UNIVERSITATEA DE STAT DIN MOLDOVA

Cu titlu de manuscris
C.Z.U.: [004.4.05 + 004.94] (043)

BERGMANN Ran

ASIGURAREA CALITĂȚII PROIECTELOR INFORMAȚIONALE

121.03 – Programarea calculatoarelor

Teză de doctor în informatică

Conducători științifici:

______________________ **BRAGARU Tudor**
Doctor în științe economice,
conferențiar universitar

______________________ **CĂPĂȚÂNĂ Gheorghe**
Doctor în științe tehnice, profesor universitar

Autor: ____________________________

CHIȘINĂU, 2020

MOLDOVA STATE UNIVERSITY

Presented as manuscript

U.D.C.: [004.4.05 + 004.94] (043)

BERGMANN Ran

PROVIDING QUALITY OF INFORMATION PROJECTS

121.03 - Computer Programming

PhD thesis in Computer Science

Scientific supervisors:

BRAGARU Tudor
Dr. in economic sciences,
associate professor

CĂPĂŢÂNĂ Gheorghe
Dr. in technical science, professor

Author:

CHISINAU, 2020

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisors - associated Prof. Dr. Tudor BRAGARU and Prof. Dr. Gheorghe CĂPĂŢÂNĂ. Their initial encouragement, continuous support, consulting and advice were enormously important for the realization of this thesis. I would like to thank them for their supervision of my thesis and for their important comments that considerably improved this work.

I would also like to thank all my colleagues and many people that volunteered to participate in my survey.

I would also like to express my gratitude to Dr. Julia Sirota and to the entire staff at the Carmel College, who have been so helpful and cooperative in giving their support at all times to help me achieve my goal.

Furthermore, I would like to thank reviewers and members of the Specialized Scientific Council, especially by reviewing and giving feedback to my thesis.

Last but not the least I am grateful to my family that supported me throughout all the process, especially my wife and my mother. Without them, neither my study of computer science, nor the dissertation, would have been possible.

CONTENTS

ADNOTARE

Teza ***„Asigurarea calității proiectelor informaționale”*** este scrisă în limba engleză și prezentată de domnul *Ran BERGMANN* pentru obținerea titlului de doctor în informatică, *specialitatea 121.03 – Programarea calculatoarelor.* Teza a fost elaborată la Universitatea de Stat din Moldova, Chișinău.

Structura tezei: Teza constă în Introducere, 4 capitole principale, Concluzii generale și recomandări, Bibliografie 167 de titluri. Textul principal cuprinde 161 de pagini, include 55 figuri, 16 tabele, 16 formule și 9 anexe. Rezultatele obținute ale tezei au fost publicate în 10 lucrări științifice, cu un volum total de peste 4 coli de autor.

Cuvinte cheie: proiect de informații (IP), sistem informațional (IS), software, calitatea IP, caracteristici de calitate, standarde de calitate, metamodel generic de calitate, model particularizat de calitate, sistem de management al calității (QMS).

Scopul și obiectivele cercetării. Scopul tezei constă în furnizarea de calitate a IPs. **Obiectivele tezei** constau în dezvoltarea metamodelului generic de calitate, care integrează cunoștințele despre modelele de calitate cunoscute, factorii de calitate și cele mai bune practici, prezentate în standardele internaționale actuale; identificarea factorilor de calitate și construirea modelelor de calitate particularizate/specifice, obținute din metamodelul generic de calitate; evaluarea calității IP/IS de-a lungul ciclului de viață; specificarea cerințelor și dezvoltarea unei aplicații software-suport pentru administrarea metamodelului, generarea modelelor specifice, evaluarea calității, ca parte a *Project Management Office* (PMO) cu implementarea rezultatelor într-o organizație.

Noutatea științifică și originalitatea sunt reflectate într-o nouă abordare pentru evaluarea și îmbunătățirea continuă a calității IPs de-a lungul ciclului de viață, bazat pe combinația între metodologia modernă de dezvoltare Agile și modele de calitate particularizate, obținute din metamodelul generic și în aplicația originală/instrument software-suport a noii abordări.

Problema științifică importantă rezolvată în cercetare: noua abordare deschide posibilitatea de a defini calitatea IPs la nivel conceptual, creând baza pentru evaluarea formală ulterioară a gradului de conformitate a IPs dezvoltate cu cerințele de calitate.

Semnificația teoretică este confirmată de analiza, generalizarea și determinarea principiilor teoretice ale noii abordări a procesului de evaluare continuă a calității IP pe întregul ciclu de viață al proiectului, bazată pe combinația și relația dintre modelul de îmbunătățire a calității Deming, modele de calitate adaptabile și metodologia modernă de dezvoltare Agile.

Valoarea aplicativă. Noua abordare de evaluare a factorilor de calitate, metamodel generic, modele particularizate, funcțiile de măsurare - toate acestea au fost realizate într-un aplicație instrumentală software IPMS - *Information Project Management System,* ca extensie PMO. Abordarea propusă și aplicația elaborată au un potențial imens pentru industria software în reducerea semnificativă a timpului și costului evaluării calității IPs și îmbunătățirii calității.

Implementarea rezultatelor. Noua abordare de evaluare a calității și instrumentul software realizat sunt implementate în „WGS”, Israel *(Anexa 5)* și în procesul de studiu al Universității de Stat din Moldova *(Anexa 6).*Dar, aceste rezultate pot fi, de asemenea, utilizate direct de orice alte organizații implicate în dezvoltarea IPs și/sau de către cercetătorii și studenții altor instituții de învățământ a disciplinelor de inginerie software.

ANNOTATION

The thesis ***„Providing Quality of Information Projects”*** is written in English and submitted by Mr. *Ran BERGMANN* for fulfillment of the requirements for the PhD in informatics, specialty *121.03 – Computer programming*. The thesis was elaborated at the Moldova State University, Chisinau.

The structure of the thesis: The thesis consists of *Introduction, 4 main chapters, Conclusions and recommendations, Bibliography* of 167 titles. The main text amounts up to 161 pages, includes 55 figures, 16 tables, 16 formulas, and 9 annexes. The obtained results of the thesis were published in 10 scientific papers, with a total volume over 4 sheets of author.

Keywords: Information Project (IP), Information System (IS), Software, Quality of IP, Quality characteristics, Quality standards, Generic Quality Metamodel, Tailored Model of Quality, Quality Management System (QMS).

