Faculdade de Engenharia da Universidade do Porto

Industrial Application of Process Performance Analysis Techniques

Ana Rita Oliveira e Silva

Mestrado em Engenharia de Software

Supervisor: Prof. Dr. João Pascoal Faria

Second Supervisor: Eng. Tiago Regala

June 25, 2018

Industrial Application of Process Performance Analysis Techniques

Ana Rita Oliveira e Silva

Mestrado em Engenharia de Software

Approved in oral examination by the committee:

Chair: Prof. Nuno Flores
External Examiner: Prof. Alberto Sampaio
Supervisor: Prof. João Pascoal Faria

June 25, 2018

Abstract

Recently, the ProcessPAIR method and tool for automatic performance analysis and improvement recommendation in software development processes was developed at FEUP. Based on a model specified by experts in the type of process being analyzed and automatically calibrated based on the data of many process users (individuals, teams or organizations), the ProcessPAIR tool is able to automatically identify and prioritize performance problems and respective causes for individual users of the process. The ProcessPAIR tool was instantiated and applied successfully in academic context for the Personal Software Process (PSP).

On the other hand, Critical Software (CSW) is a CMMI level 5 certified company with a considerable repository of project performance data with a significant level of size and detail and with an interest in applying automated process performance analysis techniques, based on quantitative methods.

Therefore, the main objective of this thesis is the identification of performance problems and their causes in software development processes, in an industrial context, demonstrating the industrial applicability of the ProcessPAIR method and tool and other relevant performance analysis techniques in Critical Software, through some sub-objectives like the development of a model of performance indicators (including top-level indicators and more detailed indicators) appropriate to the needs and data collected in CSW projects and its calibration.

To that end, two datasets provided by CSW were analyzed. The first dataset contained the values of four KPIs (Key Performance Indicators) collected over time (CPI - Cost Performance Index, SPI - Schedule Performance Index, Health and Audit Score), for a number of projects together with some project information such as the inclusion of a Software Product Assurance Engineer (SPAE). From this dataset, it was possible to conclude that the inclusion of a SPAE has a statistically significant positive impact over most of the KPIs with the exception of CPI, whose influence is more critic for this case. It was also possible to derive relevant ranges for classifying the values of each KPI into three semaphores (red, yellow and green).

The second dataset contained information related with code reviews performance (code review LOCs - Lines Of Code, defects, Defect Detection Rate (DDR), year, among others). From this dataset, it was possible to assume the adherence to log-normal distribution for DDR and the stability of the process. The comparison with the 2012 baseline also allowed the identification of a possible need for a new baseline.

The performance ranges and causal relationships derived from the datasets analyzed, were consolidated in a performance model that can be used in the future to predict the impact of including a SPAE in the project and to classify the values of KPIs into semaphores.

All the data analysis was carried out with the help of the SPSS tool.

Keywords: Audit Score, Analysis, CMMI, CPI, Health Score, ProcessPAIR, Performance, Process, SPI, SPAE, SPSS, KPI

Resumo

Recentemente, foi desenvolvido na FEUP o método e a ferramenta ProcessPAIR para análise automática de desempenho e recomendação de melhorias nos processos de desenvolvimento de software. Esta ferramenta tem como base um modelo especificado por especialistas no tipo de processo que está sob análise. A sua calibração de forma automática tem como base dados de muitos utilizadores do processo (indivíduos, equipas ou organizações). E assim, a ferramenta ProcessPAIR é capaz de identificar e priorizar automaticamente problemas de desempenho e respectivas causas para utilizadores individuais. A ferramenta ProcessPAIR foi instanciada e aplicada com sucesso no contexto académico para o "*Personal Software Process*" (PSP).

Por outro lado, a Critical Software (CSW) é uma empresa certificada CMMI nível 5 com um repositório considerável de dados de desempenho de projetos com um nível significativo de tamanho e detalhe e com interesse em aplicar técnicas de análise de desempenho de processo automatizadas, baseadas em métodos quantitativos.

Portanto, o objetivo principal desta tese é a identificação de problemas de desempenho e suas causas em processos de desenvolvimento de software, mas num contexto industrial, demonstrando a aplicabilidade industrial do método ProcessPAIR e da ferramenta na Critical Software. Isto é conseguido através de alguns sub-objetivos como o desenvolvimento de um modelo de indicadores de desempenho (incluindo indicadores de alto nível e indicadores mais detalhados) apropriados às necessidades e dados recolhidos nos projetos da CSW e a sua calibração.

De modo a atingir esse fim, dois conjuntos de dados fornecidos pela CSW foram analisados. O primeiro conjunto de dados continha valores de quatro KPIs (Indicadores de Performance Chave) recolhidos ao longo do tempo (*CPI* - Índice de Desempenho de Custos, *SPI* - Índice de Desempenho de Planeamento, *Health* e *Audit Score* - Pontuação de "Saúde" e Auditoria realizada aos projectos, respectivamente), para vários projetos em conjunto com algumas informações do projeto, como a inclusão de um Engenheiro de Garantia da Qualidade Produto de Software ("SPAE"). A partir deste conjunto de dados, foi possível concluir que a inclusão de um SPAE tem um impacto estatisticamente positivo e significativo sobre a maioria dos KPIs, com exceção do CPI, cuja influência é mais crítica para este caso.

Também foi possível derivar gamas de valores relevantes de forma a classificar os valores de cada KPI em três semáforos (vermelho, amarelo e verde).

O segundo conjunto de dados continha informações relacionadas com o desempenho de revisões de código (número de linhas de código revistas (LOCs - *Lines of Code*), defeitos, taxa de detecção de defeitos (DDR), ano, entre outros). A partir deste conjunto de dados foi possível assumir a adesão a uma distribuição log-normal para DDR e a estabilidade do processo. A comparação com a *baseline* de 2012 também permitiu a identificação de uma possível necessidade de uma nova *baseline*.

As gamas de desempenho e as relações causais derivadas dos conjuntos de dados analisados foram consolidadas num modelo de desempenho que pode ser usado no futuro para prever o impacto da inclusão de um SPAE no projeto e para classificar os valores de KPIs em semáforos.

Toda a análise de dados foi realizada com o auxílio da ferramenta SPSS.

Palavras-Chave: *Audit Score*, Análise, CMMI, *CPI*, DDR, *Health Score*, ProcessPAIR, Performance, Processo, *SPI*, SPAE, SPSS, *KPI*

Acknowledgements & Agradecimentos

The most difficult part is to acknowledge everyone I want and do not forget to mention someone that was part of this new journey. I will write my acknowledgments in Portuguese since it is my native language and it does not make sense to do it the other way.

Primeiro de tudo, gostaria de agradecer ao meu Professor e Orientador de Tese na Faculdade de Engenharia da Universidade do Porto (a casa que temporariamente me acolheu para mais um mestrado): Prof. João Pascoal. Um grande obrigada por ser uma pessoa de excelência e referência nesta instituição e por acolher o meu projecto. De seguida, quero referir os novos amigos que fiz nesta nova casa: obrigada a todo o pessoal do Mestrado em Engenharia de Software, não só meus colegas mas também os professores. Acho que aprendemos muito uns com os outros e foi um orgulho iniciar este projecto ambicioso que o MESW é.

Um grande obrigada vai também para o Eng. Tiago Regala, meu orientador nesta tese via Critical Software. Obrigada por não me deixares sair do eixos, ocupares o teu tempo livre a corrigir as minhas coisas e por estares ao meu lado desde que entrei nesta empresa.

O maior agradecimento é sem dúvida ao meu noivo, Pedro Lima. Sem ele não teria conseguido aguentar estes 2 anos "extra" de vida académica e conciliar tudo o resto à minha volta. Obrigada Pedro por confiares sempre nas minhas capacidades, limpares as minhas lágrimas quando elas me caiam nas bochechas em desespero, por festejares as pequenas vitórias que fui atingindo durante este percurso e por aceitares passares o resto da tua vida ao meu lado, mesmo sabendo das minhas loucuras!

A minha mãe, a minha heroína e "role-model" que nunca me deixa desistir e sabe do que sou capaz porque foi e será sempre o melhor exemplo que eu posso ter. Obrigada mãe! Não tenho palavras para descrever o que és para mim e quanto és importante na minha vida.

Aos meus amigos do ISEP, por serem o meu refúgio desde 2009. Às minhas amigas do coração, que estão sempre comigo e me ajudam a crescer e a alcançar sempre mais. Um grande obrigada por fazerem parte da minha vida desde que me lembro!!

Não posso deixar de agradecer a todo o pessoal da CSW que me foi acompanhando neste processo de ser trabalhador-estudante a tempo inteiro. Obrigada por toda a paciência que tiveram comigo, em especial a algumas meninas que me ajudaram a desanuviar sempre que precisava (elas sabem quem são).

Obrigada a todos por fazerem parte da minha vida! *Thank you all for being part of my day-by-day life!*

Por último, gostaria de referir que mesmo pessoas que não mencionei acima mas contribuiram (ainda que indirectamente) para mais um sorriso e/ou umas palavras de apreço foram importantes para que não vacilasse em mais uma aventura. E mesmo aqueles que torceram o nariz por achar que não conseguia levar mais uma de mil coisas até ao fim: obrigada! Fizeram-me chegar mais longe e superar-me dia após dia, mesmo perante muito cansaço.

Valeu cada noite mal dormida, cada lágrima, cada berro.

Ana Rita Oliveira e Silva

"Be the change that you want to see in the world."

Mahatma Gandhi

Contents

List of Figures

List of Tables

Abbreviations

BU	Business Unit
CAR	Causal Analysis and Resolution
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CPI	Cost Performance Index
COPQ	Costs Of Poor Quality
CSW	Critical Software
DCA	Defect Causal Analysis
DDR	Defect Detection Rate
ES	Effect Size
FD	Fishbone Diagram
FMEA	Failure Mode and Effect Analysis
IMS	Integrated Management System
IPPD	Integrated Product and Process Development
IR	Information Retrieval
IQR	interquartile range
IS	Information Systems
KPI	Key Performance Indicator
LCL	Lower Control Limit
LDA	Latent Dirichlet allocation
LOC	Lines of Code
LSI	Latent Semantic Indexing
MA	Measurement and Analysis
MSR	Mining Software Repositories
NC	Non-Conformance
OID	Organizational Innovation and Deployment
OPP	Organizational Process Performance
PA	Process Area
PI	Performance Indicator
PM	Performance Model
PMS	Performance Measurement System
PMIS	Project Management Information System
PPM	Parts Per Million
PSP	Personal Software Process
QMS	Quality Management System
QPM	Quantitative Project Management
RCA	Root Cause Analysis

RDIMS	Research Development and Innovation Management System
ROI	Return On Investment
RQ	Research Question
SE	Systems Engineering
SG	Specific-Goal
SP	Specific-Practice
SPAE	Software Product Assurance Engineer
SPC	Statistical Process Control
SEI	Software Engineering Institute
SME	Small and Medium Enterprise
SPI	Schedule Performance Index
SPI	Software Process Improvement
SS	Supplier Sourcing
SW	Software
TSP	Team Software Process
UCL	Upper Control Limit
VCS	Version Control System
WWW	*World Wide Web*

Chapter 1

Introduction

1.1 Motivation

In 1968, a NATO Software Engineering Conference in Germany initiated the concept of software engineering, identifying the problems with producing large, high-quality software applications [32], [69].

Software Engineering is a relevant field that provides a set of methods, tools and procedures for the development of quality software within the constraints of time and cost [21]. Quality Engineering is about getting results of the required quality within the schedule and budget. This often involves making compromises — engineers cannot be perfectionists [78].

The past years have seen an increasing interest in methods, models, techniques and frameworks to support the assessment and improvement of software development processes. Here, software process improvement (SPI) encompasses process assessment, process refinement (traditional SPI), and process innovation (introducing major process changes). National and international bodies have proposed and supported frameworks such as the Quality Improvement Paradigm [22].

The rate of software development project failure has been routinely documented in information systems (IS) research. The Standish Group research report [44] shows a staggering 31.1% of projects will be canceled before they ever get completed. Further results indicate 52.7% of projects will cost 189% of their original estimates. The cost of these failures and overruns are just the tip of the proverbial iceberg. The lost opportunity costs are not measurable, but could easily be in the trillions of dollars [89].

The report [44], from 2009, shows that software projects now have a 32% success rate compared to 35% from the previous study in 2006 and 16% in 1994. On the other hand, 44% of projects were challenged, while 24% failed, as can be seen in the chart below, Figure 1.1.

Critical Software (CSW), founded in 1998, is a specialist in development of software solutions and information engineering services to support mission critical, safety-critical and business-critical systems. The assurance that critical business processes are performed with the highest

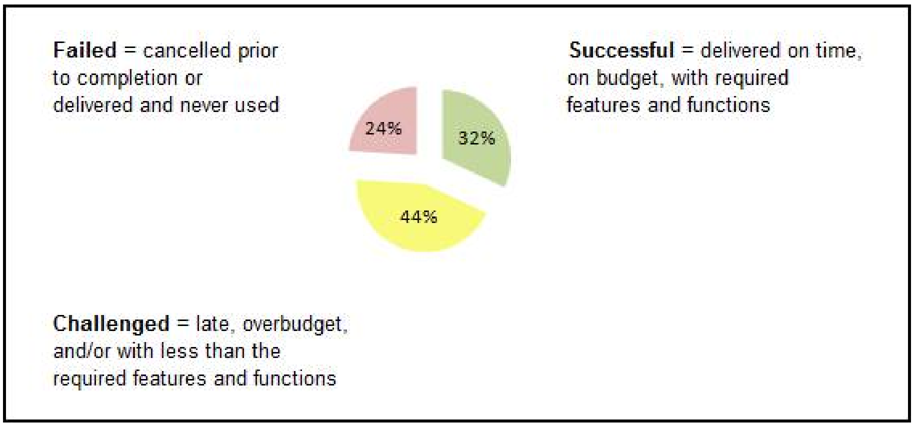


Figure 1.1: Pie Chart showing evidence percentages of "why IT projects fail".

quality standards for software security, performance and reliability is the major concern and commitment to the costumers.

This company already fulfilled CMMI (Capability Maturity Model Integration) Level 5, the highest level of this capability maturity model, one of the most prestigious and demanding reference benchmarks in the world for maturity and organizational process performance.

Due to their engagement with Quality standards, improvements and continuous level of optimization of performance, an opportunity had taken place in this Master Thesis to ally CSW to the ProcessPAIR method and tool. This is a tool for automated model-based analysis of performance data produced in the context of high-maturity software processes, like CSW, and is currently able to analyze performance data collected by PSP and TSP users.

The motivation is to adapt these concepts to a set of data of the company and to analyze its results in order to have an impact in their future.

1.2 Context

This dissertation work was conducted in the context of a collaboration between Critical Software (CSW) and the Faculty of Engineering of the University of Porto (FEUP), in the scope of the Master in Software Engineering.

Founded twenty years ago, CRITICAL Software is specialized in development of software solutions and information engineering services to support mission-critical, safety-critical and business-critical systems. CSW is a leader company with strong focus in software Quality, both from the product perspective and the process perspective. It was the first company to be certified at CMMI level 5 in Portugal, a huge achievement due to their high-quality demanding markets, such as aerospace and defense.

The Software Engineering research group at FEUP has focused its research work on software quality in three perspectives: product quality assessment and improvement (test automation), process quality assessment and improvement, and quality by construction [31].

In this Master Thesis, performance analysis and recommendation assumes a role of great importance because of metrics that are being collected relative to project management and process improvements (actions to be taken, identification of problems and related causes).

A tool, ProcessPAIR - Automated Software Process Performance Analysis and Improvement Recommendation, has already been developed (during a PhD in Computer Science [70]) in order to assist in this analysis and to automate it. This tool was applied in Mexico in an academic context and the results were promising.

ProcessPAIR is only able (until nowadays) to analyze data produced in the context of high-maturity software processes, such as the Personal Software Process (PSP)/Team Software Process (TSP).

Presently, the focus of this Master Thesis concerns with the adaptation and application of the tool and related techniques in an industrial context, which will be possible with the partnership with Critical Software, a company with high-maturity processes and methods.

1.3 Objectives

The major objective of this dissertation work is to apply high-maturity performance analysis techniques, such as the ones embodied in ProcessPAIR, in an industrial context (Critical Software).

In this way, we intend to arrive at results that are considered relevant for the interested parties.

Such high-maturity techniques involve two fundamental steps:

- **Definition** and **selection** of relevant and feasible top-level performance indicators, as well as more detailed indicators and factors that affect the top-level indicators, to support causal analysis;

- **Calibration** of ranges of values for the classification of individual indicators (in "semaphores"), as well as calibration of other predictive models for cause-and-effect relationships between indicators and factors (based on historical datasets, business objectives, industry benchmarks, literature, among others).

Hence, the specific objectives of this work are to perform these steps for CSW. Moreover, the possible adaption of the ProcessPAIR tool for semi-automated performance analysis in CSW, based on results of the above steps, will also be investigated.

1.4 Document Structure

This Master Thesis is organized in the following chapters:

In Chapter 1 motivation for this project development, objectives settled and Critical Software background are presented.

Next Chapter 2, entitled "State of the Art", provides some background and state of the art analysis of relevant topics within the scope of this thesis: performance analysis, CMMI high-maturity levels, automated performance analysis with ProcessPAIR, Statistical Process Control techniques, Mining Software Repositories.

Hereinafter, Chapter 3 refers some of Critical Software Processes within this area and their LeanQMS, a major initiative that brought vast benefits to the company.

Chapters 4 and 5 exhibit the analysis performed in this context for one of two datasets provided by CSW. Starting with an evaluation of SPSS choice instead of R and Microsoft Excel tool (the last one most commonly used), passing through the description of the datasets provided by Critical Software. All the data preparation carried out is referred as well as descriptive statistics of each dataset and relevant graphics associated. Hypothesis testing (Mann Whitney U Test and ANOVA) for first dataset is presented in detail in this chapter as well.

In Chapter 5, a code review evaluation related with second dataset is performed with a baseline comparison and control chart. In addition, the data preparation and descriptive statistics of this dataset is mentioned, similar to previous Chapter.

Finally, in Chapter 6, all the work is summarized, highlighting goals achieved and ideas for future work.

Chapter 2

Background and State of the Art

Software Industry has become nowadays a very significant economic activity with a huge growth, turning up a major economic force throughout many nations in the world. The software industry in most countries has an industrial backcloth, made up mainly of small and medium software enterprises (SMEs) [32].

Within the broad scope of the software industry, this project focuses primarily on areas such as software engineering practices and software process improvement.

Software Process is an important knowledge area that supports software development [21].

Most of software development projects face similar problems, transversal to all areas, for example: delays, overrun budget and/or customers' dissatisfaction with the quality of the software delivered. This phenomenon is so widespread that it is even being named as a software crisis [65].

The implementation of software process improvements has an important role as a way to overcome this interdisciplinary problems [33]. Process improvement projects should be planned taking into account the company as a whole. Given that the distinctive factor for organizations is achieved not only with the implementation of a specific improvement but with the ability to continuously improve processes and products, achieving the most appropriate results in the shortest possible time and at the lowest possible cost.

The Software Engineering Institute (SEI) has performed a research to help organizations to develop and maintain quality products and services. In their research, they came to the conclusion that an organization should focus on several dimensions in order to improve its business.

Figure 2.1 illustrates the three critical dimensions that organizations typically focus on: people, procedures & methods, and tools & equipment.

This leads to a reasonable question: "What holds everything together?", and the answer would be "the processes used in the organization", because processes enable the alignment of the business. Processes allow to address scalability and provide a way to incorporate knowledge of how to do things better and to leverage the resources, as well as to examine business trends [34].

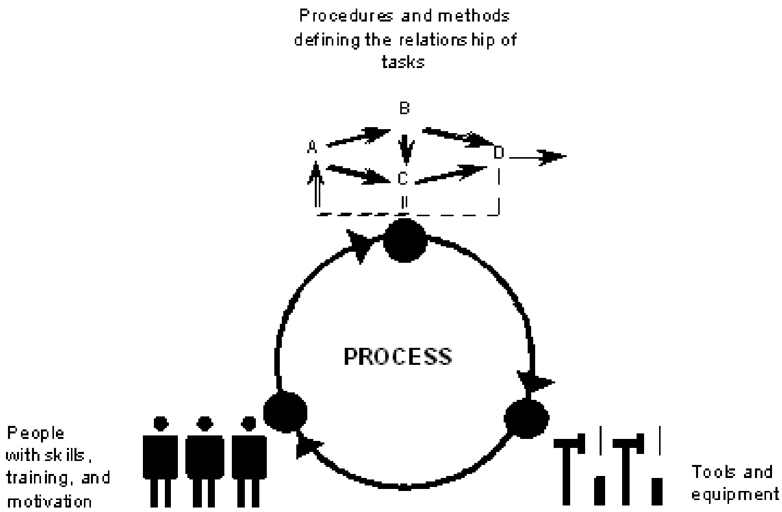


Figure 2.1: The Three Critical Dimensions.

To strengthen several types of organizations, efficient Software Engineering practices are needed and adapted to the size and type of business. Over the last two decades, the Software Engineering community has expressed special interest in Software Process Improvement (SPI) in an effort to increase software product quality, as well as the productivity of software development [68].

The implementation of improvements in the software process is a highly complex activity that requires an intense knowledge [25]. This involves, at least, a general knowledge of software engineering and an ability to use it to guide the implementation of improvements with the aim to increase the chances to achieve the expected results.

Hence, the automation of the processes and visualization of the results is an important aspect in this area. The next section 2.1 enlightens the performance analysis area, an important subject to understand how improvement of organizations can be achieved.

2.1 Performance Analysis in CMMI

Organizations are constantly seeking performance improvement through new technology, processes, and instruments. Probably the most important objective of performance measurement is to replace intuition by facts in order to get an organized plan and scheme of overall organization to improve their performance in a well-defined way.

Today, most businesses are exposed to intensive competition and, therefore, companies are forced to improve their performance steadily. Companies doing business in a competitive environment must measure their performance regularly to quantitatively assess whether the set goals are met and to be able to measure their own performance against the performance of other companies.

In addition, they should measure different facets of performance (e.g. customer retention, service quality, employee motivation, sales revenue) to better understand the interrelationships between business-relevant aspects. This generated knowledge may be used to initiate appropriate action to improve overall business performance [94].

It is almost certain that process evaluation techniques benefit software development organizations as some studies have shown [22], however, evaluating whether an assessment and consequent improvement plan have had a causal, significant, and positive impact on company success is difficult.

"Maturity" can be defined as a measure to evaluate the capabilities of an organization in regards to a certain subject. A well-known maturity model for improving the software development process is the Capability Maturity Model (CMM).

Therefore, Capability Maturity Models (CMMs) have been developed for organizations that use specialized bodies of knowledge to improve their internal process maturity, as well as to evaluate their processes [93].

Maturity models are valuable instruments because they allow the assessment of the current situation of a company as well as the identification of reasonable improvement measures. Equally, they have proved [8] to be crucial for companies in order to enhance their strategical positioning and to help find better solutions for change. Over the last few years, over a hundred maturity models have been developed to support IT management [8].

Software built under CMM guidelines has an enormous impact on the software community. According to some authors from Carnegie Mellon University [64] it is estimated that billions of dollars are spent world-wide on CMM-based improvement, having impact on productivity, cycle time, and quality of the organizations [29].

Process followed by a company is proportional to the complexity and organization of their CMMI (Capability Maturity Model Integration) level implemented, so following subsection 2.1.1 outlines capability maturity models topic, giving an overview of its focus and why it is currently adopted by companies.

2.1.1 CMMI Overview

In 2001, the Software Engineering Institute (SEI) released the successor of CMM: "CMMI", acronym for "Capability Maturity Model Integration" models [19] which are collections of best practices that help organizations to improve their processes. They are used to assess the organization's situation, deriving and prioritizing improvement measures, and eventually monitoring the progress of its implementation over time, if necessary [7]. These models are developed by product teams with members from industry, government and the SEI [19] at Carnegie Mellon University. In fact, CMM (Capability Maturity Model) has become quite popular since it was proposed by SEI [64] in September 1987 with the release of the process maturity framework and a maturity questionnaire.

Four models have been integrated together into the Software Engineering Institute's CMMI: Systems Engineering (SE), Software (SW) Engineering, Integrated Product and Process Development (IPPD) and Supplier Sourcing (SS).

In sum, CMMs have been developed for organizations that use specialized bodies of knowledge to improve their internal process maturity, as well as to evaluate their processes [93]. They provide the criteria and characteristics that need to be met to achieve a certain level of maturity and therefore the basis for the maturity assessment. Based on the results of an analysis to the organization (and its processes), recommendations for improvement measures can be derived and prioritized to ultimately achieve higher maturity levels if possible [13].

