IEEE Transactions on Software Engineering, Vol SE-13, No. 12, Dec 1987, USA, 1987.	OK
BPCH	Grady R., van Slack T., Key Lesson in Achieving Widespread Inspection Use, Hewlett-Packard, IEEE Software, July 1994, p. 46-57, USA, 1994.	OK
BPCH	Russell G.W., Experience With Inspection in Ultralarge-Scale Developments, IEEE Software, Jan 1991, p. 25-31, USA, 1991.	OK
BPCH	Basili V.R., Green R., Laitenberger O., Lanubile F., Shull F., Sørumgård S., Zelkowitz M.V., The Empirical Investigation of Perspective-Based Reading, Empirical Software Engineering, USA, 1996.	OK
BPCH	Kolkhorst B.G., Space Shuttle Primary Onboard Software Development: Process Control and Defect Cause Analysis, Unpublished IBM Report, Houston, TX, USA.	Missing

BPCH	Kelly J.C., Sherif J.S., Hops J., An analysis of defect densities found during software inspections, Journal of Systems and Software, Vol. 17 (2), Feb 1992, p.111-117, USA, 1992.	Inaccessible
SEEDS	Petersen K., Rönkkö K., Wohlin C., The Impact of Time Controlled Reading on Software Inspection Effectiveness and Efficiency – A Controlled Experiment. ESEM'08, 2008.	OK
SEEDS	Kelly D., Shepard T., Task-Directed Software Inspection Technique: An Experiment and Case Study. Proceedings of the 2000 conference of the Centre for Advanced Studies on Collaborative research, ACM, USA, 2000.	OK
SEEDS	Thelin T., Team-based Fault Content Estimation in the Software Inspection Process. Proceedings of the 26th International Conference on Software Engineering (ICSE'04), IEEE, USA, 2004.	OK
SEEDS	Phongpaibul M., Boehm B., An Empirical Comparison Between Pair Development and Software Inspection in Thailand. ISESE'06, ACM, USA, 2006.	OK
SEEDS	Baker R.A., Code Reviews Enhance Software Quality, ICSE'97, ACM, USA, 1997.	OK
SEEDS	Porter A., Siy H., Toman C.A., Votta L.G., An Experiment to Assess the Cost-Benefits of Code Inspections in Large Scale Software Development. SIGSOFT'95, ACM, USA, 1995.	OK
SEEDS	Dunsmore A., Roper M., Wood M., Systematic Object-Oriented Inspection – An Empirical Study. IEEE, USA, 2001.	OK
SEEDS	Da Cunha A.D., Greathead D., Does Personality Matter? An Analysis of Code-Review Ability. Communications of the ACM, May 2007, Vol.50 (5), p.109-112, USA, 2007.	OK
SEEDS	Dunsmore A., Roper M., Wood M., Object-Oriented Inspection in the Face of Delocalisation. ICSE 2000, ACM, USA, 2000.	OK
SEEDS	Biffl S., Freimut B., Leitenberger O., Investigating the Cost-Effectiveness of Reinspections in Software Development. IEEE, USA, 2001.	OK
SEEDS	Tyran C.K., George J.F., Improving Software Inspections with Group Process Support. Communications of the ACM, Sep 2002, Vol.45 (9), p. 87-92, USA, 2002.	OK

SEEDS	Staron M., Kuniarz L., Thurn C., An Empirical Assessment of Using Stereotypes to Improve Reading Techniques in Software Inspections. 3-WoSQ'05, ACM, USA, 2005.	OK
SEEDS	Raz T., Yaung A.T., Inspection Effectiveness in Software Development: A Neural Network Approach. Proceedings of the 1994 conference of the Centre of Advanced Studies on Collaborative research, p.61, Canada, 1994.	OK
SEEDS	Trytten D.A., A Design for Team Peer Code Review, SIGCSE'05, ACM, USA, 2005.	OK

Appendix IV – EBSE Semantic Database OWL Schema

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:qgp="http://localhost/qgp.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
    xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:units="http://sweet.jpl.nasa.gov/ontology/units.owl#"
    xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://localhost/qgp.owl">
  <owl:Ontology rdf:about=""/>
  <rdfs:Class rdf:ID="Reliability">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdfs:Class>
  <owl:Class rdf:ID="Goal">
    <owl:disjointWith>
      <owl:Class rdf:ID="Reference"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:ID="Practice"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:ID="Result"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:ID="Context"/>
    </owl:disjointWith>
  </owl:Class>
  <owl:Class rdf:ID="GoalResult">
    <rdfs:subClassOf rdf:resource="#Goal"/>
  </owl:Class>
  <owl:Class rdf:about="#Context">
    <owl:disjointWith rdf:resource="#Goal"/>
    <owl:disjointWith>
      <owl:Class rdf:about="#Practice"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:about="#Result"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:about="#Reference"/>
    </owl:disjointWith>
  </owl:Class>
  <owl:Class rdf:ID="ToolPractice">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Practice"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Practice">
    <owl:disjointWith>
      <owl:Class rdf:about="#Reference"/>
    </owl:disjointWith>
    <owl:disjointWith rdf:resource="#Goal"/>
    <owl:disjointWith>
      <owl:Class rdf:about="#Result"/>
    </owl:disjointWith>
    <owl:disjointWith rdf:resource="#Context"/>
  </owl:Class>
  <owl:Class rdf:ID="ResultCollection">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Result"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Result">
```