Research Goal and Objectives. The aim of this thesis is to provide quality of IPs. **The objectives** of thesis are described as follows: developing the generic quality metamodel, which integrates the knowledge about known quality models, quality factors, and the best practices, presented on the actual international standards; identifying of the quality factors, and building the tailored/specific quality models, obtained from generic quality metamodel; assessing of quality along lifecycle; specify the requirement, develop a software application to support for metamodel administration, generation of tailored models and quality assessment, as part of the Project Management Office (PMO), with the implementation of results in an organization.

The scientific novelty and originality are reflected in a new approach for continuous assessment and improvement of IPs quality along lifecycle based on combination between modern Agile development methodology and tailored quality models, obtained from generic metamodel and in an original digital application/tool for support of new approach.

The important scientific solved problem in the research: new approach opens up the possibility to define the quality of IPs at the conceptual level, creating the basis for the subsequent formal assessment of the degree of compliance of the developed IPs with the quality requirements.

The theoretical significance is confirmed by the analysis, generalization and determination of the theoretical principles of a new approach for the continuous process of assessing the quality of IP throughout the project life cycle, based on the combination and relationship between the Deming quality improvement model, adaptable quality models and the modern Agile development methodology.

The applicative value. The new assessment approach for quality factors, generic metamodel, tailored models, measurement functions – all of these have been realized in a software application tool IPMS - *Information Project Management System*, as extension for the PMO. The proposed approach and elaborated application have huge potential for software industry in reducing significantly the time and cost of quality assessment of IPs and improvement of quality.

The implementation of the results. The new assessment approach and the realized software tool are implemented in "WGS", Israel *(Annex 5)* and in the study process of the Moldova State University *(Annex 6)*. However, these results also can be directly used by any other organizations concerned with IP development and/or by researchers and students of other educational institutions at software engineering disciplines.

АННОТАЦИЯ

Диссертация на тему ***«Обеспечение качества информационных проектов»*** написана на английском языке и представлена господином *Ран БЕРГМАНН* для получения степени кандидата наук по Информатике, специальность *121.03 – Компьютерное программирование.* Диссертация была разработана в Молдавском Государственном Университете.

Структура диссертации: Диссертация состоит из введения, 4-ех основных глав, заключения и рекомендаций, списка литературы из 167 наименований. Основной текст составляет 161 страницы, включает 55 рисунков, 16 таблиц, 16 формул и 9 приложений. Полученные результаты опубликованы в 10-ти научных работах, общим объемом свыше *4 авторских листов.*

Ключевые слова: *информационный проект (IP), информационная система (IS), программное обеспечение, качество IP, характеристики качества, стандарты качества, обобщенная метамодель качества, частная модель качества, система менеджмента качества (QMS).*

Целью работы является обеспечение качества IS. Подцели состоят в разработке обобщенной метамодели качества, которая объединяет знания об известных моделях качества, факторах качества и лучших практиках, представленных в актуальных международных стандартах; выявление факторов качества и построение частных/ специфических моделей качества, полученных из обобщённой метамодели качества; оценка качества на протяжение жизненного по цикла *IP*; спецификация требований и разработка программного приложения-инструмента для поддержки метамодели, создания специализированных моделей и оценки качества IS, как часть Офисного Управления проектами (PMO), с внедрением результатов в некоторой организации.

Научная новизна и оригинальность отражены в новом подходе к непрерывной оценке и улучшению качества IS на протяжении жизненного цикла на основе сочетания современной методологии разработки Agile и адаптируемых моделей качества, полученных из обобщенной метамодели и в оригинальном цифровом приложении/инструменте для поддержки нового подхода.

Важной научной проблемой, решаемой в исследовании, является новый подход, который открывает возможность определения качества IS на концептуальном уровне, создавая основу для последующей формальной оценки степени соответствия требованиям качества, разработанных IS.

Теоретическая значимость работы подтверждается анализом, обобщениями определением теоретических принципов нового подхода для непрерывного процесса оценки качества ИС на протяжении жизненного цикла проекта, основанного на объединение и связь между моделью повышения качества Деминга, адаптируемых моделей качества и современной методологии гибкой разработки Agile.

Прикладная ценность работы. Новый подход к оценке факторов качества, обобщенная метамодель, адаптируемые модели, функции измерения были реализованы в программном инструменте IPMS *Information Project Management System* как расширение для PMO. Предлагаемый подход и разработанное приложение имеют огромный потенциал для индустрии программного обеспечения в значительном сокращении времени и затрат на оценку качества ИС и их улучшение в программном инструменте (Information Project Management System) как расширение для PMO.

Внедрение результатов. Новый подход к оценке качества и реализованный программный инструмент внедрены в "WGS", Израиль (Приложение 5) и в учебном процессе Молдавского Госуниверситета (Приложение 6). Эти результаты также могут непосредственно использоваться любыми другими организациями, занимающимися разработкой ИС, и/или исследователями и учащимися других учебных заведений по дисциплинам, связанных с разработкой программного обеспечения.

LIST OF ABBREVIATIONS

ANSI - The American National Standards Institute

ASQ - The American Society for Quality

BI - Business Intelligence

CDC - Centers for Disease Control and Prevention

CMM - Capability Maturity Model

CMMI - Capability Maturity Model Integrated

COBIT- Control Objectives for Information and Related Technology

COQ - Cost of Quality

COTS - Commercial Off-The-Shelf Components

CRM - Customer Relationship Management

ERP - Enterprise Resource Planning

FMEA - Failure Mode and Effect Analysis

I&CT - Information and Communication Technologies

IDA - Interchange of Data between Administrations

IDI - I&CT Development Index

IEEE - International Electrical and Electronics Engineers

IP, IPs - Information Project/ (s), such IS/IT, web, software development project etc.

IQ - Information Quality

IS - Information Systems

ITSM - IT service management, ISO standards series 20000 (taken form ITIL)

ISACA- Information Systems Audit and Control Association

ISO - International Organization for Standardization, founded in 1946

IT - Information Technology

ITIL - Information Technology Information Library

GDPR - General Data Protection Regulation

KIS - Knowledge-based Information Society

MOF - Microsoft' Operations Framework

MS - Microsoft

NIST - U. S. National Institute of Standards and Technology

OCIO - Office of the Chief Information Officer

OPM3 - Organizational Project Management Maturity Model

PDCA - Plan – Do – Check – Act (PDCA circle)

PM - Project Management

PMBOK-	Project Management Body of Knowledge
PMI -	Project Management Institute
PMIS -	Project Management Information Systems
PMO -	A Project Management Office that defines and maintains standards for PM within the organization
PRINCE2-	Projects in Controlled Environments, version 2, a development methodology
QA -	Quality Assessment
QM -	Quality Management
QMS -	Quality Management System
ROI -	Return on Investment
SDLC -	System/Software Development Lifecycle
SQuaRE -	Systems and software quality requirements and evaluation
TQM -	Total Quality Management
WEB -	World Wide Web (WWW)

Every business is a software business now. Agility isn't an option, or a thing just for teams, it is a business imperative…
(Dean Leffingwell, Creator of SAFe)

INTRODUCTION

Topicality and importance of the information project quality.