CMM introduced the concept of five maturity levels (demonstrated in Figure 2.2 below) measured by the achievement of the goals that apply to a set of process areas [13].

Figure 2.2: CMMI: Five Maturity Levels [54].

CMMI has two process improvement paths: continuous or staged representation. In the continuous representation of a CMMI model, Capability Models are used to assess the CMMI process area(s). Within each process area there are specific and generic goals that should be implemented by specific and generic practices, respectively. As an example, to satisfy Capability Level 2 for a process area, the specific goals and practices of Level 2 for that process area must be satisfied, as well as the Level 2 generic goals for that same process area. The staged representation uses Maturity Levels to asses a set of related processes across multiple process areas, including the 16 core process areas (PAs). The 16 Core PAs are common to all three CMMI Models (CMMI Common Framework) as can be seen in Figure 2.3. Within each maturity level there are process areas that contain goals, common features, and practices [76].

Although a classic staged maturity framework with 5 levels was first introduced by W. S. Humphrey back in 1988, its basic ideas remain essentially unchanged and they are still employed in CMMI and other maturity models [65]. CMMI also makes the distinction between areas, called constellations. There are 3 constellations in total, as can be seen in Figure 2.3.

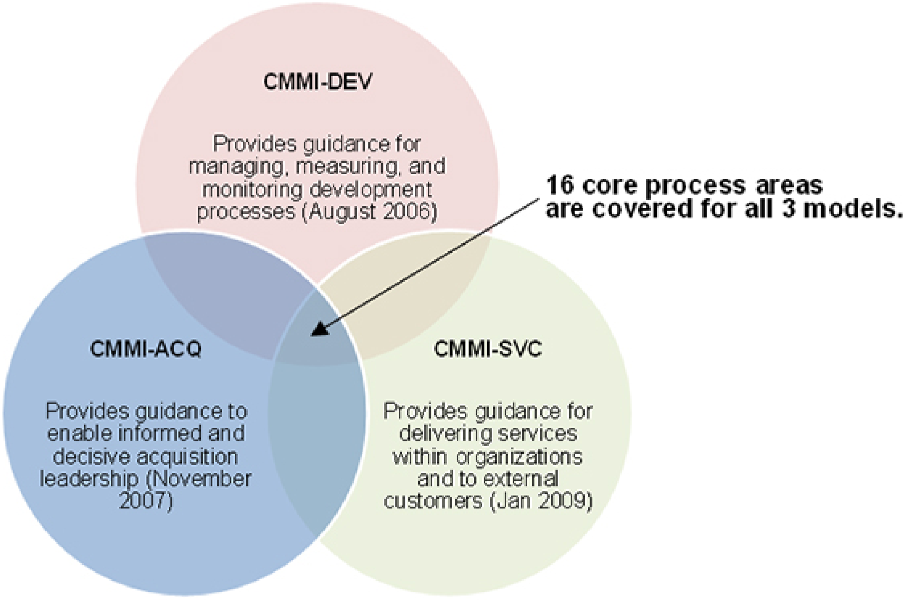


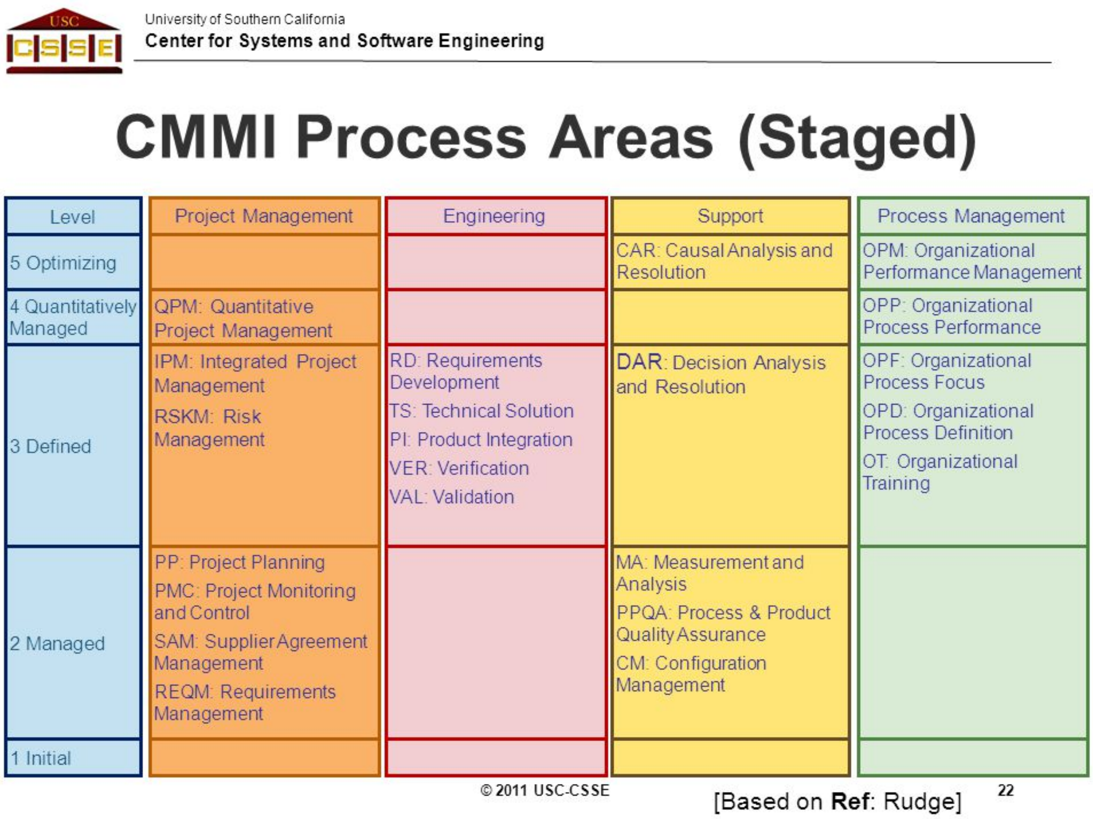
Figure 2.3: The Three Constellations of CMMI [90]: CMMI-DEV, CMMI-ACQ and CMMI-SVC.

The three models/constellations of CMMI are: for development organizations (CMMI-DEV), acquisition organizations (CMMI-ACQ), and service-type organizations (CMMI-SVC). CMMI-DEV has a total of 22 Process Areas and 5 specific PAs; CMMI-ACQ has 22 PAs with 6 specific PAs, and finally, CMMI-SVC has a total of 24 PAs, 7 specific. A brief explanation to Process Areas is made clear further ahead.

Summing up, the CMMI Model is very complex and several aspects need to be considered in the process improvement initiatives, such as: allocation of resources, selection of processes to be analyzed and improved, selection of pilot project or projects, choice of models to be used and approach taken to the initiative [25]. Nevertheless, the principal model components can be synthesized as follows:

1. **Process Areas**
   Key process areas identify the issues that must be addressed to achieve a certain maturity level (Figure 2.4). A set of process areas are related to software process activities: the CMMI identifies 16 core process areas that are relevant to software process capability and improvement, and more according to each constellation. These, in turn, are organized into four groups in the continuous CMMI model [78], [37].

2. **Goals**
   The CMMI establishes a number of goals (each maturity level comprises a set of goals), which are abstract descriptions of a desirable state that should be attained by an organization and can be specific or even generic goals.


University of Southern California
Center for Systems and Software Engineering

CMMI Process Areas (Staged)

Level	Project Management	Engineering	Support	Process Management
5 Optimizing			CAR: Causal Analysis and Resolution	OPM: Organizational Performance Management
4 Quantitatively Managed	QPM: Quantitative Project Management			OPP: Organizational Process Performance
3 Defined	IPM: Integrated Project Management RSKM: Risk Management	RD: Requirements Development TS: Technical Solution PI: Product Integration VER: Verification VAL: Validation	DAR: Decision Analysis and Resolution	OPF: Organizational Process Focus OPD: Organizational Process Definition OT: Organizational Training
2 Managed	PP: Project Planning PMC: Project Monitoring and Control SAM: Supplier Agreement Management REQM: Requirements Management		MA: Measurement and Analysis PPQA: Process & Product Quality Assurance CM: Configuration Management	
1 Initial				

© 2011 USC-CSSE [Based on **Ref**: Rudge] 22

Figure 2.4: CMMI-DEV Process Areas [50].

Specific goals are associated with each process area and define the desirable state for that area [78]. On the other hand, generic goals are associated with the institutionalization of good practice.

3. **Practices**
   A set of good practices are descriptions of ways of achieving a goal. Several specific and generic practices may be associated with each goal within a process area [78].

To recap points above: goals of a key process area must be achieved for the organization to satisfy a key process area. When these goals are accomplished on a continuing basis across projects, the organization can be said to have institutionalized the process capability characterized by the key process area [63].

CMMI structure can be decomposed taking into account each maturity level, which is composed by several key process areas. Each key process area is organized into five sections called common features. The common features specify the key practices that, when collectively addressed, accomplish the goals of the key process area.

Except for Level 1, as can be seen in Figure 2.4 each maturity level is decomposed into several key process areas that indicate the areas an organization should focus on to improve its software process [63]. This Figure refers to Process Areas of a continuous CMMI representation (4 categories of process areas - Process Management, Project Management, Engineering and Support).

The following subsection 2.1.2 briefly clarifies the differences between CMMI Level 4 and 5. These CMMI levels are having a high focus in this project, because the company with which the partnership was made is CMMI level 5 and hence the relevance of this subject.

2.1.2 CMMI Maturity Levels 4 and 5

In section 2.1.1 an overview of CMMI and its maturity levels was given. This subsection only highlights levels 4 and 5 of this maturity model by virtue of its relevance to this Master Thesis.

In a summarized form, the difference between level 4 and level 5 is the main objective of each one. In level 4 it can be said that the focus is more related with quantitative management and quality control, and at level 5, the scope, and consequently, the efforts, comprises continuous improvement, technological innovations and defect prevention [19].

The next Table 2.1 shows this difference between levels and each CMMI representation: for "Capability Level", in a continuous representation there are six capability levels [0-5]; "Maturity Level", in a staged representation there are five maturity levels, numbered 1 through 5.

Table 2.1: Capability and Maturity Levels.

Level	**Capability Levels**	**Maturity Levels**
Level 0	Not Performed	N/A
Level 1	Performed	Performed
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4	Quantitatively Managed	Quantitatively Managed
Level 5	Optimizing	Optimizing

Levels two and three mainly focus on the way to perform the process and enhance the process capability individually; while levels four and five offer a more strategic focus which aligns the quality and process performance objectives with business objectives [39].

Thereby, as an organization progresses through the different levels of CMMI, the quality of its development tends to improve as well as the overhead of the development process, adapting quickly to customers or partners who change their needs. In addition, agile practices allow rapid adaptation and early delivery of commercial value, reason why are often established by companies [87].

2.2 Automated Performance Analysis with ProcessPAIR

High-maturity software development processes can generate significant amounts of data that can be periodically analyzed to identify performance problems, determine their root causes and devise improvement actions. However, conducting that analysis manually is challenging because of the potentially large amount of data to analyze, the effort and expertise required and the lack of benchmarks for comparison.

ProcessPAIR is a novel method tool designed to help developers analyze their performance data with less effort [70], by automatically identifying and ranking performance problems and potential root causes, so that subsequent manual analysis for the identification of deeper causes and improvement actions can be properly focused. The analysis is based on performance models defined manually by process experts and calibrated automatically from the performance data of many developers [72]. This analysis is, subsequently, able to assist all project management and quality needed in projects.

Initial screen of this tool is represented in Figure 2.5.

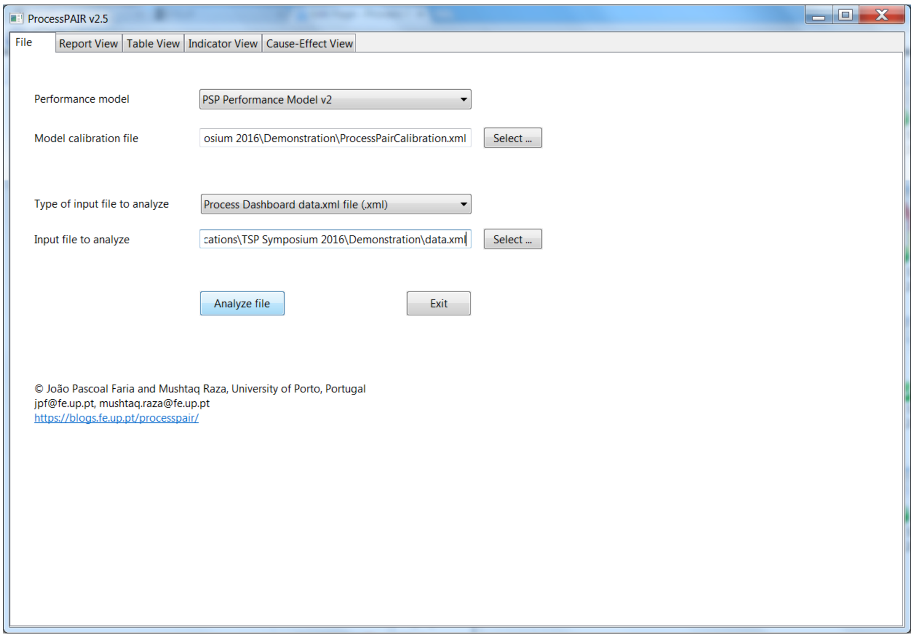


Figure 2.5: Initial Screen of ProcessPAIR Tool for files upload (File View) [71]

Three main steps were devised in the ProcessPAIR method:

1. **Definition**: Process experts define the structure of a performance model (PM) suited for the development process under consideration (TSP, PSP, or other). In this approach, a PM comprises a set of performance indicators (PIs) organized hierarchically by cause-effect relationships.

2. **Calibration**: The PM is automatically calibrated based on the performance data of many process users. The statistical distribution of each PI and statistical relations between PIs are computed from the data set.

3. **Analysis**: Once a PM is defined and calibrated, the performance data of individual subjects can be automatically analyzed with ProcessPAIR, to identify and rank performance problems and root causes [73].

Three colors are used in this tool: red - clear performance problem, yellow - potential performance problem, and green - no performance problem.

The results of the analysis are presented in multiple views:

- **Report View**: in text format (Figure 2.6), only to simply indicate, by priority, the most relevant top-level performance problems and potential root causes. Can be displayed for each project and with only leaf causes or not.

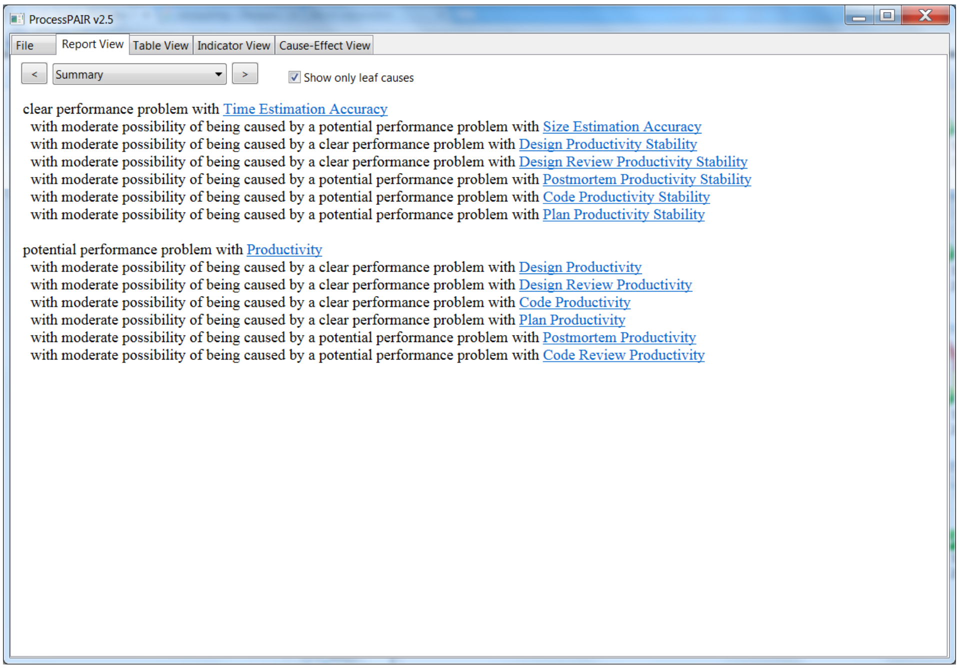
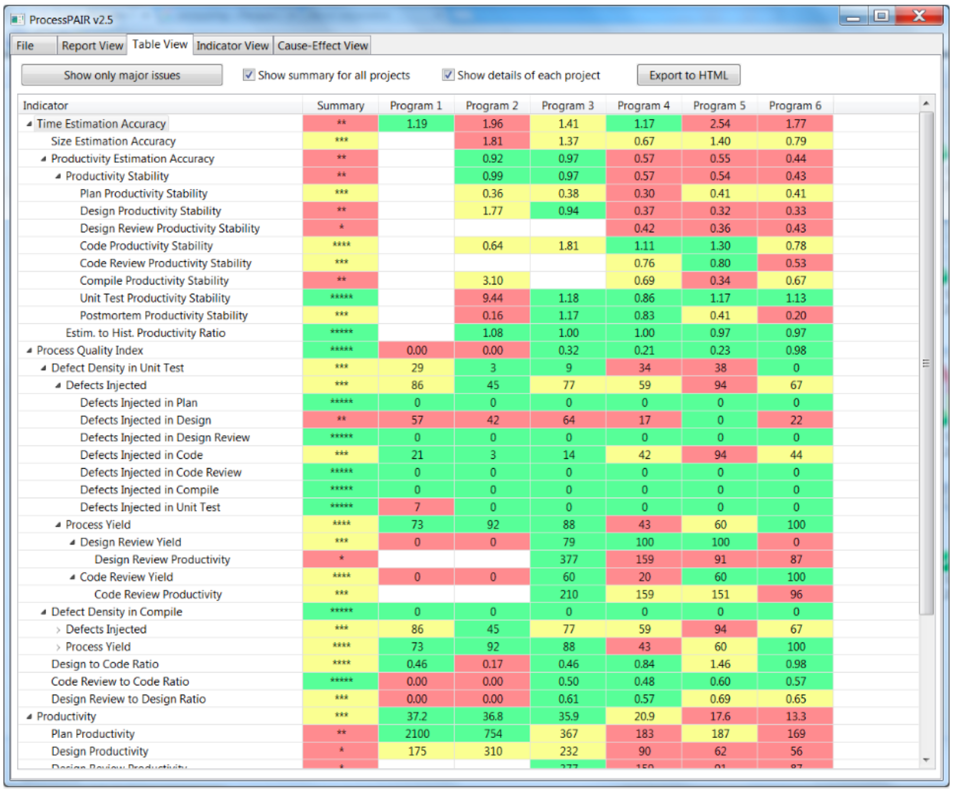


Figure 2.6: Report View of ProcessPAIR Tool with summary of the most relevant top-level performance problems and their causes [71].

- **Table View**: a colorful table that shows the values of the three top-level indicators (Time Estimation, Process Quality Index and Productivity) defined in the model (can be expanded) for the projects described in the input file, as well as summarized performance information. The default appearance of the table is by top-level performance indicators. The view can be again customized and exported to HTML. In Figure 2.7 can be seen that the indicators column is fully expanded. Each cell is colored green, yellow or red, in case its value suggests no performance problem, a potential performance problem, or a clear performance problem, respectively. This way, the Table View helps quickly identifying performance problems.

  The performance indicators (PIs) are organized hierarchically starting from top-level indicators and descending to lower level (child) indicators that affect the higher level ones according to a formula or statistical evidence.

- **Indicator View**: useful to provide all the information related with each performance indicator in a summary way: description, units, optimal value, recommended performance ranges



Indicator	Summary	Program 1	Program 2	Program 3	Program 4	Program 5	Program 6
Time Estimation Accuracy	**	1.19	1.96	1.41	1.17	2.54	1.77
Size Estimation Accuracy	***		1.81	1.37	0.67	1.40	0.79
Productivity Estimation Accuracy	**		0.92	0.97	0.57	0.55	0.44
Productivity Stability	**		0.99	0.97	0.57	0.54	0.43
Plan Productivity Stability	***		0.36	0.38	0.30	0.41	0.41
Design Productivity Stability	**		1.77	0.94	0.37	0.32	0.33
Design Review Productivity Stability	*				0.42	0.36	0.43
Code Productivity Stability	****		0.64	1.81	1.11	1.30	0.78
Code Review Productivity Stability	***				0.76	0.80	0.53
Compile Productivity Stability	**		3.10		0.69	0.34	0.67
Unit Test Productivity Stability	*****		9.44	1.18	0.86	1.17	1.13
Postmortem Productivity Stability	***		0.16	1.17	0.83	0.41	0.20
Estim. to Hist. Productivity Ratio	*****		1.08	1.00	1.00	0.97	0.97
Process Quality Index	*****	0.00	0.00	0.32	0.21	0.23	0.98
Defect Density in Unit Test	***	29	3	9	34	38	0
Defects Injected	***	86	45	77	59	94	67
Defects Injected in Plan	*****	0	0	0	0	0	0
Defects Injected in Design	**	57	42	64	17	0	22
Defects Injected in Design Review	*****	0	0	0	0	0	0
Defects Injected in Code	***	21	3	14	42	94	44
Defects Injected in Code Review	*****	0	0	0	0	0	0
Defects Injected in Compile	*****	0	0	0	0	0	0
Defects Injected in Unit Test	*****	7	0	0	0	0	0
Process Yield	****	73	92	88	43	60	100
Design Review Yield	***	0	0	79	100	100	0
Design Review Productivity	*			377	159	91	87
Code Review Yield	****	0	0	60	20	60	100
Code Review Productivity	***			210	159	151	96
Defect Density in Compile	*****	0	0	0	0	0	0
Defects Injected	***	86	45	77	59	94	67
Process Yield	****	73	92	88	43	60	100
Design to Code Ratio	****	0.46	0.17	0.46	0.84	1.46	0.98
Code Review to Code Ratio	*****	0.00	0.00	0.50	0.48	0.60	0.57
Design Review to Design Ratio	***	0.00	0.00	0.61	0.57	0.69	0.65
Productivity	***	37.2	36.8	35.9	20.9	17.6	13.3
Plan Productivity	**	2100	754	367	183	187	169
Design Productivity	*	175	310	232	90	62	56

Figure 2.7: Table View expanded of ProcessPAIR Tool.

and statistical distribution in the model. A chart is presented for example with defect density in unit test, as demonstrated in Figure 2.8.

In the bottom left, it is presented the statistical distribution of the values of the performance indicator in the PSP data set used for calibrating the model. The colors correspond to the performance ranges.

It is also possible to choose two or more indicators to analyze and determine their correlation coefficient.

- **Cause-Effect View**: this particular view is very helpful to identify and prioritize root causes of performance problems (Figure 2.9), so that improvement actions can be properly applied. Can be analyzed project by project or overall.

  It is presented a ranking coefficient that relates by T-shirt sizes (Figure 2.9) the cost of improving the value of the child indicator with the benefit on the value of the parent indicator (cost-benefit estimation). The coefficient can also be shown in a numerical form.

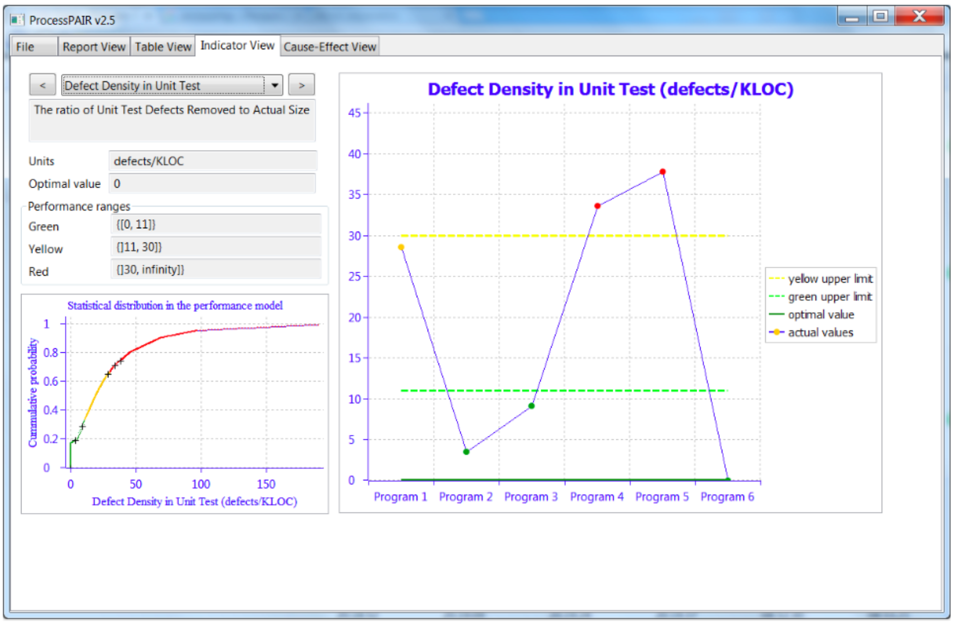


Figure 2.8: Indicator View of ProcessPAIR Tool.