```
    <owl:disjointWith rdf:resource="#Goal"/>
    <owl:disjointWith rdf:resource="#Practice"/>
    <owl:disjointWith rdf:resource="#Context"/>
    <owl:disjointWith>
      <owl:Class rdf:about="#Reference"/>
    </owl:disjointWith>
  </owl:Class>
  <owl:Class rdf:about="#Reference">
    <owl:disjointWith rdf:resource="#Goal"/>
    <owl:disjointWith rdf:resource="#Practice"/>
    <owl:disjointWith rdf:resource="#Result"/>
    <owl:disjointWith rdf:resource="#Context"/>
  </owl:Class>
  <owl:Class rdf:ID="QualityMatrixResult">
    <rdfs:subClassOf rdf:resource="#ResultCollection"/>
  </owl:Class>
  <owl:Class rdf:ID="QualityPractice">
    <rdfs:subClassOf rdf:resource="#Practice"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="QGPMatrix">
    <rdfs:domain rdf:resource="#Result"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="specializes">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty"/>
    <rdfs:range rdf:resource="#Practice"/>
    <rdfs:domain rdf:resource="#Practice"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasReference">
    <rdfs:domain rdf:resource="#Reference"/>
    <rdfs:range rdf:resource="#Reference"/>
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="isReferredBy"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasReferenceResult">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="isResultReference"/>
    </owl:inverseOf>
    <rdfs:domain rdf:resource="#Reference"/>
    <rdfs:range rdf:resource="#Result"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="isResultPractice">
    <rdfs:range rdf:resource="#Practice"/>
    <rdfs:domain rdf:resource="#Result"/>
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="hasPracticeResult"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="isGoalResult">
    <rdfs:range rdf:resource="#Result"/>
    <rdfs:domain rdf:resource="#GoalResult"/>
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="hasResultGoal"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasResultContext">
    <rdfs:range rdf:resource="#Context"/>
    <rdfs:domain rdf:resource="#Result"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasResultCollection">
    <rdfs:domain rdf:resource="#Result"/>
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="isCollectionOfResult"/>
    </owl:inverseOf>
    <rdfs:range rdf:resource="#ResultCollection"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="referenceValidity">
    <rdfs:domain rdf:resource="#Reference"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#isResultReference">
```

```
    <owl:inverseOf rdf:resource="#hasReferenceResult"/>
    <rdfs:domain rdf:resource="#Result"/>
    <rdfs:range rdf:resource="#Reference"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#isCollectionOfResult">
    <rdfs:domain rdf:resource="#ResultCollection"/>
    <rdfs:range rdf:resource="#Result"/>
    <owl:inverseOf rdf:resource="#hasResultCollection"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasSubContext">
    <rdfs:range rdf:resource="#Context"/>
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="isSubContextOf"/>
    </owl:inverseOf>
    <rdfs:domain rdf:resource="#Context"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#isSubContextOf">
    <rdfs:range rdf:resource="#Context"/>
    <rdfs:domain rdf:resource="#Context"/>
    <owl:inverseOf rdf:resource="#hasSubContext"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="referenceReliability">
    <rdfs:range rdf:resource="#Reliability"/>
    <rdfs:domain rdf:resource="#Reference"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#hasPracticeResult">
    <rdfs:range rdf:resource="#Result"/>
    <rdfs:domain rdf:resource="#Practice"/>
    <owl:inverseOf rdf:resource="#isResultPractice"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#hasResultGoal">
    <rdfs:domain rdf:resource="#Result"/>
    <owl:inverseOf rdf:resource="#isGoalResult"/>
    <rdfs:range rdf:resource="#GoalResult"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#isReferredBy">
    <owl:inverseOf rdf:resource="#hasReference"/>
    <rdfs:domain rdf:resource="#Reference"/>
    <rdfs:range rdf:resource="#Reference"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="resultEvidence">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Result"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="referenceAuthor">
    <rdfs:domain rdf:resource="#Reference"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="goalResultValue">
    <rdfs:domain rdf:resource="#GoalResult"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="practiceReference">
    <rdfs:domain rdf:resource="#Practice"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="contextName">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Context"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="referenceType">
    <rdfs:domain rdf:resource="#Reference"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="resultUnit">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:subPropertyOf rdf:resource="#resultEvidence"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="referenceURI">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Reference"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="practiceURI">
```

```
    <rdfs:domain rdf:resource="#Practice"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="practiceDescription">
    <rdfs:domain rdf:resource="#Practice"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  </owl:DatatypeProperty>
  <owl:TransitiveProperty rdf:ID="hasSubGoal">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:range rdf:resource="#Goal"/>
    <rdfs:domain rdf:resource="#Goal"/>
    <owl:inverseOf>
      <owl:TransitiveProperty rdf:ID="isSubGoalOf"/>
    </owl:inverseOf>
  </owl:TransitiveProperty>
  <owl:TransitiveProperty rdf:about="#isSubGoalOf">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:range rdf:resource="#Goal"/>
    <rdfs:domain rdf:resource="#Goal"/>
    <owl:inverseOf rdf:resource="#hasSubGoal"/>
  </owl:TransitiveProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.4, Build 533)  http://protege.stanford.edu -->
```