The term „Information Project" (*IP, to the plural IPs)* has already come in scientific use for some time. IP is a particular case of the project concept. IP may regard the computerizing of a function inside an organization or modernizing of an older computer investment, as well as all the activities that are meant to reorganize information related processes, development of the software systems and applications. In the thesis under the notion of Information Project we will mean the implementation of I&CT in any activities, including development of the information systems, software systems and applications for the information consumer needs.

Information and Communication Technology sector grooving. I&CT Development Index (IDI) for Moldova in 2017 reached 6.45, ranked 59th out of 170 countries, and Israel - 7.88, 23rd. For comparison, Iceland ranks first with the index value of 8.98 *[1]*. IDI is a composite index combining 11 indicators, categorized into three sets, which reflecting the level of I&CT access, I&CT use and I&CT capability and skills required to use I&CT effectively in the society.

Share of I&CT sector in Gross Domestic Product (GDP) from Moldova in 2017 and 2018 was 9% *[2]*. Compared with 2017, share of I&CT in GDP Russia is 3%; in South Korea – 12%, in Poland's – 6.06%, in Germany 8.15%, in Czech Republic 8.43%, in France 7.33%. In January 2019, it became known that private technology companies in Israel, which are the main driver of the country's economy, achieved a record of $ 6.47 billion in investment in 2018. This is 17% more than a year earlier *[3]*. The evolution of information technology continues to affect significantly the business environment and demonstrate *the need to provide quality in information projects*. I&CT changes business practices, reduces costs and alters the ways in which systems should be controlled. In addition, according to ISACA (https://www.isaca.org/), technology plays a key role in these actions; as it is becoming pervasive in all aspects of business and personal life; I&CT controls the level of knowledge and skills required to control and audit information systems, and it increases the need for well-educated professionals in the fields of information systems (IS) governance, assurance, security and control *[4]*.

One of the motivating factors to work on the thesis was the release of a new series of ISO 25000 standards, as a new generation of regulatory documents of the International Organization for Standardization in the field of software engineering related to standardization and software quality assessment. This is a landmark event in terms of the evolution of models that reflected changes in software engineering, which seems to be a real swamp, if you look at the number and the volume of the standards and various requirements *[5]*.

I&CT, software has become an important component of many spheres of life as it is used in all fields of activities like education, industry, services, management, etc. Often, I&CT has an influence not only on the efficiency of management and production, but also the lives of people. For example, I&CT is actively used in medicine, including massive implantation of chips in human beings.

The military, space, Internet and other important sectors are highly dependent on software. *For example, this year the US Company Google received a record fine of 50 million EUR for violating the GDPR (General Data Protection Regulation) directive. Facebook has been fined $ 5 billion for violating data privacy rights. Since the March disaster, company Boeing 737 MAX shares have fallen 15%, which is the biggest deviation in the Dow Jones Industrial Average, Bloomberg notes. The company continues to lose money after the bad reputation of only manufactured and best-selling airliners, which was damaged due to a software failure. Only in 3 months of the second quarter, the company lost about $ 5.6 billion from a massive ban on flights of new Boeing 737 MAX models.* For this reason, software quality assurance is extremely relevant.

Problem Statement. Neglecting the great importance and impact of I&CT, **today still persists poor information projects performance:** only 32% of the software projects are successful *[5], other 68% are challenged or failed*. The PMI report *„Pulse of the Profession*2017*"* showed a slight improvement, but broadly speaking, the statistics tell a shocking story. „Failed projects" are still over 30%. The average budget loss on projects for under performing companies was 46%.Over a third (34%) of projects aren't base lined at the planning stage *[6]*.

More of IT/IS/software development projects have a bad reputation for going over budget and schedule, not realizing expectations and for providing poor return on investment. Surveys and reports on the acceptability of new IT/IS/software systems seem to highlight constantly the same problems and probable causes of failure yet businesses, large and small, continue to make mistakes when attempting to improve information systems and often invest in inappropriate or unworkable changes without proper consideration of the likely risks.

For example, already in 1995, during the first stages of the Internet and information revolution, *[7]* estimated that in the U.S. alone the worth of 175, 000 IS projects that were at different progress stages that year was about $ 250 billion, out of which a total amount of $ 59 billion (about 24%) is the result of the excessive costs and an additional $ 81 billion $ (32.4%) was the value of the projects which were canceled at the end. Similarly, *[8]* report that more than half of the projects in the field of information systems in the United States exceeded their budget or their schedules, with 16.2% of IS projects failed to meet these both objectives.

These figures cannot be ignored both in the view of the vast money sums that were invested in them and in the view of the business risk which entails for organizations. Seemingly, one might have expected that with the development of the IS technology, and with its progress in PM methodologies research as well as the stabilization and maturation of the market, the initial failure and unsuccessful projects rate should have decreased. However, recent studies indicate that the phenomenon continues to exist. For example, *[9]* it is estimated that until 2001 about $ 1 trillion of $ 2.5 trillion $ invested in IS PM were directed to unsuccessful projects (failure rate of 40%), ten years later estimates show that the failure rate of 33% persists *[10]*, ***and worse of all is that the chance of the failure of a large scale IS projects is about 85% [11]***.

According to *CHAOS report „IT projects failure and success" for years 2011-2015*, published by the Standish Group *[12-16]*, one can observe some improvement in successful managing of software projects during the last decade, but still there are important problems to be addressed when undertaking software development type projects.

As given in the examined reports, the failure rate of IS/IT projects has increased compared to the success rate of such projects. There are two main disciplines to address this issue. ***The first discipline focuses on the „positive" factors***, *e.g. identifying and characterizing of successful projects.* Studies *(such as [8], [17], [18])* that review a wide range of technological and economic features as well as behavioral and management perceptions which can differentiate a relatively successful PM from other less successful ones. ***The second discipline aims at identifying the factors common to failures.*** According to this vision, these factors are naturally expressed by terms of project management and dealing with risk factors through quality assurance. In order to deal with the risk, one should first acknowledge its existence, identify different kinds and types of risks, and respectively classify these types and rank them in order of relevance and the threat they embody in relation to the project (e.g., prioritization), hence „managers and other stakeholders plan, implement, and monitor actions to control or mitigate risks" or, practically stated, manage the IS project *[10]*.