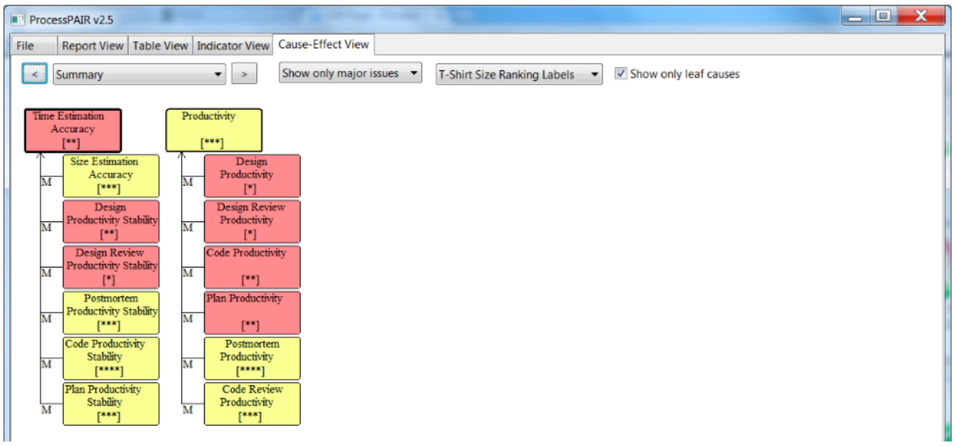


Figure 2.9: Cause-Effect View of ProcessPAIR Tool.

2.3 Performance Measurement and Analysis Techniques

2.3.1 Performance Measurement

One instrument that received considerable attention during the last years is called Performance Measurement System (PMS). A PMS tracks actual performance of an organization, helps identifying weaknesses, and supports communication and decision-making processes [94].

Software process improvement (SPI) is largely concerned with improving a software project's capability to develop high-quality software based on the requirements of customers or end users [89]. SPI is a widely recognized approach that software companies implement to improve quality, productivity, and time-to-market.

Many companies consider software process improvement a strategic issue, due to the fact that a failed process leads to failed software [46].

Project performance includes software engineering issues of efficiency and effectiveness, as well as organizational issues of control, communication, and organizational knowledge. Efficiency is often considered to be measured by the quality of the software product, adherence to budgeted time and money, and cost of the software operation, while effectiveness is considered to be the applicability and adaptability of the software [89].

2.3.2 Statistical Process Control

Statistical Process Control (SPC) is not really about statistics or control, it is about competitiveness [1]. Organizations, whatever their nature, compete on three issues: quality, delivery and price.

A well known approach used in several businesses is Six Sigma. Six Sigma is a business improvement approach that seeks to find and eliminate causes of mistakes or defects in business processes by focusing on outputs that are of critical importance to customers. As a result process performance is enhanced, customer satisfaction is improved and the bottom line is impacted through cost savings and increased revenue. Six Sigma is a strategic approach that works across all processes, products, company functions and industries [77].

Six Sigma is a powerful approach to process improvement, reduced costs and increased business profitability and revenue growth. Six Sigma has both management and technical components. The focus of management component is to select the right people for Six Sigma projects, select the right process metrics, provide resources for Six Sigma training, provide clear direction and guidance with regard to project selection, among others. The focus of the technical component is on process improvement by reducing variation, creating data which explains process variation, using statistical tools and techniques for problem solving, etc.[6].

Some results analysis of Six Sigma in the software industry: one survey was delivered by some authors [6] where the companies which participated have completed between 5 and 10 Six Sigma projects and are making bottom-line savings per project of over £100K on average. They refer that it was very interesting to observe that only 18 per cent of companies are measuring the costs of poor quality (COPQ). The average COPQ across the 10 companies was over 25 percent. This clearly indicates that many software companies still focus on getting their products out of the door without paying much attention to their core business processes in place. Most companies which are applying Six Sigma principles indicated that the Sigma capability level varies between 2.54 (150,000 defects per million opportunities) and 4 (6,210 defects per million opportunities) [6].

Six Sigma also provides a process measurement scale. The methodology utilizes “process sigma” as a measure of process capability with a 6-sigma process having a defect level of 3.4 ppm (Parts per million) and a 3-sigma process having a defect level of 66.807 ppm [77].

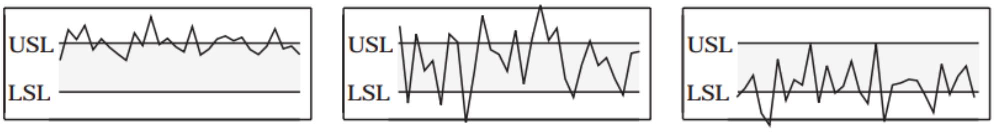
2.3.2.1 Techniques to identify problems and root causes

These techniques are the most commonly used and most effective tools for overcoming problems in terms of quality.

The main tools used for identifying quality problems include: check sheet, cause-and-effect diagram, control chart, Pareto chart, histogram, flow chart and scatter diagram.

Cause-and-effect diagrams, also known as fishbone diagrams, depict the causes of individual events and the connection between the events. These diagrams are used to pinpoint the cause of a specific quality problem in order to fix the issue. It is common for these types of diagrams to be constructed by several team members in order for discussion to take place on the legitimacy of the causes.

Control charts are made up of a horizontal line that represents the norm of the process. There is also a line above and below, to represent upper and lower control boundaries or even specification limits can be represented in a control chart. Various samples are gathered over a period of time and added to the chart. It is clear that there is a quality problem, or one that should be looked into, when there are measurements that extend above or below the norm line. One example of control charts with different specification limits, can be seen at Figure 2.10.



(a) Average level too high (b) Too variable (c) Average level too low & too variable

Figure 2.10: Example of 3 control charts with specification limits.

Figure 2.10 shows 3 processes that are failing to deliverwithin specifications for various reasons [20]: the mean level at which process (a) is producing is too high and would have to be brought down to meet specifications; process (b) is too variable; and process (c) is too variable and the mean level is too low.

Pareto charts contain line chart and bars in order to graph data. It is a method of finding out where the sources for defects come from and pinpointing the source of largest reoccurring defect. It is easy to understand due to its simply detailed visual presentation [10].

Many different process, tools and even techniques have been developed to identify causes, having software engineering organizations increasingly investing it, as a way to acknowledge failure and try to act effectively to investigate it.

As an example, Root Cause Analysis (RCA) can be a very valuable tool to any organization [51]. There are many techniques associated with RCA.

Some RCA methods and techniques are referred below:

- **Fishbone Diagrams**: also know as cause-effect diagram or "Ishikawa" diagram gets the name from its inventor, Kaoru Ishikawa, a pioneer in quality management techniques in Japan in the 1960's. The diagram is considered one of the seven basic tools of quality control. The 'fish head' represents the main problem and the potential causes of the problem are pointed out in the 'fish bones' of the diagram [95], as can be seen in 2.11. Fishbone diagram (FD) analysis can make complex systems organized, analyzing the causes of risk qualitatively, but it cannot realize the quantitative evaluation of risk [86].

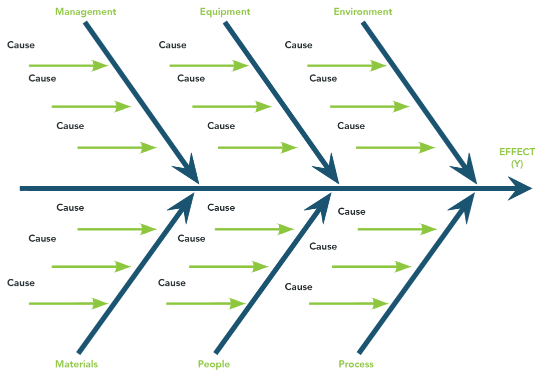


Figure 2.11: Fishbone Diagram Example [91].

- The 5 Why's
- Pareto Diagrams
- FaultTree
- FMEA (Failure Mode and Effect Analysis)
- Affinity Diagrams

The applicability of RCA Tools can be quite different. The 5 Why's Analysis is useful to look at one major cause and then drill-down to arrive at one root cause. In contrary, affinity diagrams are helpful when looking at several inter-related and detailed causes which have some things in common and can be grouped together to see the major causes. Fishbone diagram, in the other hand, is practical for looking at numerous major causes and then drill-down, probing to arrive at numerous root causes(s) based on their Cause and Effect Relationships [53].

Another important technique to refer is Defect causal analysis (DCA). DCA is a systematic process to identify and analyze causes associated with specific defect types, giving organizations opportunities to improve process assets and to take actions to prevent the recurrence of that defect type [47]. Reason why is commonly implemented in organizations/companies with high maturity process levels, like CMMI.

There are some differences between the techniques used in RCA and Defect Causal Analysis.

Causal analysis is a major component of modern process improvement approaches, e.g., the CMMI, Six Sigma, and Lean. The implementation of effective causal analysis methods has become progressively important as more software organizations transition to higher levels of process maturity where causal analysis is required as well as appropriate behavior [15]. While causal analysis of some form can be applied to almost any type of anomaly, defects are the most common subject of causal analysis. There are proofs that DCA brings a positive income for the organizations. For example, IBM's experience with DCA showed that defect rates declined by an average of 50 percent. Computer Sciences Corporation enjoyed similar benefits when it implemented DCA [14]. It can be concluded that reducing systematic sources of defects usually reduces development cost and cycle time, being a good investment for a company. Although DCA is not that expensive, it requires some investment.

The Software Engineering Institute's Capability Maturity Model envisions the deployment of DCA and other defect-prevention techniques at Level 5, Optimizing. Consequently, many organizations put off investment in DCA because they want to focus all their attention on exactly the level they are trying to attain currently — and that usually isn't Level 5 [14].

2.4 Mining Software Repositories

Mining Software Repositories (MSR) is a field of software engineering research, which aims to analyze and understand the data repositories related to software development [17]. The main goal of MSR is to make intelligent use of these software repositories to support the decision-making process of software development.

Research in the field shows that interesting and practical results can be obtained from the mining of these repositories, enabling developers and managers to better understand their systems and ultimately increase the quality of their products economically [17].

Topic Models, were originated from the field of information retrieval (IR) to index, search, summarize, infer and cluster a large amount of unstructured and unlabeled documents. A topic is a collection of terms that occur frequently in the documents of the corpus [81]. These models automatically discover structure within an unstructured corpus of documents, using the statistical properties of its word frequencies (tasks that were previously manually performed or not performed at all). In addition to discovering structure, topic models hold great promise for several reasons, for instance, they do not require training data, which makes them easy to use in practical settings [17].

The most used Topic Models in Software Engineering are Latent Semantic Indexing (LSI) [26] and Latent Dirichlet allocation (LDA) [11].

In some studies LSI is used as a tool for root cause analysis (RCA), assisting the root cause identification of a software failure. The authors built and executed a set of test scenarios that exercised the system's methods in various sequences [17]. Then, the authors used LSI to build a method-to-test co-occurrence matrix, which clustered tests that execute similar functionalities, helping to characterize the different manifestations of a fault.

According to Ahmed E. Hassan [38] experience and testimony, software repositories demonstrate to have very potential to support developers and managers while working in large software systems.

2.5 Synthesis

For high levels of maturity models (Levels 4 and 5), CMMI presupposes the application of different and more techniques as shown in the Figure 2.4.

ProcessPAIR allows the automation of some of these techniques and analysis tasks and many of these techniques originate from the "Statistical Process Control". In this way, organizations can create an organized process in order to control their projects and even do a predictive analysis of them, based on past and present data.

Moreover, the success of a company, the lowest level of a project or even a manager, may not be compromised as can be seen in the currently reported statistics (Chaos report), in which only 32% can be successfully finished.

Mining SW repositories seem to have future potential, but in this project the focus will not be on the analysis of repositories (with code) but rather the opportunity to use organized datasets in repositories for analysis.

Chapter 3

Process Improvement at Critical Software

3.1 CMMI Implementation

The journey of Critical Software (CSW) through quality processes had a massive boost in 2008, when they introduced the first Six Sigma quantitative and statistical practices, after receiving training from SW Six Sigma (an external company). Then, they started implementing and using several tools, as well as defect tracking tool adoption and deployment, like JIRA (from Atlassian). This was an adoption resulting from "CMMI2009" program, as well as WISE Projects Module (further detailed at 3.2).

CSW has launched several CMMI process improvement initiatives since 2004 having successfully achieved [27]:

CMMI-DEV V1.1 Level 3 in 2006;
CMMI-DEV V1.2 Level 5 in 2009;
CMMI-DEV V1.3 Level 5 in 2012.

In 2006 a tool to improve QMS (Quality Management System) from CSW was implemented which directly contributed to the journey for higher process capability levels. In 2009, with the passage to level 5 (CMMI-DEV V1.2 Level 5), the impact was huge. Some examples of benefits are shown next:

- Establishment and deployment of an Organizational Lessons Learned Process and Repository in WISE (this tool is going to be explained further in Section 3.2);
- Identification and deployment of a corporate defect tracking tool (JIRA);
- Deployment of WISE Projects Module with Process Performance, Analysis of CPI/SPI over time, etc.;

- Development of the first Balanced Scorecard at CSW: created and deployed an Organizational Performance Management Process to support it;
- Deployment of the first Performance Baselines and KPI's;
- Development and deployment of the first Code Review Performance Model and basis for CSW code review speed guidelines;
- Development and deployment of the first version of the QQAP - a tool for defect density and effort prediction in projects;
- Deployment of a measurement for training effectiveness, yearly training report, compilation of participants lists and training records;
- Deployment of tasks sub-attributes to identify the type of work being executed (e.g. Management, Testing, design, etc.);
- Development and delivery of newcomers training sessions, but also first corporate training materials;
- Creation and deployment of an Organizational Causal Analysis Process and establishment of the basis for formal problem causal analysis and tools.

In 2012 more benefits were achieved and of course Level 5 “opened the doors” for demanding, mission and business critical sectors such as aerospace, military and defense: the businesses areas of Critical Software.

3.2 Support Tools

In this business area, managers often struggle for a clear understanding of what is happening on their projects. A good use of process performance data depends on establishing a foundation of measurement collection, and in the CMMI, this starts with the Measurement and Analysis (MA) process area. In MA, an organization identifies its information needs and measurement objectives [18].

After referring several achievements from Critical Software with the support of CMMI, it is important to also detail the tools used in this company for process measurements and analysis, as follows:

- **Issue Tracking System:** JIRA is an issue tracking tool to collect defects data and Project Management activities. JIRA is also the tool used to support Agile/Scrum projects (providing backlog and sprint management features, burn-down charting, scrum boards, etc.) [4]. Several applications and plugins provide additional features in JIRA (e.g. Crucible, JIRA (Risk Management), JIRA (Xray), JIRA (Xporter), etc.). JIRA can also be shared with the Customer for issue management and resolution.

- **Project Management Information System (PMIS):** to collect effort data and help in Project Controlling. The first tool used by Critical Software was WISE but nowadays was replaced by Pulsar [4] where each employer can report time in each project, vacations and other information.
- **Static analysis:** for this kind of analysis CSW commonly uses SourceMonitor [4]. This freeware program allows the visualization of the software source code and the identification of relative complexity of its modules. For example, it can be used to identify the code that is most likely to contain defects which can be very helpful during a review. SourceMonitor, written in C++, runs through the code at high speed (typically at least 10,000 lines of code per second) and is able to provide the following:
  - Collection of metrics in a fast, single pass through source files;
  - Measurement metrics for source code written in C++, C, C#, VB.NET, Java, Delphi, Visual Basic (VB6) or HTML;
  - Modified Complexity metric option to select;
  - Saves metrics in checkpoints for comparison during software development projects;
  - Can operate within a standard Windows GUI or inside the scripts using XML command files;
  - Exportation metrics to XML or CSV (comma-separated-value) files for further processing with other tools.
- **Version Control System (VCS):** VCS is a generic designation of a system designed to manage a set of items, controlling their versions, access and integrity. A Version Control System can be designed to manage files or can be part of another system managing changes to metadata and information. Other systems widely used are CVS, SVN, Git, among others. SVN was the VCS used for Critical Software but now Git is the official Version Control System tool used to store and measure work product size [4].
- **Metrics Repository:** data4all is used to store effort, size and defects data. It is the official CSW metrics repository, where specific measures and calculated KPIs are stored. Typically metrics repositories are specialized uses of data warehouses [4].

3.3 Process Performance Models

An organization should define what it needs to collect [18] as they are the essential ingredients for process performance models. In these models the factors used are preferably controllable so that projects may take action to influence outcomes [43].

CMMI has several references to process performance models, as follows:

- OPP SP 1.5 Establish Process-Performance Models: Establish and maintain the process-performance models for the organization's set of standard processes;

- QPM SP 1.4 Manage Project Performance: Sub-practice 4 uses process-performance models calibrated with obtained measures of critical attributes to estimate progress towards achieving the project's quality and process-performance objectives;

- CAR SP 1.1 Select Defect Data for Analysis: useful for both either defects or problems and for predicting the impact and ROI that prevention activities will have;

- CAR SP 2.2 Evaluate the Effects of Changes: Evaluate the effect of changes on process performance;

- OID SG 1 Select Improvements: Analysis of process-performance baselines and models to identify sources of improvements;

Note: SP refers to Specific-Practice and SG to Specific-Goal. Other abbreviations concerns with Process Area involved, please refer to Abbreviations section.

Critical Software as it is a CMMI level 5 company, complies with above references, having established a long time ago models that allow them to collect factors to conduct prediction, in order to have a continuous improvement and take conscious actions.

The models predict quality and performance outcomes from factors related to one or more sub-processes involved in the development, maintenance, acquisition, or other kind of processes.

In the next Chapter 4 this statistical factors are evaluated in more detail and a very important initiative is referred along with them as can be seen in next section 3.4.

3.4 Lean QMS Initiative

Critical Software's Integrated Management System (IMS) is structured in top-down levels of detail, created to support all Critical Software activities and is formed by processes descriptions (independent from technologies), procedures, guidebooks and templates (that may be specific for technology type and/or engineering or business area).

The CSW IMS comprises two management sub-systems:

- Quality Management System (QMS): is mostly geared towards software/systems development and service supply;

- Research Development and Innovation Management System (RDIMS): is mostly oriented for R&D projects developed within the Innovation and Knowledge area.

A set of common processes are shared by both systems, but RDIMS was actually discontinued in 2014, being QMS still nowadays a system of great importance within Critical Software [3]. In this quality area a major initiative had taken place: Lean QMS. The main goals of Lean QMS initiative are:

1. Simplify or optimize by making the IMS more lean and cost-effective;

2. Maintain compliance with all international standards subscribed by the organization, namely ISO9001, AS9100, CMMI L5, and NATO AQAP;

3. Evaluate and Deploy QMS in a new structure and platform for the whole CSW Group;

4. Trigger and facilitate the alignment between CSW-UK and CSW-PT QMS;

5. Resolve weaknesses in Quantitative PM and Decision Analysis practices.

In this context, the LeanQMS team started an analysis of project audits performed between the period of 2012-2014. Table 3.1 presented next resumes the data collected in this analysis.

Table 3.1: Table with projects audited and respective number of non-conformances.

Project Type	**Projects No.**	**Non-Conformances**
Agile Software Development	2	38
Research & Development	5	23
Service	3	54
Software Development	25	241
Support & Maintenance	16	189
Verification & Validation	4	68

For this period, 55 projects were audited and a total of 613 non-compliances identified, which resulted in an average of 9.5 Non-conformances (NCs) per project.

The compilation of results for the period, allowed the LeanQMS Team the analysis of the recurrent non-compliances and find actions to mitigate and/or simplify them, ensuring higher compliance with the Quality Management System.

Furthermore, the LeanQMS team promoted several processes simplifications and practices incrementally. They picked some batches of work that taken into account the analysis of 606 non-conformities (originating from 66 project audits) within 2012 and 2014. Processes and the value of the practices were re-assessed and simplifications (reduction of waste), improvements and tailoring decisions applied.

This work resulted in a set of simplifications, as for example from the 204 Non-conformances considered in 5 processes, 116 NCs (57%) would not recur again.

3.5 Synthesis

Critical Software, a CMMI level 5 company has a major organizational process, only a little part of it was explained above. In the process (management) area in this CMMI level a required process is organizational performance management. Their performance has a huge impact in the way they work and it is of an extreme importance to their clients.

Significant amounts of performance data collected is being collected by them but not fully worked or sufficiently explored. In this Master Thesis it is the opportunity to look at it and use it, explore it in an ad-hoc way.

These set of metrics are useful to the identification of problem areas and their causes and this can be performed with the support of ProcessPAIR method. In this way, it is intended the extraction of important and useful information for CSW, with the identification of top-level indicators and calibration of ranges of values in order to support causal analysis within performance management area.

Chapter 4

Analysis of Project Performance Data

Critical Software has provided an internal audit dataset from 2011 until 2017. Using this dataset, information related with projects health was extracted, through a thorough data analysis. Also, a dataset containing information related with code reviews performance was supplied, that is going to be fully explored in next Chapter 5.

These chapters meet the objectives set out in this thesis: the selection and definition of relevant top-level performance indicators, as well as more detailed indicators and factors that affect these indicators, with the purpose of causal analysis support. In addition, the calibration of ranges of values for the classification of individual indicators in "semaphores" is performed.

Next sections 4.1 and 4.2 detail the software used to manipulate all the data provided and its characteristics.

4.1 SPSS Statistics

SPSS Statistics is a statistical software, developed by IBM, that provides a range of techniques including ad-hoc analysis, hypothesis testing and reporting, geospatial analysis and predictive analytics – making it easier to manage data, select and perform analyses [42]. Commonly, organizations use IBM SPSS Statistics to understand data, analyze trends, forecast and plan to validate assumptions and drive accurate conclusions [41].

This tool was chosen to perform several statistical analysis on the dataset provided in detriment of the most common tool: Microsoft Excel. SPSS is particularly good for analysis of questionnaire data and also, graphics are of a high standard. Furthermore, SPSS allows users to specify "value labels" a very useful feature, unavailable in Excel.

As "*Tania Prvan*, *Anna Reid* and *Peter Petocz*" refer in [83], actually Microsoft Excel was not designed to be a statistical computing package, but usually it is widely used for statistical analysis in business and industry, and also for statistics courses. However, because of that, in

Excel some diagnostics are incorrectly implemented and other common statistical procedures are not implemented at all, making it a *no choice* for this case.

Comparing some real issues: Excel holds data and SPSS holds data as well, but the major difference is that it also preserves information about that data (usually referred as metadata), e.g. measurement scale, types of values, variable labels, value labels and missing values [5].

Additionally, SPSS sends all output to a separate output window, which can be an excellent feature when manipulating a lot of data and generating several output data[5].

SPSS is well documented with great support [57] but Excel is a tool that almost everyone has and knows how to use, much popular than SPSS. In a study conducted by *Robert A. Muenchen* in [57] he concluded that open source packages are the preferred ones when people have to choose their data mining/analytic tools, and in here SPSS and Excel are both a good choice.

Another excellent option in this project could have been R. R is a free software environment for statistical computing and graphics, that compiles and runs on a wide variety of UNIX platforms, Windows and MacOS [35]. Being necessary the usage of associated programming language, R would require a more time-consuming learning curve compared to SPSS that possesses a more user-friendly, easy-to-use and intuitive interface, so SPSS was the tool chosen.

4.2 Description of the Dataset

The information provided by CSW was split into two datasets as is going to be explained next. The first dataset that Critical Software provided was related with performance of several projects through the years. The second dataset is described and analyzed in Chapter 5.

Critical Software performs several internal audits during the years to current projects being developed. This first dataset supplied - "CSW Projects Audit Scores History" - reflects the results of internal audits since 2011 and it has 12 columns, as follows:

- *ID*: internal identification of the project;
- *Project Name*;
- *Date*: closure date of audit performed;
- *Audit Score*: from 0 to 1 or 0 to 100% - value obtained in the internal audit;
- *# Audit*: also binary field, 1 or 2, corresponding to 1st or 2nd internal audit through the year;
- *Audit Type*: the different types of audit performed - Agile, Maintenance, Project, Quick and R&D.
- *CPI*: abbreviation of Cost Performance Index - performance data of the project at internal audit date. It is equal to earned value divided by actual cost, being 1 the ideal value [58] but very hard to achieve;

- *SPI*: abbreviation of Schedule Performance Index - performance data of the project at internal audit date. It is equal to earned value divided by planned value; [58]
- *Health Score*: composite indicator internally used: arithmetic mean of CPI, SPI and Audit Score;
- *BU*: abbreviation of Business Unit;
- *Year*: year of internal audit performed;
- *SPAE*: abbreviation for Software Product Assurance Engineer, it is the person responsible for monitoring the software development process, making sure that the software adheres to the standards set by the company.