Obviously, a successful system solution through information project activities depends on providing quality. This led to the fact that QM in software developments is now recognized by *ISO, ISACA, IEEE, PMI* as an important discipline, along with software engineering.

But the quality of IPs is a complex concept, it means different things to different people, and it is highly context dependent. An appropriate model is *built on recognizing steps toward a standard solution* – it helps when the work of one team can be continued by another team at the point where the first stopped. Work is thus forwarded from team to team and time zone to time zone until it has been completed. *Since the quality of IPs can not only be verified/controlled at the end of the project –it should be built and monitored throughout the lifecycle from conception to use.* This is one of the reasons to examine in this thesis the possibility of improving the quality of IPs ***using a good methodology for project development and quality management***.

It should be noted that presentation of uncertainty or the concept of the risk in IPs is quite simplistic in that it distinguishes early stages, which focus on identifying and recognizing these factors, from later stages of the PM that focus on management and quality assurance. Apparently, this might be seen as an administrative issue, however in practice, as explained by *[19]*, it turns out that with the development of the software industry in general, and especially with the growth and diversification of the use of complex and advanced technologies, project managers face an exponentially increasingly range of risks, hence the complexity and the urgent need for providing quality to Information Projects.

In conclusion, IS and software systems have been playing a key role in organizations for decision-making and efficient business flow for years. Issues regarding the evaluation of software quality, data quality and information quality have been noticed and identified increasingly within the field of information systems management in recent years. Two main needs arise from this background:

- *Provide project managers a better understanding and methodologies for assuring information projects quality.*
- *Increasing the quality of information projects along lifecycle, focusing on information systems and application software*.

The main idea to be drawn from context analysis is that ***the quality of the project and the resulting product can be effectively managed considering both the software development methodology and tools that are suited to the project requirements and team abilities, as well as good management practices focused on specific standards families and quality models.***

This is the main subject of the thesis, referred to as evaluating and improving IPs quality in accordance with a better standardized practice and using tailored quality models. Today *the quality assessment of informational projects is an important discipline and a field of study with interesting perspectives for researchers,* since the rate of failed informational projects is still quite high.

There is a consensus, both among scholars and among industry professionals, that informational project is at least a challenging process. The business domain of QM of IPs and software development is mainly characterized by complexity due to a rare combination of innovation and technical and of business and managerial complexities and uncertainty.

The research domain is focused on providing quality of information projects because within the framework of contemporary organizations about 50% of the activity takes place in the form of project activities and all 100% in the specialized organization for development and implementation of information system, web applications, software application etc.

The goals and the objectives of the given thesis.

At the present moment, on the one hand, we have the high demand for success IPs for Knowledge-based Information Society (KIS), e-business, e-economy etc., and, on the other hand, – we have the high share of IPs failure. The quality of IPs can not only be verified/controlled at the end of the project; it should be built and monitored throughout the lifecycle from conception to use.

The main purpose of the research is providing quality of information projects, considering the high-quality requirements at low costs and the modern trends of the software development methodologies, including standardization.

But the context of different organization and quality characteristics for different types of information projects/systems are very different. For example, a database could differ vastly from an Internet site and their quality characteristics should be different, accordingly.

Building an integrated metamodel of IPs quality and the quality of resulting products requires a great number of studies, analysis of I&CT and management standards, development methodologies for IPs etc. The role of resultant tailored quality models is *to support the main activities of quality assessment.*

The objectives of the thesis are the following:

- Study, analyze and identify the quality approach framework based on the best practices presented on the actual international management standards, the specific context of the organization and the most appropriate IS/software development methodologies to provide quality of IPs.
- Identify the quality characteristics from the literature review regarding IPs, analyze the best quality models and practices for different IPs and develop a generic metamodel of quality, which will integrate quality factors, suitable for widely used types of IS.
- Perform on-site research to verify the relevance of the selected quality characteristics through the survey of experts in the field of information projects.

- Specify the requirement and develop a software tool to support for new approach *(as part of Project Management Office, PMO).*
- Implementing the developed approach in an organization.

These are the main subjects of the thesis, referred to as ensuring IPs quality *(planning, management, control)* in according to the good practices, stipulated in a series of international standards, including ISO 9000, IS0 10000, ISO 20000 (ITIL), ISO 25000 families, ISACA (COBIT) and in according with concrete context of the organization, type of the project and the model of quality for this product.

Some organizations, such as ISO, IEEE, ISACA, etc. try to standardize software quality by definite models, combining and relating software quality characteristics and subcharacteristics. Meanwhile, researchers propose software metrics as tools to measure programs source code, architecture, and performances. However, the relation between software quality models, metrics and type of projects is not yet clear and consensual. Moreover, *the process of software quality assessment remains an open issue with many models, poorly applicable in practice.*

The research is based on the following **assumptions**:

- The quality of IPs processes throughout SDLC and the resulting product quality (IS and information), even if they mean different things, need to be treated together.
- In order to improve the quality of information projects, it is required to present an information project quality assessment model that can be iteratively measured and improved, during the period of the project lifecycle.
- Among the possible solutions to the mentioned issues is modifying IS/software quality models so that characteristics and subcharacteristics are more meaningful to their users.
- IPs quality management along SDLC can be streamlined by automating routine work and using input data directly from modern Agile software development processes, assisted by digitized PMO.

The scientific novelty and originality are reflected in a new approach for continuous assessment and improvement of IPs quality along lifecycle based on combination between modern Agile development methodology and tailored quality models, obtained from generic quality knowledge metamodel, which must be extensible, flexible and adaptable and which must be supported by software application with primary data extraction directly from the PMO tools (export-import). The new approach is composed by following elements*: (1) generic metamodel of quality, which includes the best knowledge about quality factors, extracted from known basic models and quality standards; (2) tailored quality models built from metamodel, based on field* *research, which permit providing quality of some type of IPs; and (3) an original support application; with the extraction of the some initial data directly from the Agile PMO*.

The new suggested approach and application use collaboration tools and modern methodology to develop IPs and standardized best practices for managing and continuously improving of quality. As Agile software development processes systematically collect multiple information *(sprint retrospective),* it can be directly used to assess the quality of the project for formulating improvement tasks and increase project's successfulness for the customer.

In addition, a new approach was built and implemented in a software application, based on the research results dealing with meta-analysis of the most actual researches and tracking the 25 most important quality characteristics. This software application enables to manage quality of IPs by performing quality assessments, in accordance with the quality characteristics for seven type of information systems.

The important scientific solved problem in the research: new approach opens up the possibility to define the quality of IPs at the conceptual level, creating the basis for the subsequent formal assessment of the degree of compliance of the developed IPs with the quality requirements. A new approach permits continuous assessment of the quality of IPs along the lifecycle that can be systematically measured, calculated, managed and improved, based on primary data directly extracted from the PMO applications such as Jira, Version One, TFS, etc.

The theoretical significance is supported by analyze, synthesis, specifying and defining the theoretical principles and new approach for continuous assessment process of the quality of IPs through the project lifecycle, based on connection between quality metamodel, tailored quality models, Deming quality improvement model (PDCA) and Agile development methodology.

The applicative value of the thesis.

The new assessment approach for quality factors, generic metamodel, tailored models, measurement functions – all of these have been realized in a software application tool *Information Project Management System,* as extension for the PMO. The proposed approach and elaborated application can be directly used as it is or can be easily extended/adapted to user needs and both have a huge potential for software industry in reducing significantly the time and cost of quality assessment of IPs and improvement of them.

The approval of results. The results were published in 10 scientific papers, among which 8 by a single author, with a total volume over 4 sheets of author, including 2 in magazines recognized abroad, 2 in journals *category B*, were reported in 4 international conferences *(Mathematics & Information Technologies: Research and Education (MITRE-2015, MITRE-2016, MITRE-2019)", Chisinau, Republic of Moldova, „European Economic Integration", Chisinau,* *Republic of Moldova, 2016)* and 2 home conferences (*The 5th social psychology conference for PhD students, SODOCO, Haifa University, Israel, 2016, The 5th Kinneret Conference on Software Engineering Education, Kinneret Academic College, Israel, 2017*). Besides, the new assessment approach and the realized software tool are implemented in „WGS", Israel *(Annex 5)* and in the study process of the Moldova State University *(Annex 6)*. These results also can be directly used by any other organizations concerned with IPs development and/or by researchers and students of other educational institutions at software engineering disciplines.

Applied methods of research. Various methods of study and comparative analysis of sources of information with synthesis were used in the thesis. To evaluate the success/failure of the project, qualitative analysis methods are used for the triple constraints – *Schedule, Cost, Scope* and recently added a few more things to manage such *Quality, Risk, and Customer Satisfaction.*

The ***work has a synthesis and applicative character with research and development elements.*** The presence of a large number of quality norms/standards, abundance of information on this subject is a real challenge, is a „swamp" that can easily „swallow" those who want to implement a robust and efficient quality system. Solving the problem requires a profound study and a general interrelated analysis of best practices/quality frameworks, summing up a relatively simple and transparent metamodel to help the user choose the right strategy, tailored model and policy quality of the organization.

Choosing the right tailored model and methodology in according with projects types and the influential quality characteristics is crucial to the success of the project. The paper tries to guide developers to Agile software development methods, which is in line with the general principles and PDCA approach of quality management (ISO 9001) to continuous improvement of quality and the integrated metamodel of quality, which encompasses the basics quality models, proposed in ISO 9126, ISO 25010, ISO 25012 etc.

To determine which characteristics are more influent for different type of IS, has realized a qualitative research – interviews of a heterogeneous sample representing managers, customers and other practitioners in the fields of Information Systems and Projects Management.

The structure of the doctoral thesis.

The thesis consists of Introduction, four chapters, General conclusions and recommendations, Bibliography, and nine Appendices.

Introduction describes the topicality and importance of the raised problems, the goals and objectives of doctoral thesis, scientific novelty, of the obtained results, theoretical importance and practical value of the work, results approval and summary of the doctoral thesis sections.

Chapter ***I „State-of-the-art in the domain of IPs quality"*** deals with the theoretical framework of the topic and includes literature review of *information project, project success/failure, project quality management, diagnose the problems and assumptions to solve of them*.

Chapter ***II „Methodological approaches of project quality management"*** describes the general *framework of quality, some of quality concepts and principles defined by Deming, including Plan-Do-Check-Act cycle, used in all of the management standards and also known as PDSA cycle, the „Deming Wheel" and „Shewhart Cycle". Chapter II attempts to bring more understanding to the use of standards appropriate to the company's needs.*

Chapter ***III „Software quality models and tools"*** offers an overview and a critical analysis of the system/software quality models, establishes the premises and formulates the basic tasks for the realization of the metamodel and the particular quality models of IPs.

Chapter ***IV „Field research on information systems quality"*** describes the general metamodel of quality, the tailored models of quality for some type of IS, the numerical methods for multicriterial calculus of quality and realized application for support of them.

The ***„General conclusions and recommendations"*** summarize the contribution of the research from different aspects: mention the important scientific and applicative solved problem, describes three main obtained results the significance and potential of the proposed metamodel and application for software development organizations and suggestions for perspective research in the domain of IPs and software quality improvement.

In addition, there is a *Bibliography, Publications and List appendices*, with supplementary information of doctoral thesis, including questionnaire, software listings, acts confirming the implementation of the obtained results, etc.

Keywords: Information Project (IP), Information System (IS), Software, Quality of IPs, Quality characteristics, Quality standards, Basic Model of Quality, Tailored/particularized Model of Quality, Quality Management System (QMS).

Where there is a lot of knowledge - there is a lot of annoyance (Kohelet, Bible)

The more you know, the more you hurt; the more you understand, the more you suffer (Contemporary English version)

I. STATE-OF-THE-ART IN THE DOMAIN OF IPs QUALITY

Despite the basic insight from literature that failure of information project is still big, and that software quality is related to failures of the software, but also to the lifecycle costs of it, a-inacceptable definition of IPs quality is still missing. In the first chapter, we introduce a comprehensive discussion of the information projects, quality of project, quality of resulting product and of related terms. This chapter highlights the importance of project development for modern society and the unsatisfactory state of IPs resolution, emphasize some important challenges regarding the quality of information projects and its resulting products.