Table 4.1 illustrates the number of entries (N) valid and missing in the first dataset provided for a total of 152 entries.

Table 4.1: First Dataset Frequencies Table.

N	**Audit Score**	**CPI**	**SPI**	**Health Score**	**Year**	Total
Valid	149	70	71	102	152	
Missing	3	82	81	50	0	
Total						152

Only the main fields used within this analysis were presented in the above Table. SPI and CPI are the fields with more missing values, although audit and health score were already calculated in this dataset.

4.3 Data Preparation

Before presenting the results obtained, it is relevant to refer some obstacles faced when inserting and manipulating data in SPSS, and how they were circumvented, starting with the first dataset available. Firstly, the date type from Excel file provided did not agree with SPSS standard (should be "dd-mm-yyyy"). Moreover, some rows did not have this field complete, having for example only the month, in this case it was assumed that it was in the first day of the month.

Moreover, there was more data from 2013 onwards, having 2011 and 2012 (the first years of data provided) a lot of missing values, although SPSS does not count empty or zero value cells, some techniques were a bit complicated to apply to them.

Later, a new sheet with exclusive information about SPAE's allocation to projects from 2011 and 2012 was provided. As previously mentioned, these two years were, in the first dataset, without information regarding the presence of SPAE's within projects. So, this new dataset allowed the addition of more information: 11 SPAE's from projects that already were in the first dataset and 26 SPAE's from new projects (without further info). Given this, some evaluations and techniques

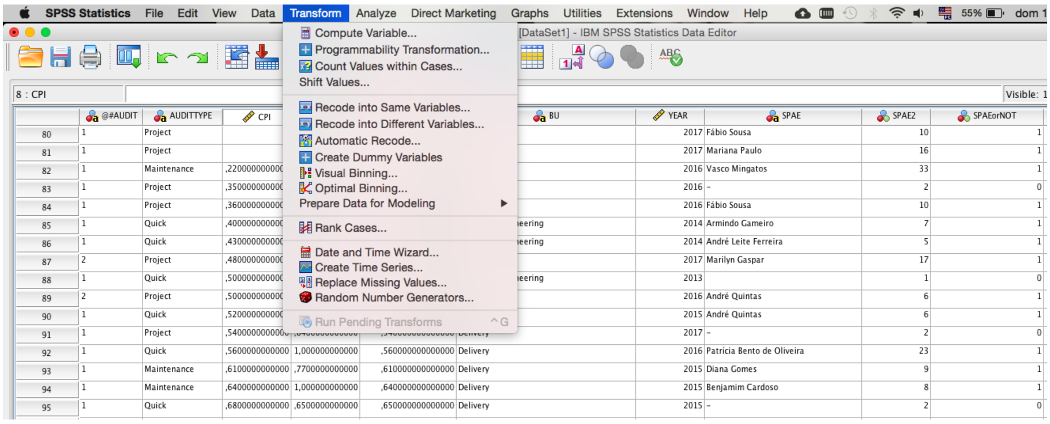


Figure 4.1: SPSS "Transform" option for first dataset.

were only conducted from 2013 to 2017 due to still some lack of data from previous years, as can be seen in the next section 4.4 and charts associated.

Another important thing to mention for this first dataset was the addition of a new numeric column: "SPAEorNOT" (represented by "*RECODED SPAE INTO 2 VARIABLES*"), in order to be able to collect information related with the presence or absence of SPAE within different projects and years. The only information related with SPAE in this dataset was his/her name in the case of presence and this kind of information was not sufficient for further evaluation. Transforming this variable in a discrete type demonstrating its presence or absence in 152 entries for each project was of extreme importance in order to measure its impact.

This is possible with a SPSS option to transform a variable from the dataset into a different one, as can be seen in Figure 4.1. In this way, a new column was created in this dataset, named "SPAEORNOT" and with the label "*RECODED SPAE INTO 2 VARIABLES*". This discrete/binary variable with an IF condition evaluated if there was any name in the column "SPAE" and in this case an "1" was the result. The opposite case had a "0" as output value (SPAE absence from project).

4.4 Descriptive Statistics

Prior to manipulating the first dataset in order to obtain desired outputs and conclusions, it was performed a statistic overview regarding the different process performance indicators.

The calculation of a number of summary statistics such as the mean, standard deviation and others, and the creation of informative graphics is the beginning of overall analysis. The aim at this stage is to describe the general distributional properties of the data, to identify any unusual observations (outliers) or any unusual patterns of observations that may cause problems for later analysis to be carried out on the data [30].

Following Table 4.2, presents the number of table entries for each field (represented by "N"), the minimum and maximum value and also, mean and standard deviation.

Table 4.2: Summary Table of descriptive statistics of main fields from 1st dataset.

Field	N	Min.	Max.	Mean	Std. Deviation
Audit Score	149	,117	1,000	,846	,168
Health Score	102	,215	1,932	,816	,251
CPI	71	,220	2,910	,999	,503
SPI	70	,280	1,610	,891	,241
"RECODED SPAE INTO 2 VARIABLES"	152	0	1	,52	,501

One goal established was to evaluate the impact of a SPAE presence within a project.

First, to perform this, it was necessary to create a new field, binary, representing the presence of a Software Product Assurance Engineer in a specific project (1) or absence (0): *RECODED SPAE INTO 2 VARIABLES*, as elucidated in previous section 4.3. This way, the whole evaluation and data interpretation could be more easily construed.

The diagram in Figure 4.2 is a representation of the influence of the factor SPAE for each performance indicator ("KPIs").

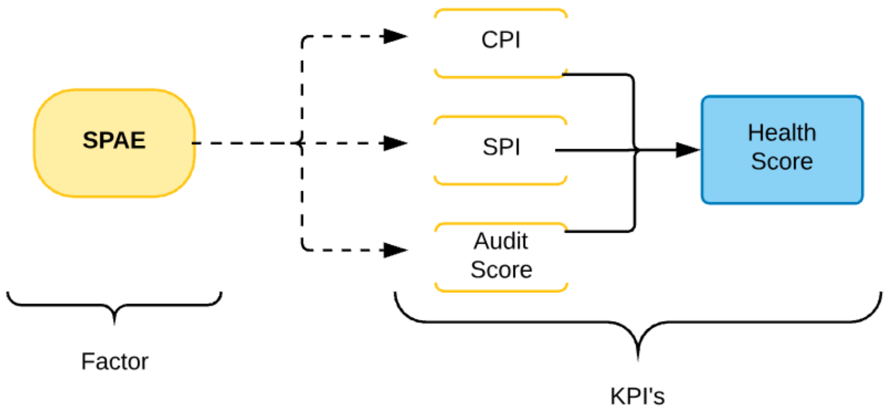


Figure 4.2: Diagram representing Factor (SPAE) influence over KPI’s under analysis.

Health Score, as explained before, is a composite indicator calculated as the arithmetic mean of SPI, CPI and Audit Score. Consequently, it is more difficult and risky to make assumptions based on this indicator. Thereby, Health Score was less used within this evaluation. Contrarily, Audit Score is the most used indicator, as it reflects the overall result of the internal audit.

4.4.1 Frequency Evaluation

After obtaining the previous descriptive statistics, a frequency evaluation can take place. This kind of evaluation was only performed for the first dataset due to the nature of the data.

The evaluation of SPAE influence (main goal for first dataset evaluation) began with some charts regarding the frequency of a certain KPI (key performance indicator) and where the SPAE presence and absence affects it the most.

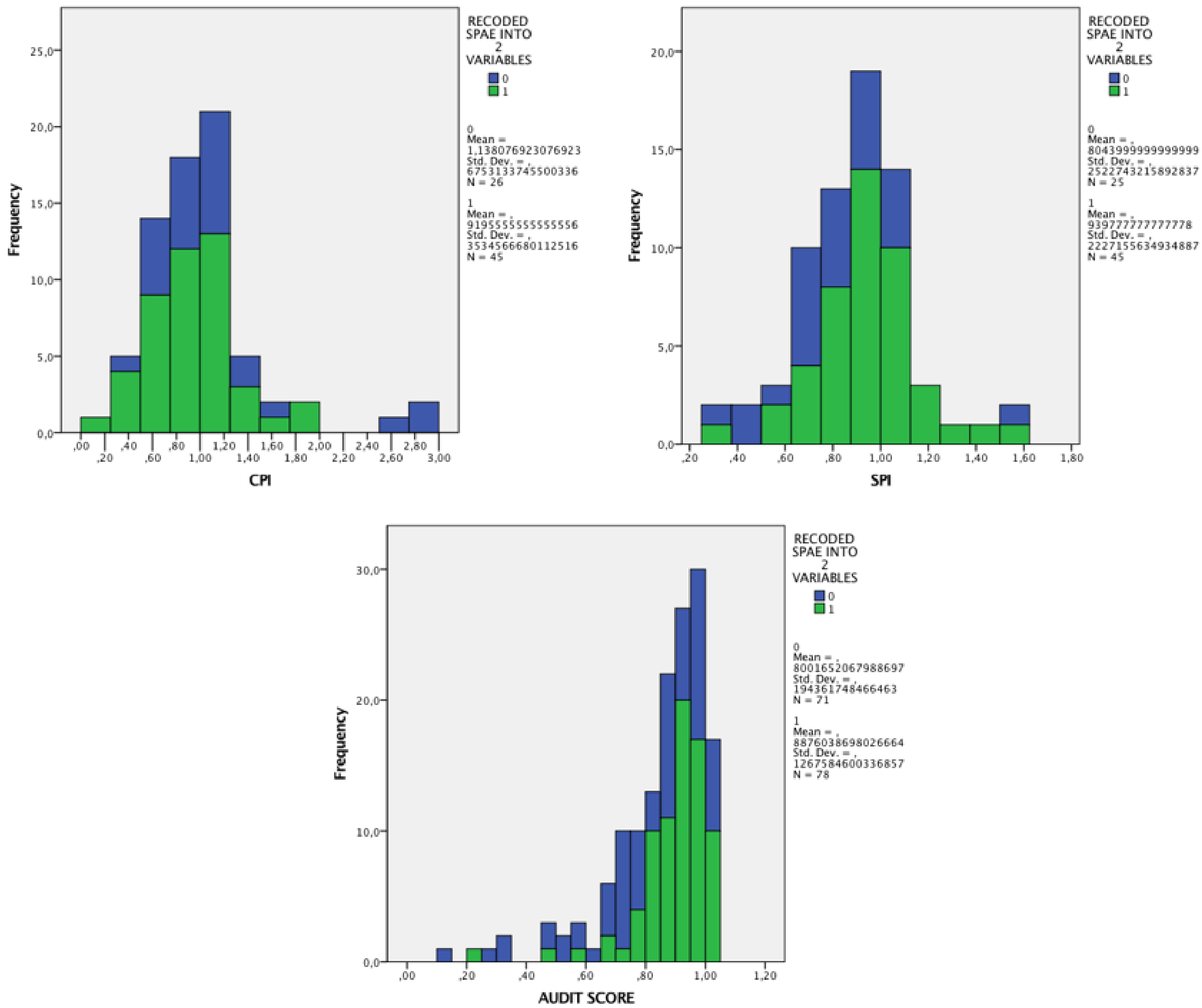


Figure 4.3: Frequency Charts for CPI, SPI and Audit Score vs SPAE.

Starting with the CPI key performance indicator, the chart at the upper left corner in Figure 4.3, there are some outliers in values above 2, but the majority of them are situated around 1 (ideal value). Almost double value for SPAE presence points number but there is a clear difference: the values majority it is situated in an ideal range [0.8-1.20]. The mean value for SPAE absence is above 1, so there is a chance that the project value skidded in this situation, in contrary of when SPAE is present within projects.

In relation to SPI (Figure 4.3 upper right corner) the distribution of the values fits better in a normal shape/distribution, without outliers. The mean values are lower (with SPAE ≈ 0.94, without ≈ 0.80), but still there is a clear distribution of SPAE presence values around 1, although some of them are above this value and in case of absence, not.

Health Score was not taken in consideration in this analysis, because it is a derived KPI computed from the other 3 KPIs, but it is available in Appendix A. For Audit Score, the lower chart in Figure 4.3, values distribution is clearly different, but the number of data is also more. Mean value for SPAE presence is below 1 (≈ 0.89) but also for SPAE absence (≈ 0.80), being this last value way much further than 0.9 (very close to ideal value). The outliers for this case are all for small values, however, overall are positioned very well for both cases, pointing out the presence of SPAE in a better range.

Subsequent charts illustrate the temporal evolution, from 2013 until 2017 of the main fields regarding *RECODED SPAE INTO 2 VARIABLES*. The years 2011 and 2012 were not considered in this analysis due to almost inexistence of data from those years, especially for the presence/absence of a SPAE.

4.4.2 Temporal Evolution

Subsequently to Frequency Evaluation, Temporal Evolution of KPIs of this dataset was performed. For all the following graphics, mean value is always used. 2011 and 2012 years were not considered in this analysis as explained before.

In the first pair of graphics, Figure 4.4 mean CPI and mean SPI can be compared through the years with SPAE impact. Can be seen that SPAE presence (green line - Figure 4.4 (b)) is increasing through the years and in SPI case had a good impact, always above approximately 0.90 values. For CPI, Figure 4.4 (a), the case is not the same: SPAE presence started to have a major impact only in 2014 and seems to be decreased since past year.

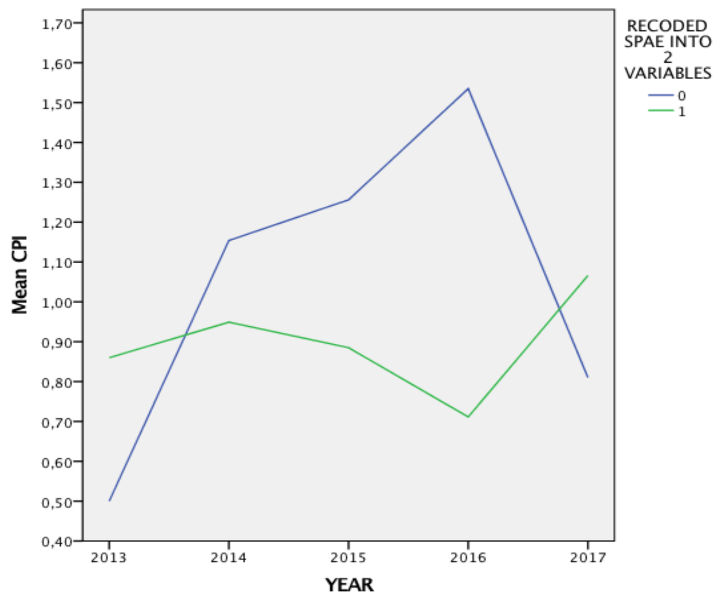
There is a clear peak in both charts in 2016. It was concluded that one initiative conducted by Critical Software in 2014 to 2015 had repercussions later on 2016. This initiative called "Lean QMS" (detailed in previous Chapter 3, Section 3.4) examined all the NC's (Non Conformities) in the system and tried to understand how to avoid them in the future, at least half of them. So, from 2015 the NC's became a correction and this had an impact in some KPI's like SPI and CPI.

The other charts evaluated for temporal evolution was to Audit and Health Score, as can be seen at Figure 4.5. Being Health Score a composite indicator, the chart is less linear (Figure 4.5 (a)) than the others, but also a peak in year 2016 stands out. In the last 2 years, the SPAE presence (green line) is gaining ground in the best values of Health Score: stepping-up from approximately 0.70 and rising to 0.90 in one year. The trend seems to be that Health Score values are growing to an almost optimal value around 1.0.

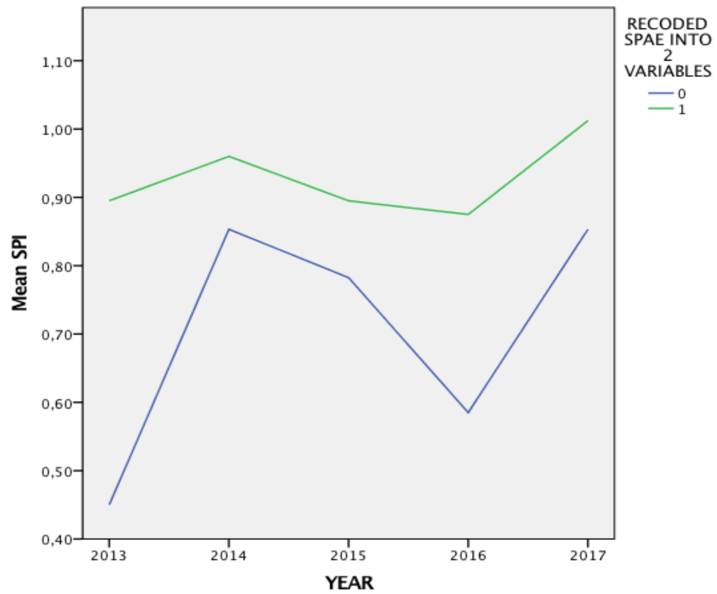
Regarding Audit Score (Figure 4.5 (b)) the temporal evolution was plotted since 2011 until 2017, due to the new data that was merged with the dataset provided. Even without CPI and SPI values, Audit Score and SPAE presence/absence was evident. Is not comparable with all the charts from this subsection but an individual analysis can be discussed.

It is clear that the SPAE presence had always a good influence over years for Audit Score temporal evolution appraisal, since it is also almost reaching an ideal value (around 1.0), but in general the values for SPAE absence were below through the years.

The overall tendency when assessing this temporal evolution for KPI's is that SPAE presence is overcoming SPAE absence and growing for almost optimal values. 2018 values are probably going to be even better.



(a) Mean CPI through years vs "*Recoded SPAE*".



(b) Mean SPI through years vs "*Recoded SPAE*".

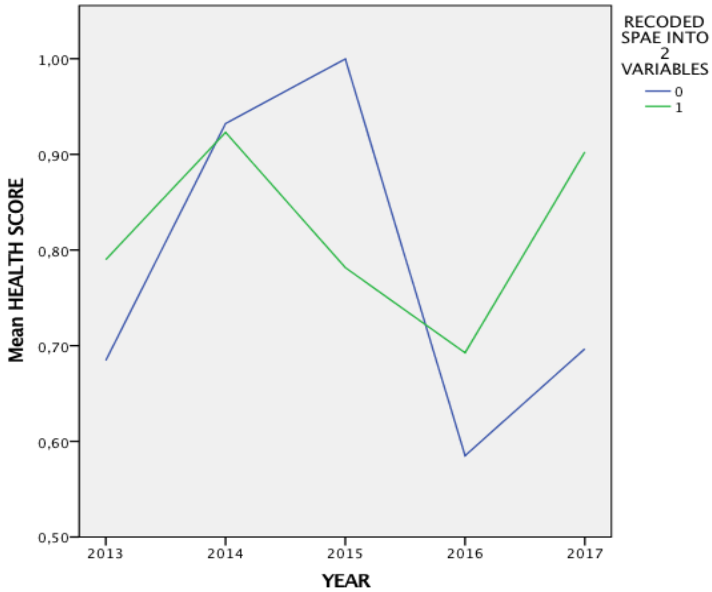
Figure 4.4: Temporal Evolution of mean SPI and CPI values and effect of SPAE within projects.

4.5 Hypothesis Testing

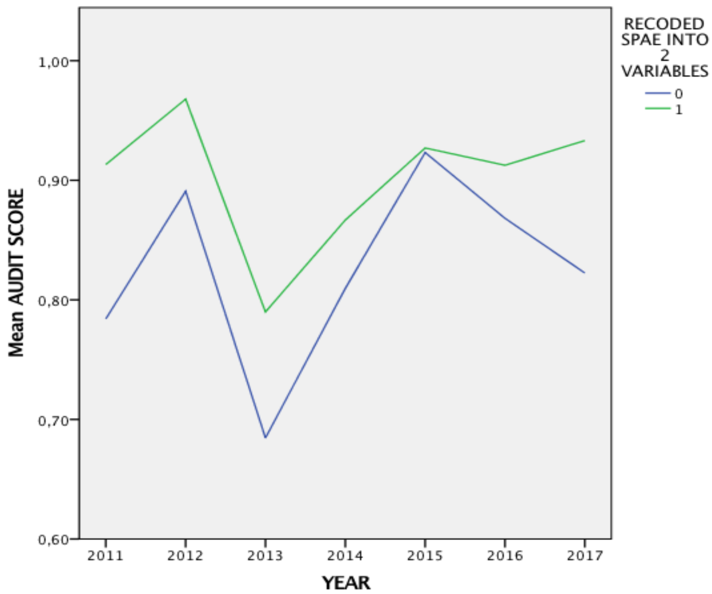
Hypothesis Testing uses statistics to choose between hypothesis regarding whether data is statistically significant or occurred by chance alone [16].

Two methods (One way ANOVA - 4.5.2 - and Mann-Whitney U Test - 4.5.3) were chosen in order to evaluate in detail, SPAE influence in KPI's selected within this dataset.

Next, subsection 4.5.1, starts with four Research Questions for subsequent analysis.



(a) Mean Health Score through years vs "*Recoded SPAE*".



(b) Mean Audit Score through years vs "*Recoded SPAE*".

Figure 4.5: Mean value Audit and Health Score through the years affected by the presence or absence of SPAE in projects.

4.5.1 Research Questions

In order to verify the SPAE influence on relevant KPIs within CSW projects, four research questions (RQ) were investigated, as follows:

1. Does the SPAE presence have a positive influence on the SPI?

2. Does the SPAE presence have a positive influence on the CPI?
3. Does the SPAE presence have a positive influence on the Health Score?
4. Does the SPAE presence have a positive influence on the Audit Score?

Succeeding subsections clarify the further analysis performed in order to answer these research questions.

4.5.2 One-way ANOVA

One-way ANOVA (Analysis of Variance with One Factor) approach was applied to the first dataset in order to see if two variables are different from each other [85]: in this case the factor "SPAE" and each of the three main KPI's (SPI, CPI and Audit Score). The procedure is a One-way ANOVA, since there is only one independent variable [40].

This approach can help understand if there is an interaction between the independent/grouping variable or factor (SPAE in this case) on the dependent variable (each of KPI's) [80]. But, for this, it is necessary to fulfill certain assumptions, which, in summary, can be described as follows [23]:

- The groups being observed have to be independent from each other;
- Each group of observations must come from a population with normal distribution;
- The variance of the populations should be the same (variance homogeneity).

This analysis will indicate whether there are significant differences in the mean scores with the presence/absence of SPAE across three performance indicators selected (Health Score is not much relevant in this analysis since it is a composite indicator).

The Variance Analysis hypothesis for one factor (ANOVA: One-Way) involves the definition of two hypothesis:

- **Null Hypothesis** (*H0*): The means of all groups are the same.
- **Alternative Hypothesis** (*H1*): At least two means differ each other.

Post-hoc tests can then be used to find out where these differences lie after one-way ANOVA conducted.

This procedure in SPSS generates 4 tables: Descriptives, Test of Homogeneity of Variances, ANOVA and Robust Tests of Equality Means, as can be visualized in below Figures for each of the KPI. The following items describe each table carefully.

1. The first table, "**Descriptives**" contains statistical information about the dependent variable: the number of observations (N), the Mean, the Standard Deviation, the Standard Error of the Mean, a 95% confidence Interval for the Mean, the Minimum value, and the Maximum value.

2. The second table, "**Test of Homogeneity of Variances**", provides Levene's F Test for Equality of Variances, which is the most commonly used statistic to test the assumption of homogeneity of variance [40]. The main value observed in this table is Significance (Sig.): if it is less than 0.05 it is advised by literature [12] to consult "Robust Tests of Equality of Means" (fourth and last table generated). For this, Levene's test uses the level of significance set a priori for the ANOVA ($\alpha = 0.05$) to test the assumption of homogeneity of variance.

3. The third table, "**ANOVA**", also called "F-ratio Test" [75], contains information about the ANOVA test comparing means both between and within groups. It includes the Sum of Squares (a measure of variance), df (degrees of freedom), Mean Square, the F value, and the Sig. value (when the Sig. value is .05 or less, the probability that the difference between the groups was due to chance is 5% or less) [16]. It is used when there is homogeneity of variance. If the test returns a significant f-statistic, an ad hoc test (like the Least Significant Difference test) is needed in order to know exactly which groups had a difference in means [16].