1.1. Project development as an important activity for modern society

Modern information and communication technology change the way we live, work, learn, entertain, etc. Today almost everyone uses various and many software applications, especially mobile, which implies advanced product quality requirements and the challenge of manufacturers to survive in the highly competitive global marketplace. Contemporary society, called the Knowledge-based Information Society (KIS), is a new stage in its evolution, where change and innovation prevail. Modern KIS is characterized by:

- *Increasing* the role of information and knowledge. *Today not the one is rich who own tangible goods* (buildings, land, etc.), *but he who possesses rich knowledge* (intellectual property). *Information is estimated as the fourth vital element: after air, water, and fire.*
- The driving force of development has become the production of information and knowledge. The main product and the main value of the KIS have become information and knowledge.

- In KIS *there* is an increase in the share of informational communications, information products and *services* in gross domestic product (for example in the Republic of Moldova at 9-10% *[20]*).

The *Global* digital economy, the global information space, new electronic and mobile business models, and the old ones are supported by modern information technologies, all of which lead to the continuous increase of the demand for quality software.

Building globally KIS, *integrating* new information technology into all areas of human activity, developing digital information products and services, including *e-banking, e-payments, e-government, e-education, e-health,* etc., are the strategic objectives of the governments of most countries of the world, including the Republic of Moldova and Israel, which have adopted Digital Development Strategies *[20], [21]. KIS-building plans at national level involve the realization of many informational projects aimed at meeting the social and personal needs in information products and services,* including access to information resources for anyone who is empowered, wherever and whenever necessary, in safety conditions.

To do all of the above-named, *KIS requires many projects, especially systems and software applications,* including mobile. I&CT allow change the modern society and building the Global Information Society (GIS), with a large specter of IPs. Today demand of quality IPs is very big. Many organizations invest millions in hope to get some value in return from informational projects. The success of information projects and the quality of resulting product continues to be discussed and studied from many of international and national specialized organizations:

- Project Management Institute *(PMI, http://www.pmi.org/)*, with the *Guide to the Project Management Body of Knowledge (PMBOK), sixth edition, 2017, [22], Brilliant Agile project management. A practical guide to using Agile, Scrum and Kanban, first published 2015 [23], The Project Manager's Guide to Mastering Agile, principles and practices for an adaptive approach, 2015, [24]* etc.
- International Organization for Standardization *(ISO, http://www.iso.org/)*, with *ISO 9001:2015, Quality Management Systems, requirements [25], universal standard applicable to any organization; ISO 10006:2017 [26], Quality management in projects, family of ISO 25000, about 50 standards of Systems and software Quality Requirements and Evaluation (SQuaRE); ISO 20000:2011, a sets of detailed practices* for IT service management (ITSM) that focuses on aligning IT services with the needs of business *[27], ISO 27000, Information Security, today over 6o published standard [28],* etc.

- The Institute of Electrical and Electronics Engineers (IEEE, https://www.ieee.org/), *with IEEE 12207-2017, an ISO/IEC/IEEE International Standard, which describe software lifecycle processes [29], ISO/IEC/IEEE 29119 Software Testing; to support static testing, the following standard has been defined ISO/IEC 20246:2017, Work Product Reviews; to support the adoption of ISO/IEC/IEEE 29119-2, the Process Assessment Model for Software Testing has been defined in ISO/IEC 33063:2015* etc.
- American Society for Quality, *(https://asq.org/)* with *Guide to the Quality Body of Knowledge (QBOK [30]).*
- Information Systems Audit and Control Association, (ISACA, *http://www.isaca.org/*) wi*th your main product Control objectives for IT, the globally recognized Framework, the leader in ensuring effective and strategic enterprise governance of information and technology, updated in 2019 with new information and guidance facilitating easier, tailored implementation* (*COBIT, https://www.isaca.org/cobit/pages/default.aspx*), etc.

1.1.1. Project and Project quality management

The development of software for an improved business process, the construction of a building or bridge, the relief effort after a natural disaster, the expansion of sales into a new geographic market – all of this are projects.

The project is defined by different authors in many ways, sometimes in different standards differently, since each one pursues certain goals. For example, a *project* can be defined as:

1. *What We Want to Do (Larousse).*
2. *The goal we are pursuing, the image we are shaping about what we will do and the means we will use (Le Robert).*
3. *A limited-time effort to create a new product or service (PMI).*
4. *A set of integrated activities designed to achieve a pre-defined target within a determined timeframe and following an established action plan (The Organization for Economic Co-operation and Development, https://www.oecd.org/).*
5. *A combination of steps and techniques for timing and budgeting the work needed to achieve a result.*
6. *A management area that covers the activities of creating a product or service as a unique set of activities* linked to each other under specified time, budget, and expected result qualities.
7. *Unique process undertaken to achieve an objective (ISO 10006:2017).*

8. James P. Lewis considers the project to be a set of activities characterized by „*clearly defined start and finish deadlines, a range of activities to be carried out, a budget and an expected level of performance*".
9. *Something designed or proposed to be executed, a plan, a goal, a proposal (The Oxford English Dictionary).*

According to *[22], [31] „A project is a temporary endeavor undertaken to create a unique product, service, or result. The temporary nature of projects indicates a certain beginning and end. The end is reached* when *the project's objectives have been achieved or when the project is terminated because its objectives will not or cannot be met, or when the need for the project no longer exists".*

Definition no.6 and no.8 are the most complete. Definition no.1, no.7 and no.9 are short. Which is the best or the worst? No one is bad, no one is better. All definitions have the right to life, because they point out certain aspects. *Defined formulations of project have a higher value when considered together because they complement one another adding more features*.

So, a project consists of a sequence of interrelated actions (a set of unique complex activities) with a set start and end, with activities that are limited by time, resources and budget in order to achieve a clearly defined unique and precise objectives and requiring organizational structures and special methods.

A project is temporary, i.e. it has a defined beginning and end in time, and therefore defined scope and resources. *A project is unique* in that it is not a routine operation, but a specific set of operations, designed to accomplish a singular goal. *A project team also is temporary*, and can includes people who don't usually work together, sometimes from different organizations and across multiple geographies. And all must be expertly managed to deliver the results on-time, on-budget, learning and integration that organizations need.

The project has a precise purpose *(project features)* in response to a new problem to be solved. The goal determines the size of the project: how many resources, how long and how large the project is. The project has a specific, desired, and predefined outcome, as a rule, indicated by the name of the project. A project has a budget that limits the number of people, materials and money that can be used to end it. One project has a responsible person called a project manager. A project is a complex activity involving specific risks.