4. The fourth table, "**Robust Tests of Equality of Means**" presents two supplemented F-ratio tests: (F-)Welch and (F-)Brown-Forsythe [75]. These tests should be used when homogeneity of variance of the data is invalid or inexistent [36]. F increases with the effect of the independent variable. Thus, the larger the F ratio is, the more reasonable it is that the independent variable has had a real effect [40].

   The Brown and Forsythe Test is a test for equal population variances. It is a robust test based on the absolute differences within each group from the group median. Welch's Test is a parametric test which compares two means to see if they are equal [84] and it is an alternative test for the one factor analysis of variance F test. The Welch test is more powerful and more conservative than the Brown-Forsythe test [40], [36].

That being said, the first set of tables generated with SPSS for One-way ANOVA method was for SPI performance indicator (Figure 4.6). The mean value for 1 (SPAE presence) in "Descriptives" Table is clearly higher with an upper bound closet to optimal value (1).

In "Test of Homogeneity of Variances" the most important value to refer is Significance (represented by "Sig." in the table and "p" in literature [40]) which in this case is very high: 0.509 (comparing with 0.05 reference value) and Levene Statistic very low: 0.441. Because the Sig. value is greater than the alpha of 0.05 ($p > 0.05$), the null hypothesis, H0 (no difference) is retained for the assumption of homogeneity of variance and conclude that there is not a significant difference between the group's variances. That is, the assumption of homogeneity of variance is met [40], [74].

However, in table "ANOVA" sig. value is equal to $0.023 < 0.05$, so can be concluded that there is a statistically difference between groups.

Regarding CPI one-way ANOVA results, at Figure 4.7 can be said that they are very different from SPI. Starting with "Descriptives" table, the mean values are exceeding the optimal value (1)

Descriptives

SPI

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
0	25	,804400000	,252274322	,050454864	,700266278	,908533722	,280000000	1,52000000
1	45	,939777778	,222715563	,033200476	,872866615	1,00668894	,320000000	1,61000000
Total	70	,891428571	,240905813	,028793752	,833986608	,948870535	,280000000	1,61000000

Test of Homogeneity of Variances

SPI

Levene Statistic	df1	df2	Sig.
,441	1	68	,509

ANOVA

SPI

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	,295	1	,295	5,399	,023
Within Groups	3,710	68	,055		
Total	4,004	69			

Robust Tests of Equality of Means

SPI

	Statistic[a]	df1	df2	Sig.
Welch	5,024	1	44,711	,030
Brown-Forsythe	5,024	1	44,711	,030

a. Asymptotically F distributed.

Figure 4.6: SPSS Tables for SPI One Way ANOVA Procedure.

Descriptives

CPI

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
0	26	1,13807692	,675313375	,132439849	,865311948	1,41084190	,350000000	2,91000000
1	45	,919555556	,353456668	,052690209	,813365417	1,02574569	,220000000	1,97000000
Total	71	,999577465	,502637150	,059652055	,880605192	1,11854974	,220000000	2,91000000

Test of Homogeneity of Variances

CPI

Levene Statistic	df1	df2	Sig.
4,781	1	69	,032

ANOVA

CPI

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	,787	1	,787	3,213	,077
Within Groups	16,898	69	,245		
Total	17,685	70			

Robust Tests of Equality of Means

CPI

	Statistic[a]	df1	df2	Sig.
Welch	2,350	1	33,070	,135
Brown-Forsythe	2,350	1	33,070	,135

a. Asymptotically F distributed.

Figure 4.7: SPSS Tables for CPI One Way ANOVA Procedure.

for SPAE absence (0) and this can help validate the early analysis performed in 4.4.1 and 4.4.2 previous subsections.

Again, in contrast with SPI, the Levene statistic is very high (4.781) and p-value very low (0.032), below the 0.05 threshold. This means that the equal variance assumption has been violated, so Welch and Brown-Forsythe statistic values should be consulted in "Robust Test of Equality of Means" (adjust F statistic) [40].

In the "ANOVA" table stands out the F ratio value that is the variance between the groups divided by the variance within the groups: F(df1,df2), where df1 is the value between groups (combined) and df2 the value within groups, respectively [75]. So for CPI, $F(1, 33.070) = 2.350$ and $p > 0.05$ (0.135).

Can be concluded that for CPI, the null hypothesis can not be rejected: CPI is on average better with SPAE (possible due to some very high outliers), but the difference is not statistically significant.

The last set of tables presented in Figure 4.8 refers to Audit Score. For "Descriptives" table the values between SPAE presence (1) and absence (0) are very close, with a mean difference of only approximately 0.08, but the upper bound for SPAE presence is closer to ideal value. In next table,

Descriptives

AUDIT SCORE

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
0	71	,800165207	,194361748	,023066496	,754160532	,846169882	,117044209	1,00000000
1	78	,887603870	,126758460	,014352569	,859024253	,916183487	,214793417	1,00000000
Total	149	,845938467	,167770249	,013744274	,818778097	,873098836	,117044209	1,00000000

Test of Homogeneity of Variances

AUDIT SCORE

Levene Statistic	df1	df2	Sig.
13,054	1	147	,000

ANOVA

AUDIT SCORE

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	,284	1	,284	10,762	,001
Within Groups	3,882	147	,026		
Total	4,166	148			

Robust Tests of Equality of Means

AUDIT SCORE

	Statistic[a]	df1	df2	Sig.
Welch	10,359	1	118,542	,002
Brown-Forsythe	10,359	1	118,542	,002

a. Asymptotically F distributed.

Figure 4.8: SPSS Tables for Audit Score One Way ANOVA Procedure.

"Test of Homogeneity of Variances", the p-value (Sig.) is approximately zero. So, this means that the null hypothesis (H0) should be rejected (population means are equal). When the alternative holds, the expected value of F ratio is larger, and the null hypothesis should be rejected [56] and actually it is the higher F value in this analysis (10.762 comparing with 5.399 for SPI and 3.213 for CPI). A large F ratio represents more variability between groups than within each group.

One-way ANOVA approach for Health Score can be visualized in Appendix A.

However, the results of the ANOVA alone do not indicate which groups differ significantly [16], and this is why post-hoc tests are important for a further evaluation.

One advantage of ANOVA technique is the variance comparison (variability in scores) between different groups with the variability within each of the groups [12] that it can calculate. But as a con, it can not make perceptible which groups are different, will be referred in 4.5.3, although this is not an issue in this case since the comparison was performed only with two groups (SPAE and each KPI).

Synthesis

In conclusion, based on the ANOVA analysis:

$\boxed{\text{SPI}}$ The presence of SPAE has a positive impact on the SPI (with a higher mean value, closer to the ideal value, than in the absence of SPAE) and the impact (difference of means) is statistically significant.

$\boxed{\text{CPI}}$ The presence of SPAE has a negative impact on the CPI (with a lower value than mean threshold) and the impact (difference of means) is not statistically significant.

$\boxed{\text{Audit Score}}$ The presence of SPAE has a positive impact on Audit score (with a higher mean value, closer to the ideal value, than in the absence of SPAE) and the impact (difference of means) is statistically significant.

4.5.3 Mann-Whitney U Test

The Mann-Whitney U test is a nonparametric statistic test used to assess whether two independent groups are significantly different from each other [66]. This test was chosen in order to fill some gaps from one-way ANOVA method previously explained for the first dataset.

The Mann-Whitney test is appropriate to find out whether the median of two continuous and independent populations are equal. One advantage of this approach is that the two samples involved do not need to have the same size [24].

The test is based on the joint ranking of the observations from the two samples (as if they were from a single sample). If there are ties, the tied observations are given by the average of the ranks for which the observations that are competing [30]. The test ranks all of the dependent values, i.e., lowest value gets a score of one and then uses the sum of the ranks for each group in the calculation of the test statistic [49].

Similar to (one-way) ANOVA, Mann-Whitney U test has its assumptions and hypothesis, as described in the following items:

- **Null Hypothesis** (*H0*): the two groups being compared have identical distributions (for two normally distributed populations with common variance, this would be equivalent to the hypothesis that the means of the two populations are the same [30]);
- **Alternative Hypothesis** (*H1*): the distribution of the groups differ in location (median) [67].

The null hypothesis (*H=0*) is a statement about a population parameter, such as the population mean, that is assumed to be true [28]. An alternative hypothesis (*H=1*) is a statement that directly contradicts a null hypothesis by stating that the actual value of a population parameter is less or greater than, or not equal to the value stated in the null hypothesis.

This test can be performed taken into consideration at least two assumptions [49], [79]:

1. The two groups must be independent;
2. The dependent variable must be ordinal or numerical (continuous).

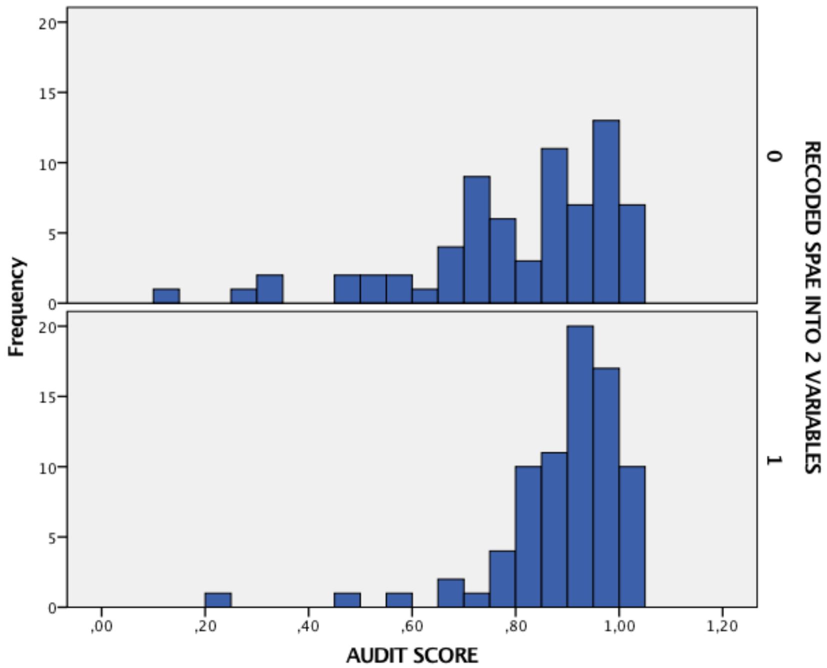


Figure 4.9: Histogram distribution comparison between "Recoded SPAE into 2 variables" and Audit Score.

These assumptions are necessary because it is only appropriate to use a Mann-Whitney U test if the data fulfill the assumptions that are required for a Mann-Whitney U test to give a valid result.

Means and Medians Tests

SPAE is in one of two groups (Group = presence or absence) and the mean difference in KPI's is of interest so an independent t-test is appropriate. However, one of the assumptions of an independent t-test is that the dependent variable needs to be approximately normally distributed for both groups [49].

Therefore, in order to report the difference between groups as medians, the shape of the distributions of the dependent variable by group must be similar. It does not matter if the distributions have a different location on the x-axis, they just have to have a similar shape [49], this way the chart represented in Figure 4.9 was developed in order to really understand if this premise was being fulfilled.

As can be seen in Figure 4.9 the distributions have a similar shape, with the values located in similar ranges (although less values for SPAE absence (0) in a higher range), so the test can continue.

The test is developed based on the Sum of Ranks occupied in the ordered sample by the elements of the initial sample of smaller size [24]. An illustrative boxplot chart was developed

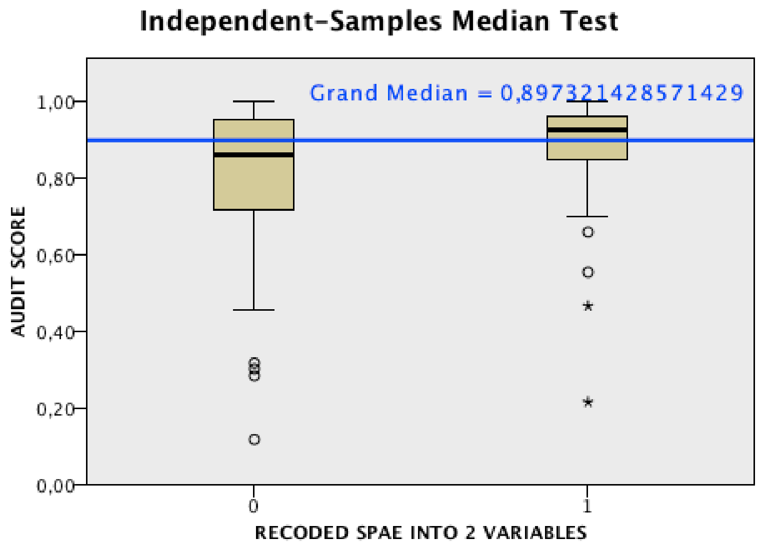


Figure 4.10: Boxplot representing grand median value for "Recoded SPAE into 2 variables" and Audit Score.

as can be seen in Figure 4.10, where for Audit Score example, can be observed that the median values are relatively close with SPAE presence and absence.

There are two different procedures in SPSS Statistics to run a Mann-Whitney U test: a new and legacy procedure.

In next subsection 4.5.3.1, a brief reference to the Mann-Whitney U test old approach in SPSS and its results is laid out.

4.5.3.1 *Old Approach*

In this approach the Mann-Whitney U test uses mean ranks rather than medians to calculate the output value.

As already said, the distribution of KPI's and "Recoded SPAE into 2 variables" is similar and obey to a normal distribution, so this assumption is accomplished and it is why this legacy procedure (old approach) is only presented for Audit Score vs SPAE presence/absence, by way of example.

This procedure generates three tables: Descriptive Statistics, Ranks and Test Statistics. As an example, Figure 4.11 represents the old Mann-Whitney procedure which can be consulted for Audit Score (only exemplification purposes so only one KPI was evaluated for this legacy procedure).

This table does not provide a lot of vital information since it is not possible to evaluate a lot of significant differences between Audit Score and "Recoded SPAE into 2 variables" groups. Due to data conform to a normal distribution in this table we can take a closer look to mean and interquartile range (IQR) from 25th to 75th percentile, to observe where Audit Score is situated [79]. Still not possible to compare both, this is another reason this legacy procedure was discarded.

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum	Percentiles 25th	50th (Median)	75th
AUDIT SCORE	149	,845938467	,167770249	,117044209	1,00000000	,791367624	,897321429	,958718402
RECODED SPAE INTO 2 VARIABLES	152	,52	,501	0	1	,00	1,00	1,00

Mann-Whitney Test

Ranks

	RECODED SPAE INTO 2 VARIABLES	N	Mean Rank	Sum of Ranks
AUDIT SCORE	0	71	64,36	4569,50
	1	78	84,69	6605,50
	Total	149		

Test Statistics[a]

	AUDIT SCORE
Mann-Whitney U	2013,500
Wilcoxon W	4569,500
Z	-2,874
Asymp. Sig. (2-tailed)	,004

a. Grouping Variable: RECODED SPAE INTO 2 VARIABLES

Figure 4.11: Old Mann-Whitney U Test Approach generated by SPSS for Audit Score.

Ranks table is where the mean and sum of ranks are presented. Also the number of points for each (N) dependent variable. In this example can be observed that the mean rank is clearly higher with SPAE presence with only 7 entries difference.

And finally, test statistics table is similar to an auxiliary table generated in the new procedure that comes with a chart, but with less fields, that is described in next sub section 4.5.3.2.

4.5.3.2 *New Approach*

Although the old approach for carrying out Mann-Whitney U Test was previously referred, the new procedure is the one recommended when the two distributions have the same shape [79], so it was conducted for all KPI's.

In this new approach one table is generated with the final result "Hypothesis Test Summary" - to reject or retain the null hypothesis (Figure 4.12). Then, double clicking in the table a chart is presented (Figure 4.13) with the distributions of the two groups and also the mean ranks for each group [49]. Also, a small table with some information like number of points (N), standard error and asymptotic sigma is presented, and for example, this sig. value according to literature, should be consulted if a large sample was used [92] (not more than 150 points under analysis).

In addition, there are three test statistics in this table (U [Mann-Whitney U], W [Wilcoxon W] and Z): the Wilcoxon W, which simply presents the lowest sum of ranks in order to calculate the p-value (Asymp. Sig), for Z statistic and resulting p-value, SPSS uses an approximation to the standard normal distribution, but this is only appropriate for large samples so if the sample

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of AUDIT SCORE is the same across categories of RECODED SPAE INTO 2 VARIABLES.	Independent-Samples Mann-Whitney U Test	4,000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is ,05.

Figure 4.12: Table "Hypothesis Test Summary" generated by SPSS for Mann-Whitney U Test for Audit Score.

size is 40 or less, an Exact test is performed automatically and another output entitled "Exact Sig. [2*(1-tailed Sig.)]" would appear [49]. Here the sample size is large (more than 40 samples) so the Z approximation p-value can be reported. Also, with the support of this table calculation of effect sizes (ES) can be performed, dividing the absolute (positive) Standardized test statistic Z by the square root of the number of pairs (N) [49].

$$\frac{Z}{\sqrt{N}}$$

According to Cohen's classification of ES 0.1 is considered to be a small effect, 0.3 a moderate effect and 0.5 and above a large effect [9].

Effect sizes can be thought of two ways: as the average percentile standing of one group in relation to other, or in terms of the percent of non-overlap of one group's scores with the other group. In the first case, an ES of 0.0 indicates that the mean of the one group is at the 50th percentile of the other. An ES of 0.8 indicates that the mean of one group is at the 79th percentile of the other group, and an effect size of 1.7 indicates that the mean of one is at the 95.5 percentile. In case of non-overlap percentage: an ES of 0.0 indicates that the distribution of scores for one group overlaps completely with the distribution of scores of the other group (there is 0% of non-overlap). An ES of 0.8 indicates a non-overlap of 47.4% in the two distributions and an ES of 1.7 indicates a non-overlap of 75.4% in the two distributions.

However, this last table is not going to be presented in this results due to lack of relevance to this analysis, only some values are going to be referred if they are necessary to support the analysis.

Starting with Audit Score, the decision is to reject the null hypothesis (*H0*), as can be seen in Figure 4.12: the distribution of "Audit Score" is the same across categories of "Recoded SPAE into 2 variable" and Sig. value is 4.0.

Although the distribution is similar, in Figure 4.13 it can be observed that mean rank for SPAE presence (1) is higher, so can be concluded that values of Audit Score for this distribution with presence of SPAE are higher. Here the sample size is large (more than 40 samples) so the Z approximation p-value of 0.004 can be reported. Relative to calculation of effect size, as follows:

$$\frac{Z}{\sqrt{N}} = \frac{2.874}{149} = 19.28859$$

The effect size is of approximately 19.3 which is a large effect according to Cohen's classification of effect sizes, what means that SPAE has a high magnitude effect in Audit Score.

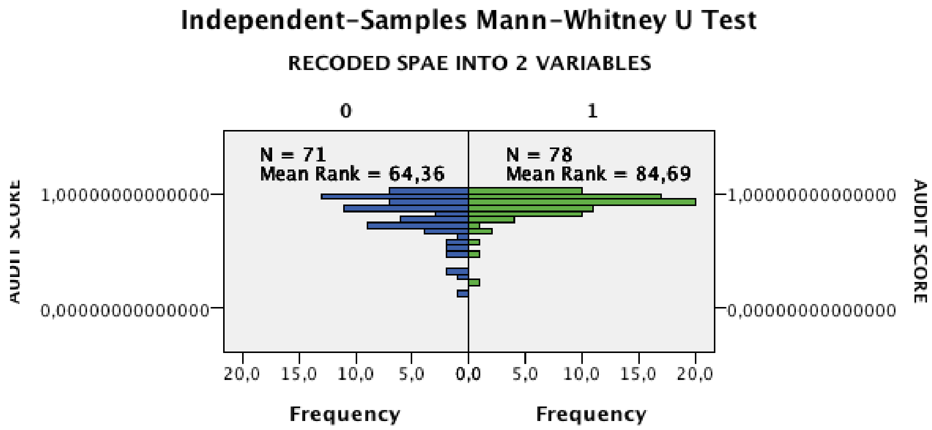


Figure 4.13: Accompanying Chart for Mann-Whitney U Test for Audit Score and "Recoded SPAE into 2 variables".

For SPI the decision is the same as for Audit score: reject the null hypothesis. The mean rank is also higher with SPAE presence (Figure 4.14), but contrarily to Audit Score, exists a difference of 20 points (N) between absence (0) and presence (1) of SPAE, so it is not fair to conclude a lot more relative to mean ranks.

The Z approximation p-value is of 0.015, less than the 0.05 standard value and relative to effect size the value is higher than in Audit Score evaluation, as can be observed next:

$$\frac{Z}{\sqrt{N}} = \frac{2.434}{70} = 34.7714$$

The effect size is of approximately 34.8 which is a large effect according to Cohen's classification of effect sizes, what means that SPAE has a high magnitude effect in SPI performance indicator.

Continuing the analysis for CPI, such as in One-way ANOVA test, this performance indicator presents again suspicious results: a very large value of Sig. and the decision to retain the null hypothesis (Figure 4.15). In this case, the mean rank is situated in a lower value for the case when SPAE is present in the projects. Even with more points to evaluate (N=45 when 1 vs N=26 when 0) the mean rank for SPAE presence is 34.06 and without SPAE 39.37. Should be noted the presence of outliers in this chart which can influence overall result.

The Z approximation p-value is of 0.296, higher than the 0.05 standard value. The calculation of effect size can be checked next:

$$\frac{Z}{\sqrt{N}} = \frac{1.045}{71} = 14.7183$$

The effect size is of approximately 14.8 which is a large effect according to Cohen's classification of effect sizes, what means that SPAE has a high magnitude effect in CPI performance indicator.

In Mann-Whitney U Test it was decided to also present in this section the results relative to Health Score performance indicator. Because it is a composite indicator, it was predictable it was going to also retain the null hypothesis and it was verified, as can be checked in Figure 4.16. Mean rank is higher with SPAE presence but it has much points to rank comparing with SPAE absence.

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of SPI is the same across categories of RECODED SPAE INTO 2 VARIABLES.	Independent-Samples Mann-Whitney U Test	15,000	Reject the null hypothesis.

Asymptotic significances are displayed. The significance level is ,05.

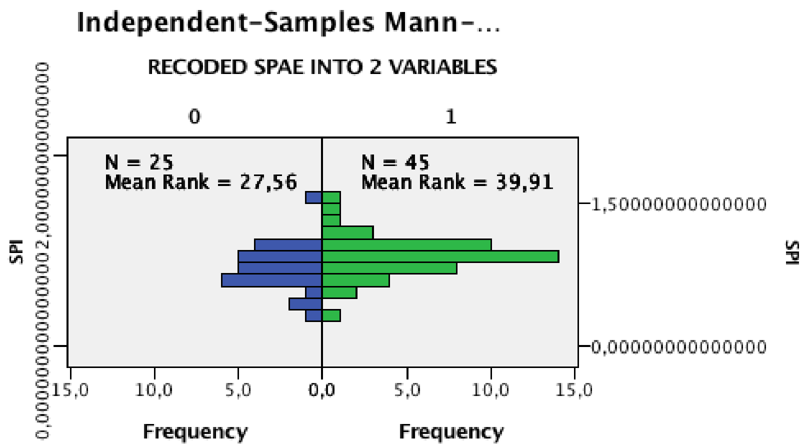


Figure 4.14: Table and chart representing Mean Rank result for SPI after Mann-Whitney U Test performed.

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of CPI is the same across categories of RECODED SPAE INTO 2 VARIABLES.	Independent-Samples Mann-Whitney U Test	296,000	Retain the null hypothesis.

Asymptotic significances are displayed. The significance level is ,05.

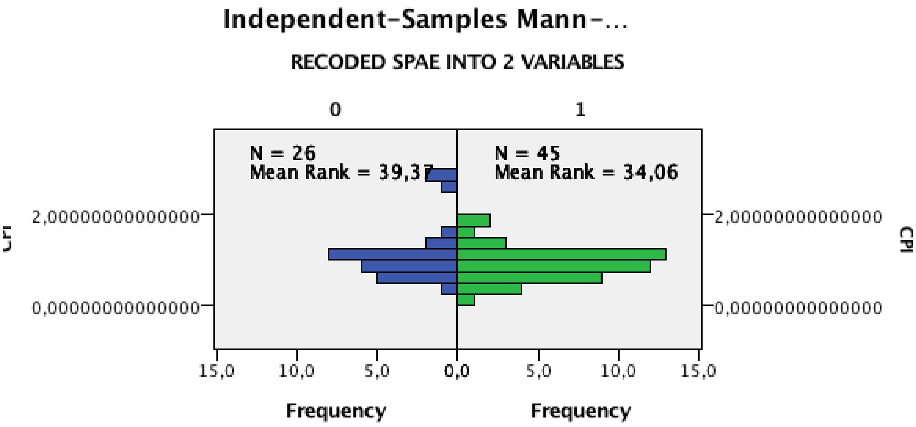


Figure 4.15: Table and chart representing Mean Rank result for CPI after Mann-Whitney U Test performed.

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of HEALTH SCORE is the same across categories of RECODED SPAE INTO 2 VARIABLES.	Independent-Samples Mann-Whitney U Test	319,000	Retain the null hypothesis.

Asymptotic significances are displayed. The significance level is ,05.

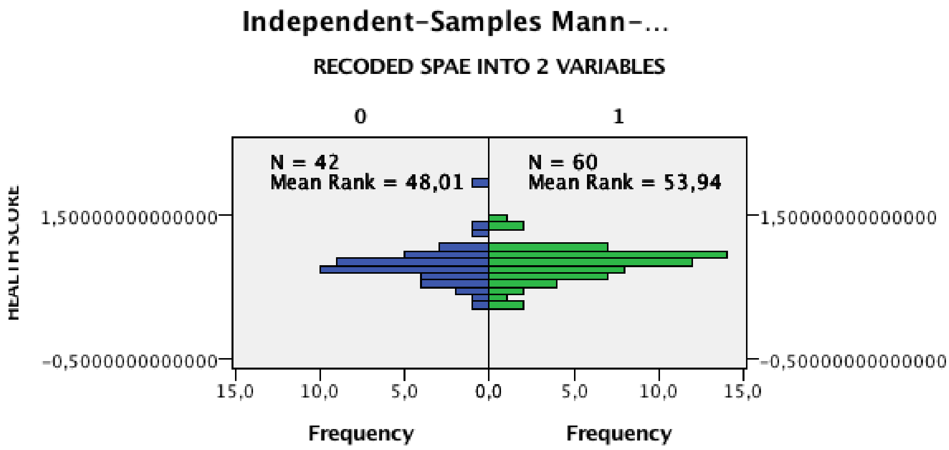


Figure 4.16: Table and chart representing Mean Rank result for Health Score after Mann-Whitney U Test performed.

The Z approximation p-value is of 0.319, way higher than the 0.05 standard value. The calculation of effect size can be checked next:

$$\frac{Z}{\sqrt{N}} = \frac{0.996}{102} = 0.009764$$

The effect size is of approximately 0.0097 which is a very small effect according to Cohen's classification of effect sizes, what means that SPAE has just a little magnitude effect in Health Score performance indicator, which can make sense since it is a composite indicator (this can make the influence of SPAE be masked because it has a positive impact in some other indicators and negative in others and this can nullify its impact on this indicator).

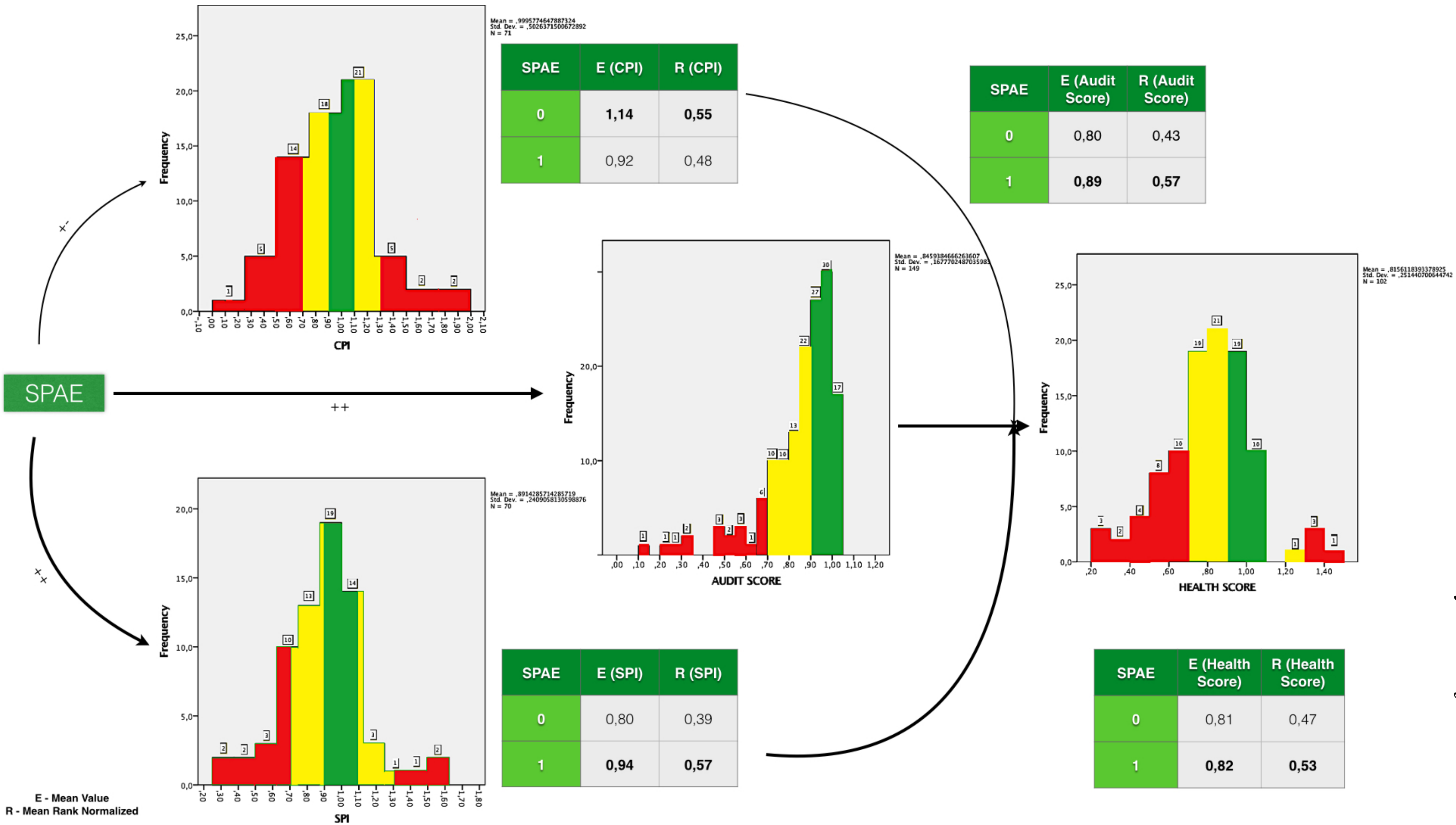
4.5.4 Research Answers

As conclusion, previous Research Questions (RQ) 4.5.1 must be answered.

1. **Answer to RQ1:** Does the SPAE presence have a positive influence on the SPI?
   (a) According to One-way ANOVA approach (Figure 4.6) there is **a significant statistically difference** of the SPI mean between both groups (with/without SPAE).
   (b) According to Figure 4.14, the SPAE presence has a **positive influence** over SPI.
   (c) According to Mann-Whitney U Test, available in Figure 4.14, SPAE presence has a **positive influence** over SPI, since the mean rank of its values is higher.
2. **Answer to RQ2:** Does the SPAE presence have a positive influence on the CPI?
   (a) According to One-way ANOVA approach (Figure 4.7) there is **no statistically significant differences** of the CPI mean between both groups (with/without SPAE).
   (b) According to Figure 4.15, the SPAE presence has a **negative influence** of CPI means between both groups.
   (c) According to Mann-Whitney U Test, available in Figure 4.15, the SPAE impact **is not statistically significant** over CPI, since the mean rank of its values is lower.
3. **Answer to RQ3:** Does the SPAE presence have a positive influence on the Health Score?
   (a) According to Mann-Whitney U Test, available in Figure 4.16, SPAE presence has a **negative influence** of Health Score, means between both groups, since the mean rank of its values is lower.
4. **Answer to RQ4:** Does the SPAE presence have a positive influence on the Audit Score?
   (a) According to One-way ANOVA approach (Figure 4.8) there is a **statistically significant difference** of the Audit Score mean between both groups (with/without SPAE).
   (b) According to Figure 4.13, the SPAE presence has a **positive influence** over the KPI "Audit Score".
   (c) According to Mann-Whitney U Test, available in Figure 4.12, SPAE presence has a **positive influence** over SPI, since the mean rank of its values is higher.

4.6 Summary

Subsequent Figure 4.17 summarizes the factor (SPAE) influence over each one of the performance indicator in the following way:



SPAE	E (CPI)	R (CPI)
0	**1,14**	**0,55**
1	0,92	0,48

SPAE	E (Audit Score)	R (Audit Score)
0	0,80	0,43
1	**0,89**	**0,57**

SPAE	E (SPI)	R (SPI)
0	0,80	0,39
1	**0,94**	**0,57**

SPAE	E (Health Score)	R (Health Score)
0	0,81	0,47
1	**0,82**	**0,53**

Figure 4.17: Summary Analysis developed for Process Performance Indicators used at Critical Software.

SPAE is the main factor in this diagram, with a positive or negative influence over some key performance indicators (KPI's) selected: Cost Performance Index (CPI), Schedule Performance Index (SPI), Health and Audit Score. Charts for each of them are represented in this figure with a clear notation: color semaphore. This color notation represents the different stages that values can fall into:

1. Green: this is the most strict semaphore. Critical Software demands that a good value must be above 0.90! For calculation purposes and only considering this as performance indicators (not as actually loss or gain of money - CPI - for example) it was considered green values must be between 0.90 and 1.10, with exception of Audit Score, which maximum value is 1.0;

2. Yellow: for this color and the next one, ProcessPAIR method was implemented so, 1/3 of the next range values starting on green, are painted yellow;

3. Red: lastly, red bars in the charts are in 2/3 range values.

Above semaphores depiction was also based in ProcessPAIR method of 1/3 implementation.

Besides presentation of the graphics an extra summary evaluation was performed in tabular form: each chart has a table associated to it with Mean Value (represented by E) and Mean Rank Normalized (represented by R) and calculated as the mean rank divided by total N for each KPI. So, for SPAE presence (1) and absence (0) each value is presented and highlighted in bold the most convenient and good one, according to the standards.

It should also be noted that each of the the arrows coming from SPAE factor has a marker in it. Two plus (++) and a large arrow for a more strong influence and one plus and a negative (+-) combined with a skinny arrow representing a less strong influence from factor to KPI.

Chapter 5

Analysis of Code Reviews Data

In this chapter it is analyzed a second dataset provided by Critical Software related with code reviews.

Since CSW set of projects within datasets provided had two different approaches to follow, this Chapter is related only to this dataset and its analysis starting with its description and preparation, followed by the main code reviews evaluation. In this evaluation, a technique of statistical process control is applied in order to assess the current capability and stabilization of the process.

5.1 Description and Preparation of the Dataset

The second dataset provided by Critical Software covered another area of interest: performance in code reviews. The following items describe each column of the dataset:

- *Code Review Record*: ID for the code review performed, according to the format "CRR-xxx", where "xxx" represents the record number;
- *Code Review LOC*: number of Lines of Code (LOC) reviewed;
- *Code Review Defects*: number of defects found during the code review;
- *Defect Detection Rate (DDR) in code reviews*: rate obtained by the division of Code Review Defects by Code Review LOC;
- *Year*: year of the record from 2009 to 2014;
- *Area*: area of the project being evaluated;
- $\log_{10}$*(DDR)*: logarithm of DDR;
- *Average_A*: average value calculated based on 2012 data: baseline value;
- *UCL_2012*: Upper Control Limit (UCL) defined according to 2012 data.

For this dataset there were 197 entries and all of them were valid, without any value missing, so any frequency table (similar to first dataset) for this dataset is shown due to no missing data.

Contrarily to the first dataset, data preparation for this one was simplified. Only required one workaround into the worksheet importation to SPSS that contained cells with equations, the VLOOKUP function and cell references to other sheets in the workbook. The solution was to open the workbook in Excel, create a new sheet and Paste Special (to paste only the values) all the values that we want to analyze in here [82].

5.2 Descriptive Statistics

For the second dataset the kind of evaluation performed is not similar to the first one, due to the nature of data. Table 5.1 shows the main fields used in this analysis and respective descriptive statistics.

Table 5.1: Summary Table of descriptive statistics of main fields from 2nd dataset.

Field	N	Min.	Max.	Mean	Std. Deviation
Code Review LOC	197	24	8465	702,76	1039,762
Code Review Defects	197	1	240	23,31	30,275
DDR	197	,629	402,985	57,043	70,853
$\log_{10}$(DDR)	197	-,201397	2,605	1,489	,511

All the fields are present and have valid values for the 197 entries. DDR is calculated through Code Review Defects divided by Code Review LOC but it is also dependent from the type of review performed [45] so mean value cannot be very conclusive. On the other hand, regarding code review LOCs and Defects the maximum values for these fields can be very high (8465 and 240, respectively). According to "Kamsties and Lott" the time to find a defect is dependent on the subject [48], but "Basili and Selby", instead, claim that the fault rate is dependent on the software under study, and that the defect detection rate is unrelated to tester experience [88]. Also, "Wood et al" refer that defect detection rate depends on the type of faults in the program [96], [97]. All of these studies show that factors other than the test case design technique or type of code review can have significant effects on the testing results [45] and consequently on review.

5.3 Code Reviews Evaluation

This section refers to a code review evaluation performed using SPSS for the whole analysis.

The code review model of Critical Software was first developed in 2009. Based on the code review speed baseline and code review defect detection rate it was establish a correlation between them. Being the review speed easy to control it was considered as an input variable and defect detection rate is an effect resulting from the review speed. By correlating these two variables, a logarithmic curve was defined and the first statistical Process Model was developed at CSW [27]. After this, a baseline based on 2012 data was defined as the best achievement to compare to. Next subsection 5.3.3 a comparison with this baseline is performed. However, in this dataset we did not have time/effort data, so the review speed cannot be computed, only the DDR.

5.3.1 Adherence to Log-Normal Distribution

Many variables in software engineering (defect density, function size, etc.) that exhibit a large number of small values and a long tail of high values, can be approximated by a log-normal distribution [61], [62].

A log-normal distribution is a continuous probability distribution of a random variable whose logarithm is normally distributed. Thus, if the random variable X is log-normally distributed, then Y = ln(X) or $\log_{10}$(X) has a normal distribution. The log-normal distribution is used to model continuous random quantities when the distribution is believed to be skewed [59].

Before adherence to this distribution, in Figure 5.1 can be visualized the type of chart produced before logarithmic transformation of DDR.

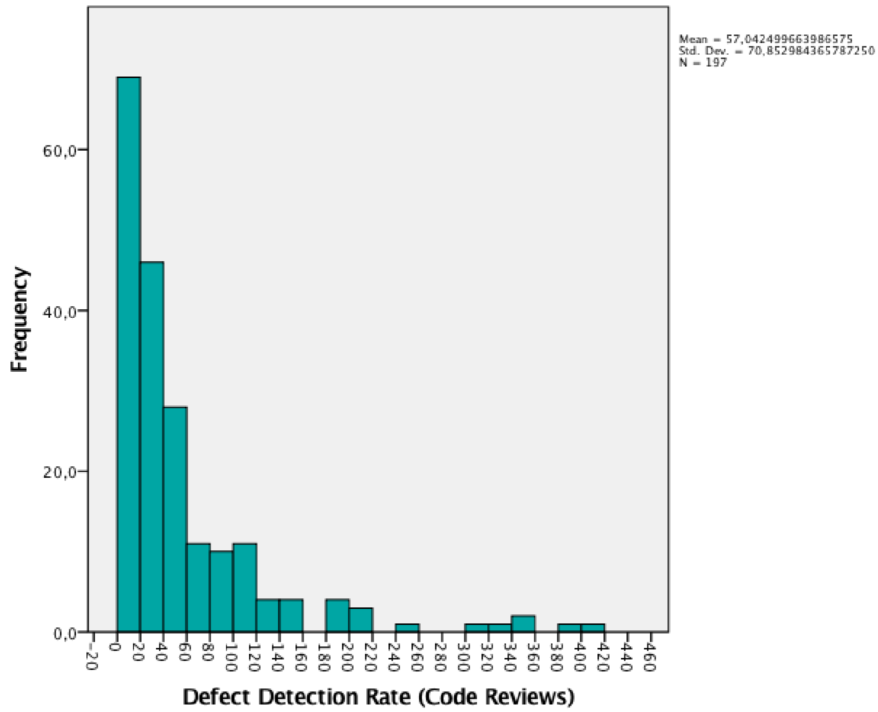


Figure 5.1: Frequency Chart of DDR before logarithmic transformation.

To test the adherence to the log-normal distribution, two hypothesis are considered in here:

- **Null Hypothesis** (*H0*): there is no significant departure from normality.
- **Alternative Hypothesis** (*H1*): there is a significant departure from normality.

In the first place, descriptive statistics for $\log_{10}$(DDR) were calculated and presented in Table 5.2.

Table 5.2: Table of descriptive values for $\log_{10}$(DDR).

			Statistic	Std. Error
$\log_{10}$***(DDR)***	Mean		1,489	,364
	Median		1,52	
	Variance		,262	
	Std. Deviation		,511	
	Range		2,807	
	Skewness		-,299	,173
	Kurtosis		,161	,345
	Interquartile Range		,679	
	95% Confidence Interval for Mean	Lower Bound	1,417	
		Upper Bound	1,560	

Adherence to a log-normal distribution in SPSS generates not only the previous table but as well two tests of Normality: Kolmogorov-Smirnov and Shapiro-Wilk (Table 5.3). Since the Shapiro-Wilk test is advised to use when the evaluation is for less than 200 entries, Kolmogorov-Smirnov was discarded.

Table 5.3: Tests of Normality for $\log_{10}$(DDR).

Kolmogorov-Smirnov			**Shapiro-Wilk**		
Statistic	df	Sig.	Statistic	df	Sig.
,77	197	,006	,990	197	,186

The Shapiro-Wilks Test is a statistical test of the hypothesis that sample data have been drawn from a normally distributed population. From this test, the Sig. (p) value is compared to the a priori alpha level (level of significance for the statistic) and a determination is made as to reject (p <alpha) or retain (p >alpha) the null hypothesis [40]. Sig. value is 0.186 for this test so the null hypothesis can be rejected.

As the Shapiro-Wilks test is rather conservative, many references agree that the Shapiro-Wilks Test should not be the sole determination of normality. So, assessment of normality with an examination of skewness and kurtosis, histogram chart or box plots and an evaluation of the Normal Q-Q Plots should supplement this analysis (Figure 5.3).

Starting with skewness and kurtosis, the reference value should be above +3.29. In this case is a concern because according to Table 5.2 both values are way below this mark.

As for the histogram chart, Figure 5.2 shows an approximation to normal distribution.

And finally, examination of Normal Q-Q (Quartile-Quartile) Plot - Figure 5.3 - shows a really excellent linear relationship.

Retaining the null hypothesis (*H0*) indicates that the assumption of normality has been met for the given sample.

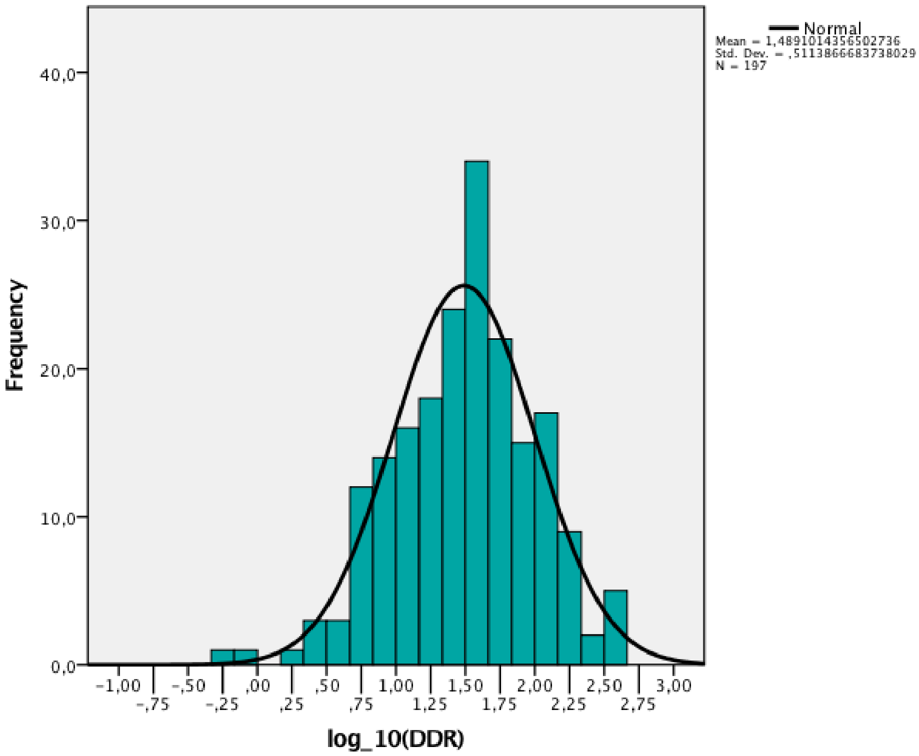


Figure 5.2: Frequency Chart for $\log_{10}$(DDR) accepting a normal distribution with a normal distribution curve.

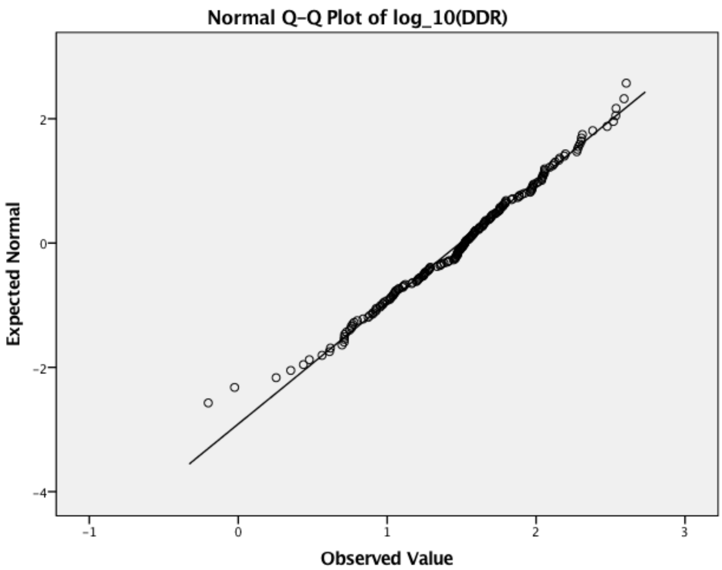


Figure 5.3: Normal Q-Q plot of $\log_{10}$(DDR)

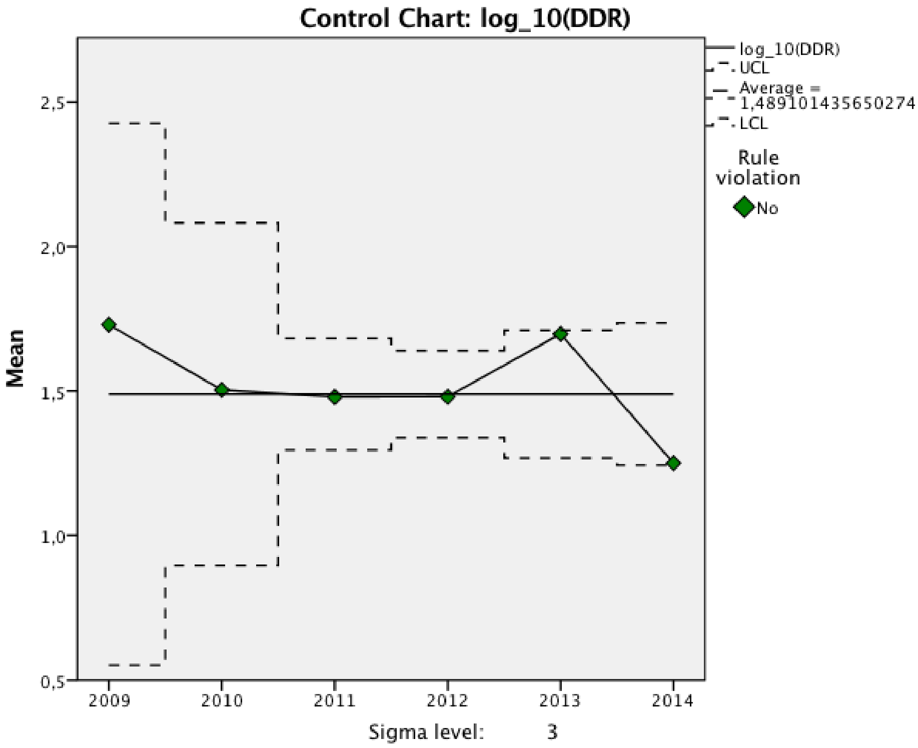


Figure 5.4: Control Chart of mean $\log_{10}$(DDR) from 2009 to 2014.