It is important to detach from the project definitions *three major aspects of quality, according to which one can define and relate to a project*. ***The first aspect*** is the need to undertake some endeavor, implicitly by some, which also raises the need for allocating some resources to some activities, and hence the need to coordinate, control and monitor them. ***The second aspect*** relates to the effort and resources invested in a project should provide a unique product *(usually tangible),* result, usually intangible *(such as a report, a document or other outcome*) or capability to perform a service *(usually in the context of as a business function).* This is why, according to this definition, a project, and hence project management, might relate to construction of a building, executing a structural organizational change or developing or modifying an Information System. ***The third aspect*** relates to a time frame of a project. More precisely, in spite of its tentative nature a project should end within a definite time frame from its initial. This period is not necessarily short, and more importantly is defined with relation to the completion of the objectives the project undertook and not to the outcome itself (for example, a building construction project will end after several months, but the building itself will stand for decades). Actually, the tentative nature of a project emerges from this focus on completion of the project's objectives since, as stated by the definition above, a project might end prematurely if the project managers might conclude that its objectives will not or cannot be met, or that the need for the project does not longer exist.

Traditional Project Management describes *[22], [32]:*

- Six process groups of PM: *Initiating, Planning, Executing (Implement), Monitoring and Control, Adapt and Closing (Figure 1.1).*

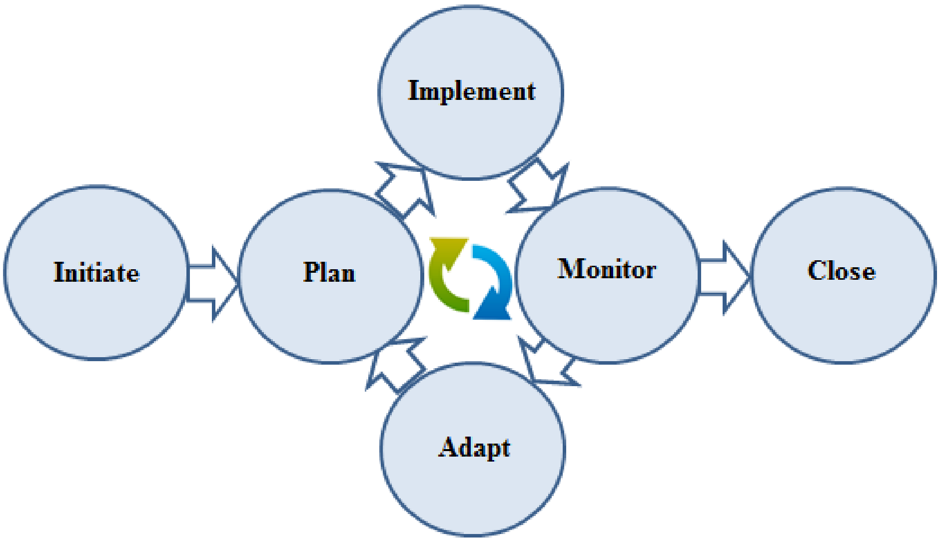

Figure 1.1. Project management cycle (lifecycle)

Source: *Adapted by the author based on [22] (Page 668), [32] (Page 12), (emphasize the PM cycle)*

- The 47 project management processes grouped into ten separate knowledge areas: *Project integration management, Scope, Schedule, Cost, Quality, Resource, Communications, Risk, Procurement, Stakeholder.*

- Recently added a few more things to manage: *Risk*, *Customer satisfaction, and Quality* which is *a subject of present work*

Full details on the core processes of PM we can found in *PMBOK [22], AGILE [23]. Below there will be briefly described only QM activities.*

The quality must to be defined at the **initiating stage** of project „Concept of quality" has the following sections:

- Quality policy and strategy;
- General requirements and principles of quality assurance;
- Standards, norms and rules;
- Integration of quality assurance functions;
- Requirements for a quality management system.

A policy and strategy are defined to ensure the quality of the product being developed that satisfies the expected demands of the consumer. At the **quality planning stage**, standards are defined that should be used to ensure that the content of the project meets the expectations of the project participants. Quality planning includes both the identification of these standards and the search for ways to implement them. The following are the main tasks of the planning phase:

- Determination of quality assessment indicators;
- Definition of technical specifications;
- Description of quality management procedures;
- Drawing up a list of objects of control;
- Selection of methods and means of quality assessment;
- A description of relationships with other processes;
- Development of a quality management plan.

Stage **Execution of project** includes organizing quality management, implies the creation of necessary and sufficient organizational, technical, financial and other conditions to ensure compliance with the quality requirements of the project and the project's products and the possibilities for meeting them.

Stage **Monitoring and control** of project in QM area consists in determining the conformity of the project results with the quality standards and the reasons for the violation of such compliance, regulation and analysis of the quality. The stage of quality control implies a regular check of the project implementation in order to establish actual compliance with previously defined requirements.

- Comparison of actual project results with requirements;

- Analysis of the progress of quality in the project throughout its lifecycle;
- Formation of the list of deviations;
- Corrective action;
- Documenting changes.

At the **Closing stage,** QM completion, including a summary assessment of the quality of the project results is completed, followed by the final acceptance, the compilation of a list of quality claims, the resolution of conflicts and disputes, documentation, experience analysis and lessons learned in quality management. Project quality management is carried out throughout the project lifecycle. Figure 1.2 shows the stages of project quality management.

The quality of software products can be **internal** – *measuring the static characteristics of the product, e.g. code length, module structuring*; **external** – *measurement of code behavior attributes when executed, e.g. correctness, reliability* and **quality in use** – by measuring attributes in a concrete context of use, e.g. pleasure, comfort. *The ultimate goal is quality in use*.