5.3.2 Control Chart

As the data assumes a Log-normal distribution, a control chart (for the mean along the years) was made after the transformation, as can be seen in Figure 5.4.

In Statistical Process Control (SPC), control charts are commonly used to graphically represent the behavior over time of variables that characterize process performance and that help assessing process stability and capability (also refer to Chapter 2, Section 2.3.2.1). Stability refers to the level of variability in the variable under analysis and capability to the ability to meet desired performance levels. To assess if a process is under control (or stable) assessment of two control limits is conducted: for upper control limit (UCL) and lower control limit (LCL).

Control limits are derived based on the standard deviation computed from historical process data. The area between each control limit and the centerline is divided into zones. The closest zone to the centerline is referred to as Zone A (1-sigma zone), the next is referred to as Zone B (2-sigma zone), and the third one is referred to as the control limit Zone C (3-sigma zone). The standard UCL and LCL are usually chosen as the 3-sigma limits and this was also the case in this analysis [70].

Using the measurements of the sampled units and based on the statistical sampling theory, it is said ([2]) that at a certain level almost all items (99.7%) will fall in approximately 3 standard errors (standard deviation divided by squared number of sampled items) from the target value (the mean of all measurements). In this case for 197 entries according to this theory should have a

0.591 standard deviation. Consulting graphic in above subsection 5.3.1 (Figure 5.2) the standard deviation is still above the standard 0.3%: 0.511.

A control chart (which implement the sampling theory) assumes great importance in here, to test if there are measurements outside these limits ("control limits") and if so, act in order to avoid going out of these limits (the specification limits are defined by the quality standards considering the natural variation, not caused by and special events) [2]. When such out of control limit measurements are found, it is a warning that may be there are special (non-natural) causes influencing the production process that may lead to a low quality production [55].

For this control chart (Figure 5.4) there is a clear stabilization from 2010 until 2012 with an average mean of 1.49 approximately. There is no rule violation, although the mean value in 2013 is almost exceeding mean value and control limit associated. Control limits shown are variable since the sample sizes varied through the collection period (these limits represent the amount of variation) [52].

To conclude, it can be said that the process is stable because control limits (derived from the process performance data) are within the specification limits.

5.3.3 Baseline Comparison

Before adherence to normal distribution (next Subsection 5.3.1) evaluation, a baseline comparison is accomplished with 2012 UCL definition and average value. The UCL baseline value for 2012 is 142, but a comparison with values without logarithmic transformation must also be performed as follows:

Baseline Mean Value:

$$\overline{X} = 44$$

Sample Mean Value:

$$\overline{X} = 57$$

The DDR follows approximately a log-normal distribution with parameters: $\log_{10}(\text{DDR}) \approx \text{N}$ (μ=1.489, σ^2=0.262)

5.4 Summary

The control chart performed without establishing any previous UCL and LCL values laid out a stable process but almost passing by the current limits.

Add in the fact that the baseline comparison stand out the possible need for a new baseline, the 2012 values should be revised.

The baseline comparison performed can outline a better revision quality (more defects were detected) and a great capacity to detect defects or even the poor quality being reviewed. The best option after these conclusions, would be the assessment of this analysis with more information like review speed (in order to analyze this phenomena) or number of defect detected in a posterior phase of the projects (it would allow the assumption of how many defects leaked from the reviews performed).

Chapter 6

Conclusions and Future Work

It is a certainty that the quality entails several costs: prevention costs, defect detection (appraisal) costs and internal and external failure costs [60]; but is also a certainty that defect prevention and early defect prevention investments pay off, especially if implemented in continuous and organized processes and early in the projects, in order to prevent later detection of defects, at a later stage of a project, which results in much more expensive fix price (comparing to an earlier defect detection).

Many companies within Software Engineering are adopting some techniques, process or models/frameworks, like CMMI, in order to have, for example, a good balance between the costs and the quality implemented in their software projects, establishing methodical processes, with goals and practices that should be followed.

This type of process improvement and set of best practices can be facilitated with their automation through auxiliary tools and a new "niche" such as mining software repositories.

After analysis of the state of the art, namely the ProcessPAIR method and tool and general concepts of analysis and improvement of process performance, the next step was the assessment of the existing metrics in CSW and the needs and opportunities of application of quantitative methods, in order to be able to analyze and assess some improvements of performance in software development processes and projects.

Therefore, definition of a model of performance indicators (including top-level indicators and more detailed indicators and factors) tailored to needs through data collected from CSW projects was the main accomplishment, in line with what was defined in the early stages.

Data is most useful when collected systematically through time, because it can be ascertain whether the process is behaving consistently through time (is in "statistical control") and it can be assessed the success of the efforts to improve it. Critical Software provided two different datasets with data collected through some years and statistical analysis was conducted in SPSS in order to perform a full causal analysis for the different performance indicators. This tool was not directly used in the phase of model derivation, but some of its underlying techniques for prediction (causal analysis) and classification (problem identification) were used successfully.

For the first dataset hypothesis testing was performed: an one-way between groups analysis of variance (ANOVA) was conducted to explore the impact of SPAE presence/absence on the different process performance indicators.

Regarding the SPI, the p-value is greater than 0.05, so there is no statistically significant difference between averages mean for the groups. For the Audit Score the p-value (Sig.) is equal to zero, meaning that the null hypothesis (H0) is rejected (population means are equal).

Can be concluded for CPI that there are statistically significant differences between group means, comparing with SPAE. Also, in Mann-Whitney U Test similar to One-way ANOVA test, this performance indicator presents again suspicious results.

In Mann-Whitney U Test it was decided to also present in this section the results relative to Health Score performance indicator. Because it is a composite indicator, the decision to retain the null hypothesis was again carried out and it was also concluded that SPAE has just a low magnitude effect in Health Score.

For SPI the decision was the same as for Audit score: to reject the null hypothesis - they have identical distribution to SPAE.

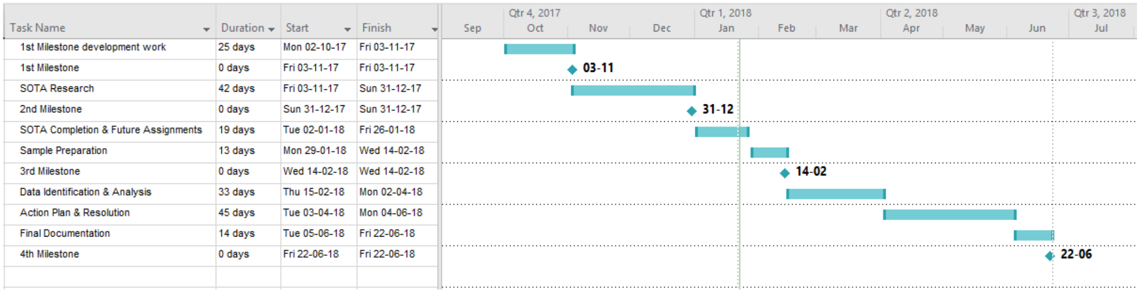
The second dataset, related with code reviews, refers to an important task in CSW. A baseline comparison is carried out, as well as adherence to log-normal distribution is reviewed, leading to retain the null hypothesis, indicating the assumption of log-normality for the given samples. A control chart was designed in order to evaluate the capability of the process (only possible because they adhere to a normal distribution). It was concluded that current process is capable because control limits are within specification limits.

In general, the application of a defined and calibrated model to automatically analyze the performance (problems and causes) of projects, teams or departments of the CSW was the main goal of this Master Thesis. Afterwards, a critical review of the quality and usefulness of the results obtained is an important issue. This whole analysis allowed Critical Software a more conscious understanding of the impact of SPAE within projects, being clear that the relationship with CPI is not the best, due to an additional cost for the projects, for example. For Health Score the SPAE presence has only a little impact due to the fact of being a composite indicator. Besides this, a positive influence over SPI and Audit Score was evidenced.

The application of ProcessPAIR as a method providing a range of values, allows the evaluation and assessment of a more knowledgeable projects performance and a more informed support for decision making on performance objectives.

Next Figure 6.1 illustrates the four major milestones outlined for steps and work defined within this Master Thesis: first public presentation of the Master Thesis planning; second milestone - SOTA research (first) delivery; third milestone - named "dissertation summit", a public presentation of work developed at the moment and future assignments; finally, fourth milestone - thesis delivery.

To conclude, a comparison with predefined objectives in Section 1.1.3 was performed as follows:



Task Name	Duration	Start	Finish
1st Milestone development work	25 days	Mon 02-10-17	Fri 03-11-17
1st Milestone	0 days	Fri 03-11-17	Fri 03-11-17
SOTA Research	42 days	Fri 03-11-17	Sun 31-12-17
2nd Milestone	0 days	Sun 31-12-17	Sun 31-12-17
SOTA Completion & Future Assignments	19 days	Tue 02-01-18	Fri 26-01-18
Sample Preparation	13 days	Mon 29-01-18	Wed 14-02-18
3rd Milestone	0 days	Wed 14-02-18	Wed 14-02-18
Data Identification & Analysis	33 days	Thu 15-02-18	Mon 02-04-18
Action Plan & Resolution	45 days	Tue 03-04-18	Mon 04-06-18
Final Documentation	14 days	Tue 05-06-18	Fri 22-06-18
4th Milestone	0 days	Fri 22-06-18	Fri 22-06-18

Figure 6.1: Gantt Diagram.

1. **Definition** of relevant and feasible top-level performance indicators for CSW, also definition of relevant and more detailed indicators and factors that affect the highest level indicators, to support causal analysis - Chapter 4 was the main focus in this Master Thesis with this definition. Chapter 5 dwell on other type of performance indicators, related with code reviews.

2. **Calibration** of ranges of values for the classification of individual indicators in "semaphores". As well as calibration of regression models for cause-and-effect relationships between indicators (based on CSW data sets, business objectives, industry benchmarks, literature, among others) - calibration performed for main KPIs in this Master Thesis: CPI, SPI, Health and Audit Score.

Some objectives defined previously, unfortunately, were not achieved due to the type of data provided by Critical Software, which was incompatible with ProcessPAIR. This tool allows the analysis of continuous variables and the main independent variable for CSW is SPAE, a discrete variable. Without this feature from ProcessPAIR only the causal analysis, the relationship between the indicators was performed being the next step the automation. Therefore, some future work was defined and it is be described next.

Future Work

The development of an automatic calibration model based on data in the CSW repositories is the next target to achieve after this Master Thesis.

Also, development of a plug-in for ProcessPAIR tool, in Java, with the codification of the previous model and an adapter for reading data from the CSW repositories would be a useful goal for a future work to take place.

An important step is as well, the adaption of ProcessPAIR tool to support discrete variables with the incorporation of whole analysis performed into information systems.

Appendix A

SPSS Data

This Appendix shows additional charts and tables generated with SPSS.

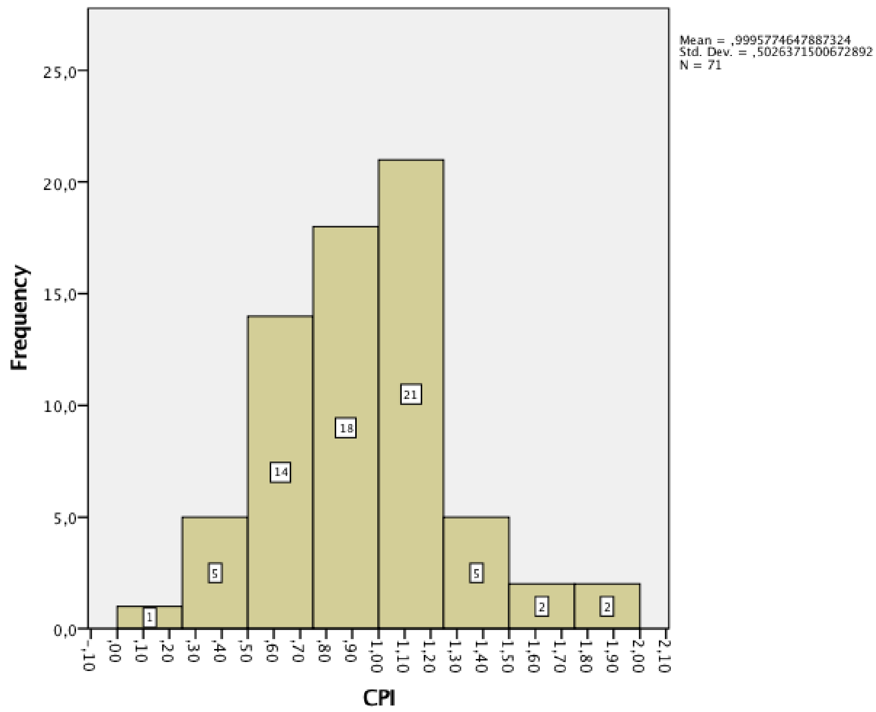


Figure A.1: Frequency Graph for CPI.

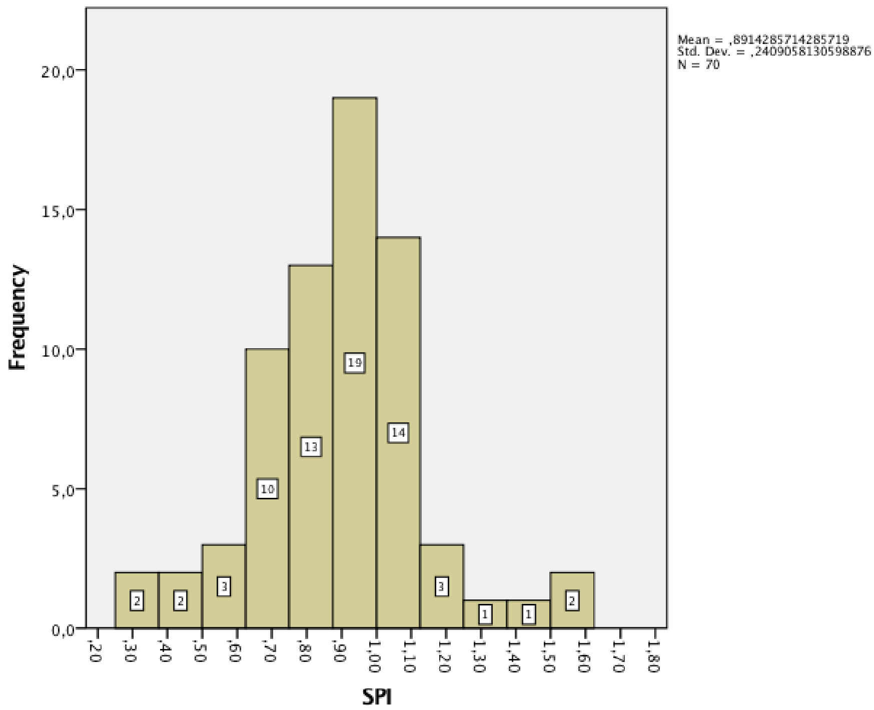


Figure A.2: Frequency Graph for SPI.

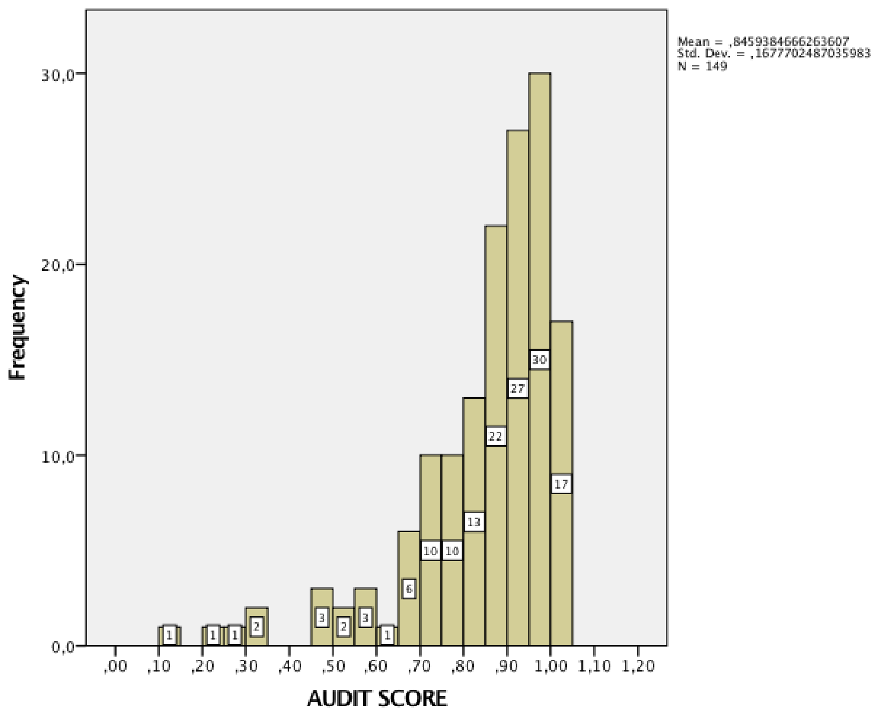


Figure A.3: Frequency Graph for Audit Score.

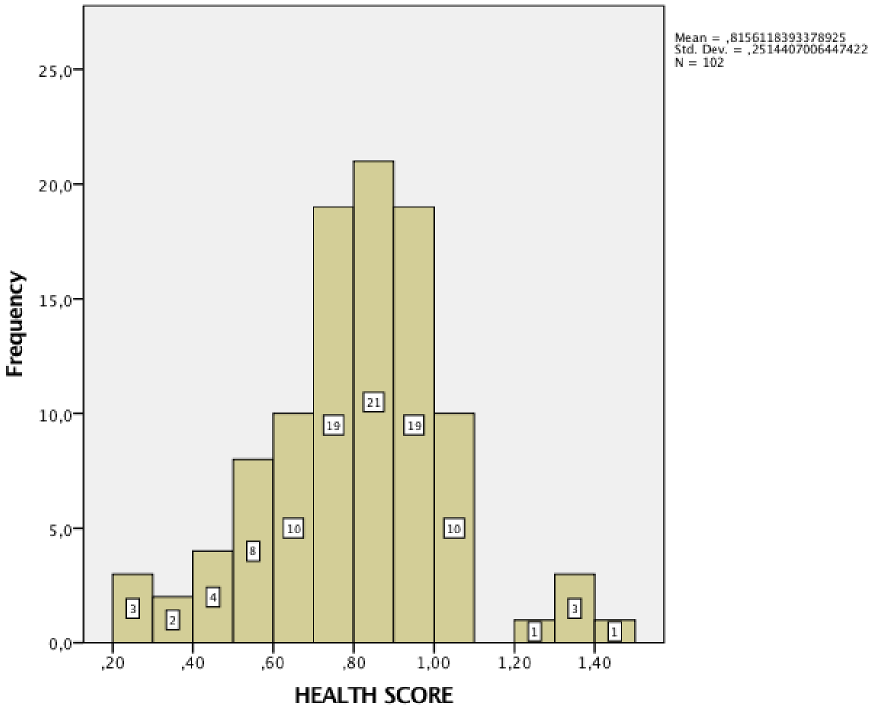


Figure A.4: Frequency Graph for Health Score.

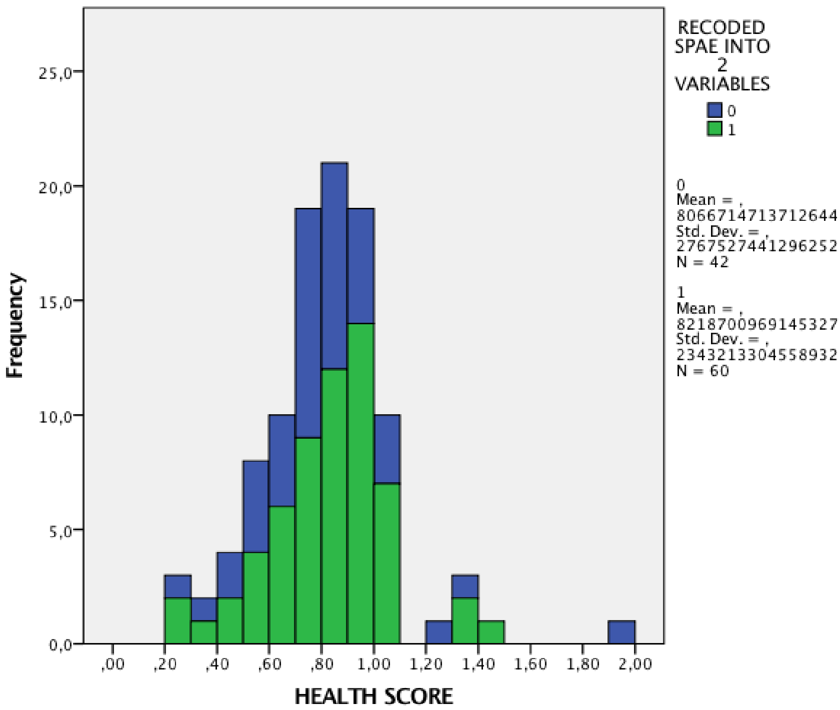


Figure A.5: Frequency Graph for Health Score vs SPAE.

Oneway

Descriptives

HEALTH SCORE

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
0	42	,806671471	,276752744	,042703876	,720429245	,892913698	,280000000	1,93227273
1	60	,821870097	,234321330	,030250754	,761338479	,882401715	,214793417	1,45296296
Total	102	,815611839	,251440701	,024896339	,766224199	,864999479	,214793417	1,93227273

Test of Homogeneity of Variances

HEALTH SCORE

Levene Statistic	df1	df2	Sig.
,108	1	100	,743

ANOVA

HEALTH SCORE

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	,006	1	,006	,089	,765
Within Groups	6,380	100	,064		
Total	6,385	101			

Robust Tests of Equality of Means

HEALTH SCORE

	Statistic[a]	df1	df2	Sig.
Welch	,084	1	78,701	,772
Brown-Forsythe	,084	1	78,701	,772

a. Asymptotically F distributed.

Figure A.6: SPSS Tables for Health Score One Way ANOVA Procedure.

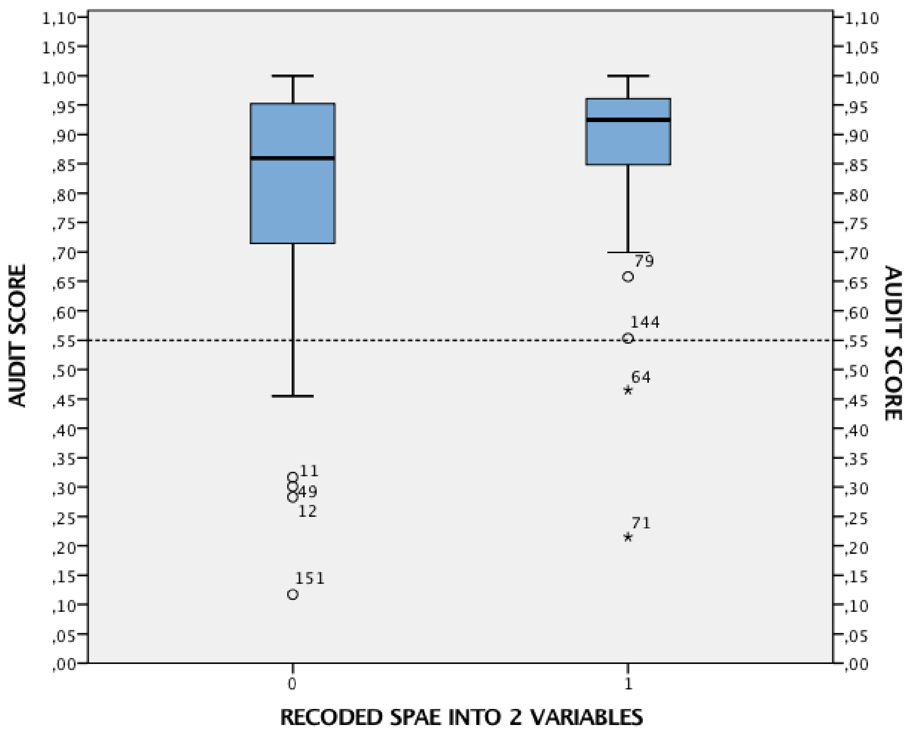


Figure A.7: Boxplot for Audit Score vs SPAE discrete variable.

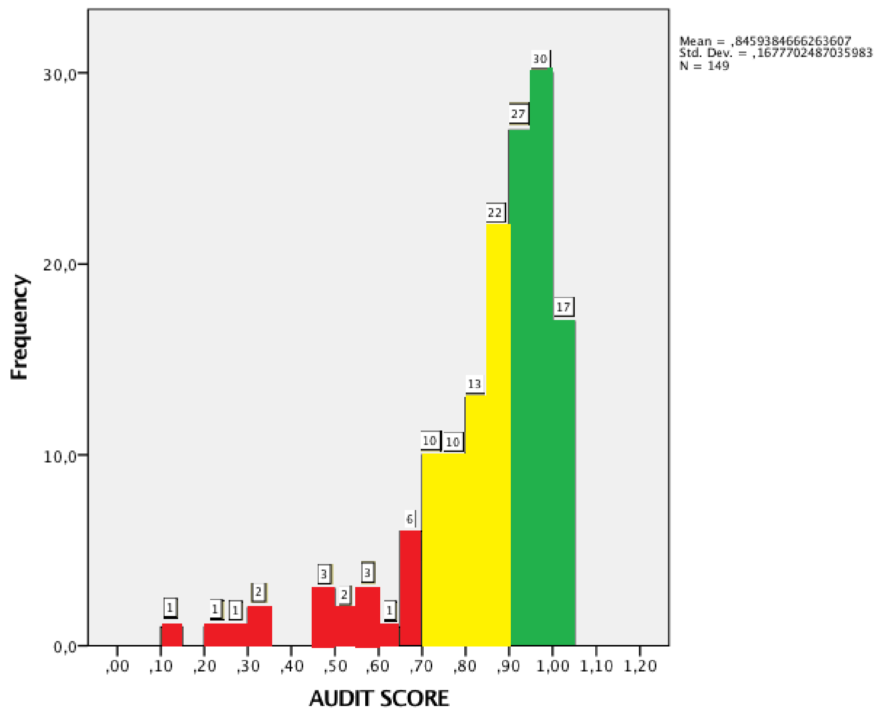


Figure A.8: Semaphore Graph for Audit Score.

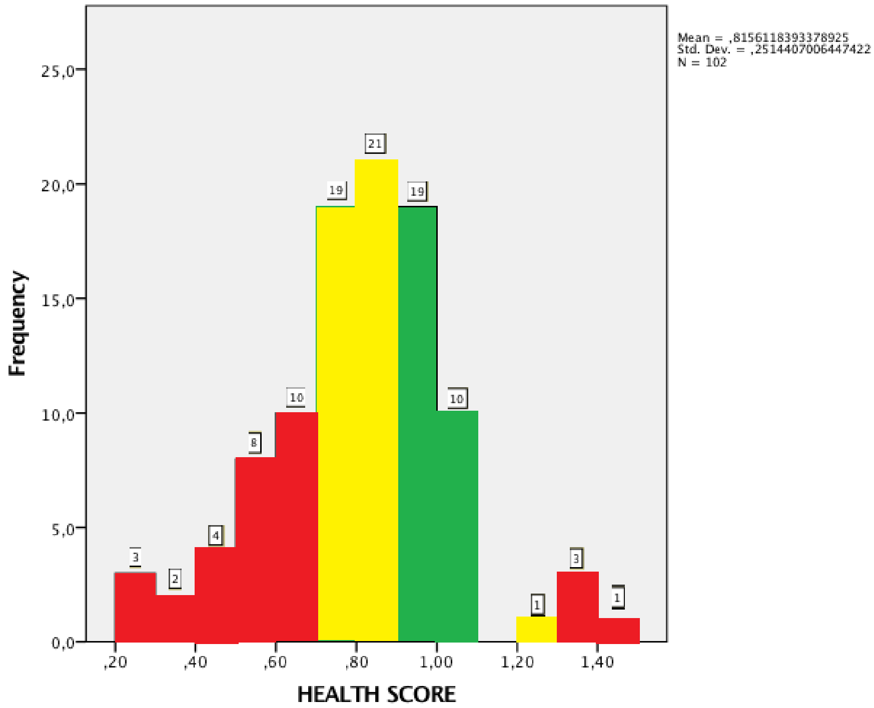


Figure A.9: Semaphore Graph for Health Score.

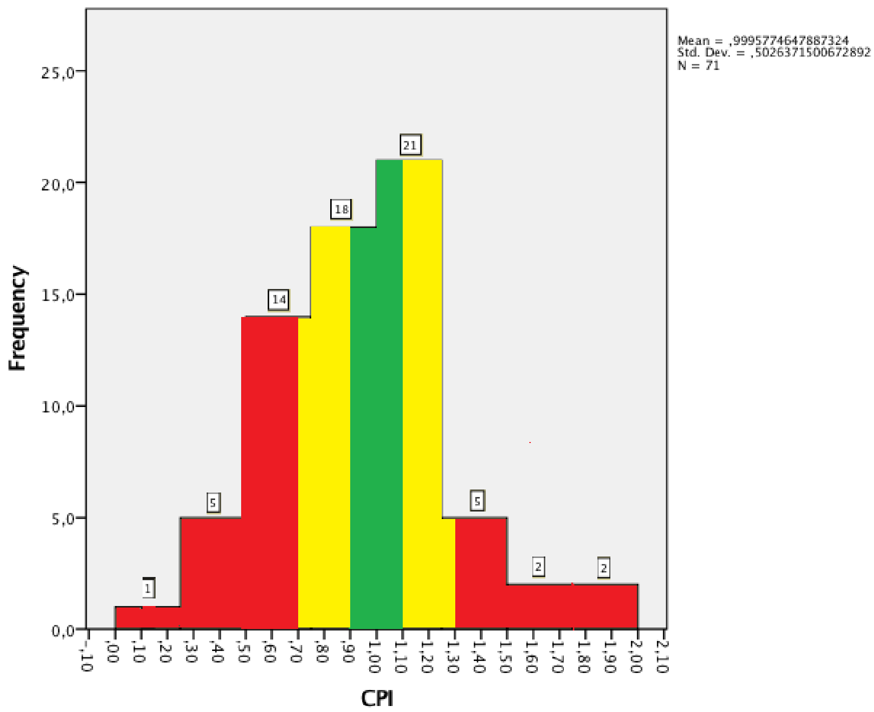


Figure A.10: Semaphore Graph for CPI.

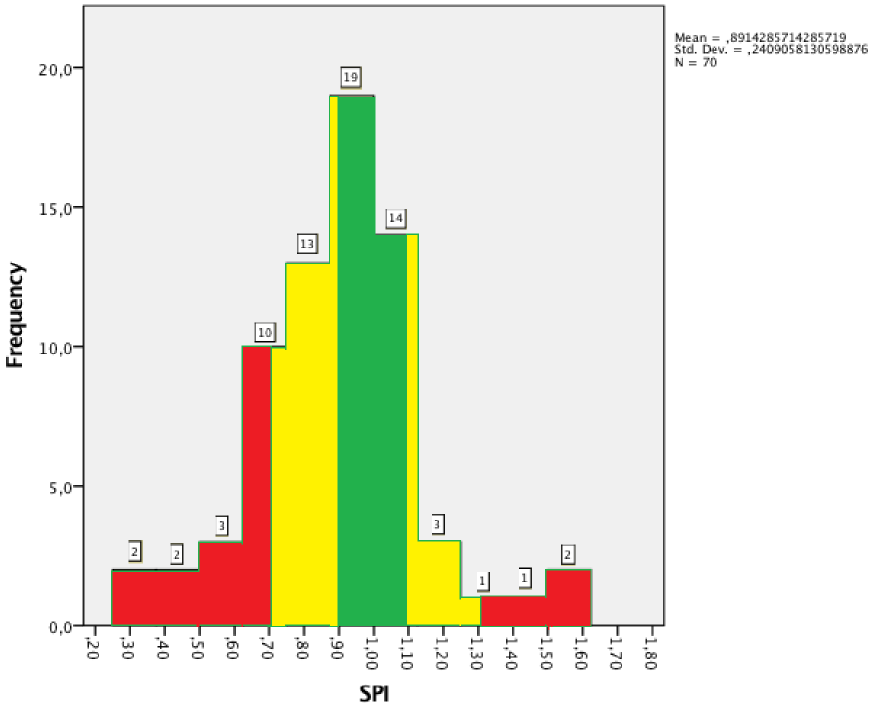


Figure A.11: Semaphore Graph for SPI.

References

[1] In John S. Oakland, editor, *Statistical Process Control*, page chapter 1. Butterworth-Heinemann, Oxford, sixth edition, 2008.

[2] *Using SPSS for Process Quality Control - A Critical Review*, Sofia, Bulgaria, February 2014.

[3] Braselina; Gomes Tiago et. al Almeida, Délio; Sousa. Quality management. Critical Software Internal Documentation, september 2017.

[4] Ricardo; Sousa Pedro et. al Almeida, Délio; Jesus. Measurement and analysis. Critical Software Internal Documentation, february 2018.

[5] OLSPS Analytics. Statistical analysis software: Spss vs excel. Online reference available at: http://www.olspsanalytics.com/statistical-analysis-software-spss-vs-excel/. Last access at May 2018., 05 2018.

[6] Jiju Antony and Craig Fergusson. Six sigma in the software industry: Results from a pilot study. 19:1025–1032, october 2004.

[7] J Niehaves Becker, Jens Poeppelbuss, and A. Simons. Maturity models in i.s. research. In *18th European Conference on Information Systems, ECIS 2010*, page 12, january 2010.

[8] Jörg Becker, Ralf Knackstedt, and Jens Pöppelbuß. Developing maturity models for it management. *Business & Information Systems Engineering*, 1(3):213–222, Jun 2009.

[9] Lee A. Becker. Effect sizes (es). Universitat de València - Online Publication, 2000.

[10] Shruti Bhat. 7 important tools to identify a 'quality' problem. september 2014.

[11] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[12] Daniel Boduszek. One-way between groups analysis of variance (anova). University Lecture. Last access at July 15th, 2018., November 2013.

[13] Tonia De Bruin and Michael Rosemann. Towards a business process management maturity model. In D Bartmann, F Rajola, J Kallinikos, D Avison, R Winter, P Ein-Dor, J Becker, F Bodendorf, and C Weinhardt, editors, *ECIS 2005 Proceedings of the Thirteenth European Conference on Information Systems*, pages 1–12, Germany, Regensburg, 2005. Verlag and the London School of Economics.

[14] D. N. Card. Learning from our mistakes with defect causal analysis. *IEEE Software*, 15(1):56–63, Jan 1998.

[15] David Card. Myths and strategies of defect causal analysis. In *Proceedings: Pacific Northwest Software Quality Conference*, november 2006.

[16] Barnard College Empirical Reasoning Center. Spss tutorial - hypothesis testing: Anova tests. Online Article. Available at: https://erc.barnard.edu/spss. Last access at June 2018., 2018.

[17] Tse-Hsun Chen, Stephen W. Thomas, and Ahmed E. Hassan. A survey on the use of topic models when mining software repositories. *Empirical Software Engineering*, 21(5):1843–1919, Oct 2016.

[18] Dale Childs and Paul Kimmerly. Process performance: Words of wisdom. Technical report, USMC-TSO Kansas City, july/august 2012. The end of the PC.

[19] Mary Beth Chrissis, Michael Konrad, and Sandra Shrum. *CMMI for Development: Guidelines for Process Integration and Product Improvement*. Addison-Wesley Professional, third edition, 2011.

[20] C.J.Wild and G.A.F. Seber. Control charts. In Wiley, editor, *Chance Encounters: A First Course in Data Analysis and Inference*, page 207, New York, 2000.

[21] P.M. Clarke, R.V. O'Connor, T. Rout, and A. Dorling. *Software Process Improvement and Capability Determination: 16th International Conference, SPICE 2016, Dublin, Ireland, June 9-10, 2016, Proceedings*. Communications in Computer and Information Science. Springer International Publishing, 2016.

[22] Reidar Conradi and Fuggetta. Improving software process improvement. *IEEE Software*, pages 2–9, july-august 2002.

[23] Universidade de Coimbra. Análise de variância: Análise de variância com um factor e efeitos fixos. Online notes from Universidade de Coimbra - 1º Curso Intensivo de Estatística Computacional. Last access at July 2018., october 2007.

[24] Universidade de Coimbra. Testes de localização: Duas amostras independentes: teste u de mann-whitney. Online notes from Universidade de Coimbra - 1º Curso Intensivo de Estatística Computacional. Last access at July 2018., october 2007.

[25] Marcelo Santos de Mello. Melhoria de processos de software multi-modelos baseada nos modelos mps e cmmi-dev. Master's thesis, Universidade Federal do Rio de Janeiro, Rio de Janeiro, March 2011.

[26] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[27] CSW Quality Department. Critical software "cmmi book": Report on cmmi benefits at critical software. Technical report, Critical Software, january 2015.

[28] SAGE Edge. In *Hypothesis Testing: Significance, Effect Size and Power.*, pages 240–268. SAGE Publications, Inc., 2nd edition, 2018.

[29] K. Emam and D. R. Goldenson. An empirical review of software process assessments. Technical report, National Research Council Canada, Institute for Information Tech., Canada, 1999. NRC/ERB-1065 (NRC 43610).

[30] Brian S. Everitt and Sabine Landau. *A Handbook of Statistical Analyses Using SPSS*. CRC Press, Inc., Boca Raton, FL, USA, 2003.

[31] year = 2018 url = https://dei.fe.up.pt/sergup/ urldate = 2018-06-03 Faculdade de Engenharia da Universidade do Porto, title = Software Engineering Research Group.

[32] Mohamed E. Fayad, Mauri Laitinen, and Robert P. Ward. Thinking objectively: Software engineering in the small. *Commun. ACM*, 43(3):115–118, March 2000.

[33] A. Ferreira and G. et al Santos. Iso 9001:2000, mps.br nível f e cmmi nível 3: Uma estratégia de melhoria de processos na bl informática. In *Simpósio Brasileiro de Qualidade de Software 2006*, 2006.

[34] Eileen Forrester, Brandon Buteau, and Sandra Shrum. *CMMI for Services: Guidelines for Superior Service*. Addison-Wesley Professional, second edition, 2011.

[35] The R Foundation. The r project for statistical computing. R Foundation Website. Available at https://www.r-project.org/, june 2018.

[36] Research Gate. What is the assumption for brown forsythe test and welch test? Online Article. Available at: https://www.researchgate.net/post/what_is_the_assumption_for_Brown_Forsythe_test_and_Welch_test. Last access June 2018., 2015.

[37] Ashok Gunasekaran. Quality excellence: An attempt to change the world - cmmi process areas, 2008.

[38] A. E. Hassan. Mining software repositories to assist developers and support managers. In *22nd IEEE International Conference on Software Maintenance*, pages 339–342, Sept 2006.

[39] D. Ho, A. Kumar, and N. Shiwakoti. Maturity model for supply chain collaboration: Cmmi approach. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 845–849, Dec 2016.

[40] Robert A. Horn. Understanding the one-way anova. In *Educational Psychology 525: Introduction to Statistics*. Northern Arizona University, 2009.

[41] IBM. Bm spss software. IBM Website. Available at https://www.ibm.com/analytics/data-science/predictive-analytics/spss-statistical-software, may 2018.

[42] IBM. Bm spss statistics. IBM Website. Available at https://www.ibm.com/uk-en/marketplace/spss-statistics. Last access at April 2018., may 2018.

[43] Software Engineering Institute, editor. *Exactly What are Process Performance Models in the CMMI?* Carnegie Mellon University, june 2007.

[44] Standish Group International. The chaos report. Technical report, 2009. www.standishgroup.com.

[45] J. Itkonen, M. V. Mantyla, and C. Lassenius. Defect detection efficiency: Test case based vs. exploratory testing. In *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, pages 61–70, Sept 2007.

[46] Jakob Iversen and Ojelanki Ngwenyama. Problems in measuring effectiveness in software process improvement: A longitudinal study of organizational change at danske data. 26:30–43, may 2002.

[47] M. Kalinowski, D. N. Card, and G. H. Travassos. Evidence-based guidelines to defect causal analysis. *IEEE Software*, 29(4):16–18, July 2012.

[48] Erik Kamsties and Christopher M. Lott. An empirical evaluation of three defect-detection techniques. In Wilhelm Schäfer and Pere Botella, editors, *Software Engineering — ESEC '95*, pages 362–383, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[49] Sofia Maria Karadimitriou and Ellen Marshall. Mann-whitney u test. Online notes from Statstutor, a Community Project: `http://www.statstutor.ac.uk/`, University of Sheffield. Last access at June 2018., 2018.

[50] Supannika Koolmanojwong. Software engineering ii - introduction, 2017. University of Southern California, Center for Systems and Software Engineering.

[51] Robert J Latino, Kenneth C Latino, and Mark A Latino. *Root cause analysis : improving performance for bottom-line results*. Boca Raton, FL : CRC Press, 4th edition, 2011.

[52] Nick Lee and Mike Peters. *The p chart - Monitoring Defective Items*, chapter 16, pages 517–519. Sage, 2016.

[53] Jeremy Jay V. Lim. Root cause analysis (rca) tools. Linkedin Profile. Available at `https://pt.slideshare.net/jeremyjaylim/root-cause-analysis-rca-tools-24376829`. Last access at July 2018., july 2013.

[54] CMMI Institute LLC. What is capability maturity model integration (cmmi)®?, 2017.

[55] Amitava Mitra. *Fundamentals of Quality Control and Improvement.* John Wiley & Sons, Inc., USA, 4th edition, 2016.

[56] Donald F. Morrison. Multivariate analysis of variance. In *Encyclopedia of Biostatistics*. John Wiley & Sons, Ltd, 2 edition, july 2005.

[57] Robert A. Muenchen. The popularity of data analysis software. *r4stats*, april 2013.

[58] Raghav S. Nandyal. *CMMI: A Framework for Building World-Class Software and Systems Enterprises*. New Delhi : Tata McGraw-Hill, xxxvi edition, 2004.

[59] Péter Nándori. The lognormal distribution. Online publication for Virtual Laboratories. Budapest, Hungary.

[60] ASQ: The Global Voice of Quality. Learn about quality: Cost of quality (coq). Excerpted from The ASQ Quality Improvement Pocket Guide: Basic History, Concepts, Tools, and Relationships, Grace L. Duffy, ASQ Quality Press, 2013, pages 62–65., january 2018.

[61] Y. Park, H. Choi, and J. Baik. A framework for the use of six sigma tools in psp/tsp. In *5th ACIS International Conference on Software Engineering Research, Management Applications (SERA 2007)*, pages 807–814, Aug 2007.

[62] Youngkyu Park, Hyuncheol Park, Hojin Choi, and Jongmoon Baik. A study on the application of six sigma tools to psp/tsp for process improvement. In *5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering,Software Architecture and Reuse (ICIS-COMSAR'06)*, pages 174–179, July 2006.

[63] Mark Paulk, William Curtis, Mary Beth Chrissis, and Charles Weber. Capability maturity model for software (version 1.1). Technical Report CMU/SEI-93-TR-024, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1993.

[64] Mark C. Paulk, Charles V. Weber, Bill Curtis, and Mary Beth Chrissis. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[65] Stasys Peldzius and Saulius Ragaisis. Comparison of maturity levels in cmmi-dev and iso/iec 15504. In *Proceedings of the 2011 American Conference on Applied Mathematics and the 5th WSEAS International Conference on Computer Engineering and Applications*, AMERICAN-MATH'11/CEA'11, pages 117–122, Stevens Point, Wisconsin, USA, 2011. World Scientific and Engineering Academy and Society (WSEAS).

[66] J.D. Perez Gonzalez. Mann-whitney u test for equality of distributions. Online Article at WikiofScience. Last access at May 2018., july 2012.

[67] Marjorie A. Pett. *Nonparametric Statistics for Health Care Research: Statistics for Small Samples and Unusual Distributions*. Sage Publications, Inc., 2 edition, 2016.

[68] Francisco Pino, Felix Garcia, and Mario Piattini. Software process improvement in small and medium software enterprises: A systematic review. 16:237–261, 06 2008.

[69] B. Randell. Software engineering in 1968. In *Proceedings of the 4th International Conference on Software Engineering*, ICSE '79, pages 1–10, Piscataway, NJ, USA, 1979. IEEE Press.

[70] Mushtaq Raza. *Automated Software Process Performance Analysis and Improvement Recommendation*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, Porto, july 2017.

[71] Mushtaq Raza. Processpair: Automated software process performance analysis and improvement recomendation, 2017.

[72] Mushtaq Raza and João Pascoal Faria. Processpair: A tool for automated performance analysis and improvement recommendation in software development. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, ASE 2016, pages 798–803, New York, USA, 2016. ACM.

[73] Mushtaq Raza and João Faria. Empirical evaluation of the processpair tool for automated performance analysis. 07 2016.

[74] Amanda Ross and Victor L. Willson. *One-Way Anova*, chapter 5, pages 21–24. SensePublishers, Rotterdam, 2017.

[75] Célia M. D. Sales. Anova a 1 factor no spss testes de post hoc. Online Slides from Universidade Autónoma de Lisboa. Last access at June 2018., july 2010.

[76] S. Shrum. Choosing a CMMI Model Representation. *The Journal of Defense Software Engineering*, 13(7):6–7, 2000.

[77] Ron Snee. Impact of six sigma on quality engineering. 08 2000.

[78] Ian Sommerville. *Software Engineering*. Addison-Wesley Publishing Company, USA, 9th edition, 2010.

[79] Laerd Statistics. Mann-whitney u test using spss statistics. Online Article. Available at: https://statistics.laerd.com/spss-tutorials/mann-whitney-u-test-using-spss-statistics-2.php. Last access at July 2018., 2018.

[80] Laerd Statistics. One-way anova in spss statistics. Online Article. Available at: https://statistics.laerd.com/spss-tutorials/one-way-anova-using-spss-statistics.php, 2018.

[81] Xiaobing Sun and et al Bin Li. Mining software repositories for automatic interface recommendation. *Scientific Programming*, 2016.

[82] IBM Support. Pm62819: Reading an xlsx format file containing certain complex cell equations into ibm spss statistics 20 fails with error 2063. IBM Support Website. Available at: http://www-01.ibm.com/support/docview.wss?uid=swg1PM62819. Last access at February 2018., may 2018.

[83] Anna Reid Tania Prvan and Peter Petocz. Statistical laboratories using minitab, spss and excel: A practical comparison. *Teaching Statistics*, 24(2), september 2002.

[84] Statistics How To. Welch's anova: Definition, assumptions. Online Article. Available at: http://www.statisticshowto.com/welchs-anova/. Last access June 2018., october 2016.

[85] Statistics How To. Anova test: Definition, types, examples. Online Article. Available at: http://www.statisticshowto.com/probability-and-statistics/hypothesis-testing/anova/. Last access June 2018., March 2018.

[86] Wu Tongyuan, Luo; Chao and Duan Lixiang. Fishbone diagram and risk matrix analysis method and its application in safety assessment of natural gas spherical tank. *Journal of Cleaner Production*, 174:296–304, 2 In progress 2018.

[87] Carlos Torrecilla-Salinas. An agile approach to cmmi-dev levels 4 and 5 in web development. 08 2016.

[88] IEEE TSE, editor. *Comparing the Effectiveness of Software Testing Strategies*, volume 13. Springer Berlin Heidelberg, 1987.

[89] Dany Di Tullio and Bouchaib Bahli. The impact of software process maturity on software project performance: The contingent role of software development risk. *Systèmes d'Information et Management (French Journal of Management Information Systems)*, 18(3), 2013.

[90] Defense Acquisition University. Capability maturity model integration (cmmi) - primary functional area : Engineering and technology, 9 2017.

[91] Truly Innovative Vative. Quality tools & root cause analysis: Using six sigma quality and root cause analysis tools. Website Page. Available at https://vative.com.au/quality-tools-root-cause-analysis/. Last access at July 2018., 12 2017.

[92] Gregory W. Corder and Dale I. Foreman. *Nonparametric Statistics: A Step-by-Step Approach*. Wiley, 2 edition, 01 2014.

[93] R W. Covey and D J. Hixon. The creation and use of an analysis capability maturity model (trademark) (acmm). page 66, 07 2005.

[94] T. Wettstein and P. Kueng. A maturity model for performance measurement systems. In *Management Information Systems 2002*, pages 113–122, Ashurst Lodge, Southampton, SO40 7AA, UK, 2002. CA Brebbia and P Pascolo.

[95] Kam Cheong Wong. Using an ishikawa diagram as a tool to assist memory and retrieval of relevant medical cases from the medical literature. *Journal of Medical Case Reports*, 5(1):120, Mar 2011.

[96] Murray Wood, Marc Roper, Andrew Brooks, and James Miller. Comparing and combining software defect detection techniques: A replicated empirical study. *SIGSOFT Softw. Eng. Notes*, 22(6):262–277, November 1997.

[97] Murray Wood, Marc Roper, Andrew Brooks, and James Miller. Comparing and combining software defect detection techniques: A replicated empirical study. In *Proceedings of the 6th European SOFTWARE ENGINEERING Conference Held Jointly with the 5th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ESEC '97/FSE-5, pages 262–277, New York, NY, USA, 1997. Springer-Verlag New York, Inc.