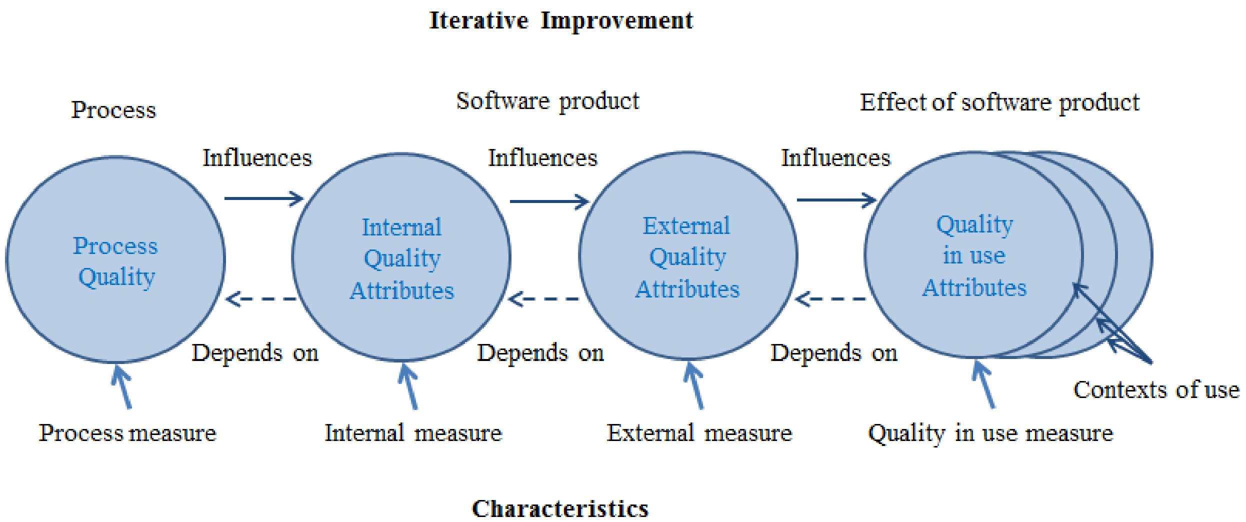

Figure 1.2. Quality in the life-cycle of software development

Source: *Taken from ISO 25010 [33] (Page27)*

The *internal indicators* of the software quality, inherent in the product itself are: complexity, quantity, severity and failure rate due to defects, determined during testing. This topic is about software engineering and is not examined in the thesis.

External quality of system/software measure a degree of programmable output that allow for functional systems to meet the requirements and implementations of this system, including programmable performance in the context of the use terms and conditions. Example – the number of failures detected during the test is an external indicator of the software quality, associated with the number of defects in the computing system.

Quality in use, the extent to which a product or system can be used by certain users to meet their requirements in achieving the goals of efficiency (including economic), avoiding risk, satisfaction, and context coverage in specified conditions of use.

In modern treatment of software quality (ISO 25000), both of external quality and quality in use is the product's quality.

For effectively management of project information, PMO uses digital application tools, such as:

- MS Project (https://products.office.com/en-us/project/project-management-software)
- Open Project (https://www.openproject.org)
- Primavera (https://www.oracle.com/applications/primavera/products/)
- Jira (https://www.atlassian.com/software/jira)
- Confluence (https://www.atlassian.com/software/confluence) etc.

In the thesis IPs are organizations, along with all the basic, essentially repetitive, processes that allow the organization to operate reliably, in other words, a sound and meaningfully built quality system. Project or Quality management is an application *of knowledge, skills, tools, and techniques to project activities to meet project requirements.* A framework for PM/QM *includes the project stakeholders, PM/QM knowledge areas, and PM/QM tools and techniques*. Stakeholders are the people involved in, or affected by project activities.

1.1.2. Triple constraint of the project and challenges of quality

Quality of products is multidimensional, is subjective, and only some aspects of quality can be measured. Better products, better project performance, and lower costs translate directly into increased competitiveness in an ever-more-global market place. This is the essence of a quality chain reaction described by W. Edwards Deming *improve quality, reduce costs, improve productivity, capture the market, stay in business, and provide more jobs [34].*

The general principle which governs the quality of all projects is represented by three elements: *time, cost and scope (Figure 1.3).*

The time that it takes to deliver a successful project, the complete cost of this project and the scope or specific description of what is to be achieved by the project, can each be represented by one side of a triangle. Like a triangle, any change to one element or side will affect the other elements or sides, and, ultimately, the shape of the triangle, which represents the level of quality (*Figure 1.4).*

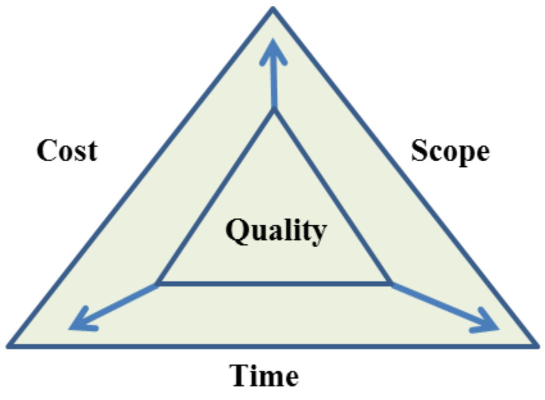

Figure 1.3. Traditional triple constraint of the project to obtain Quality

Source: *Adapted by the author based on PMBOK [22] (Annex A3, page 110), [63] (Page 43), [35] (Page 1) (emphasize the quality element in the triangle)*

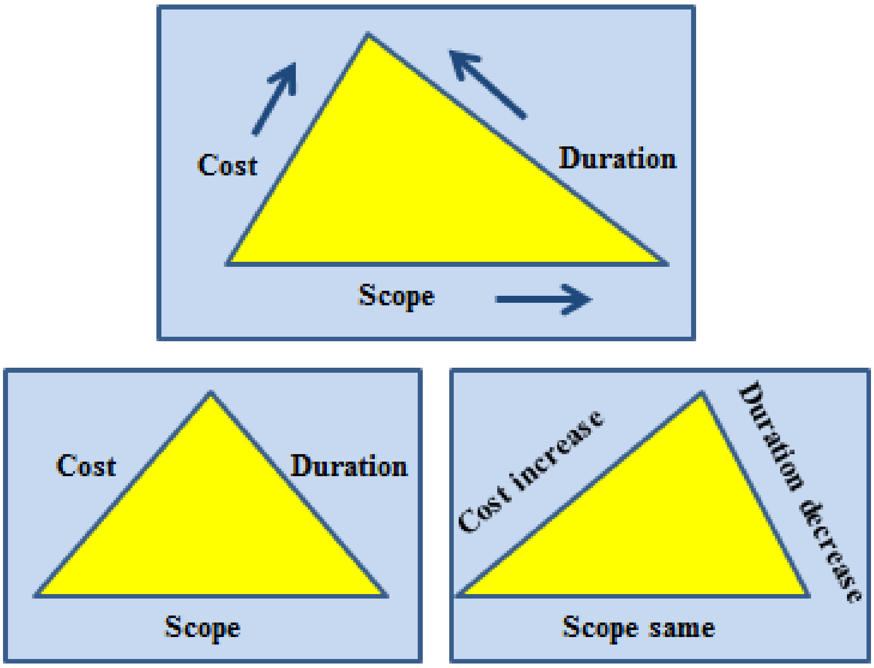

Figure 1.4. Dependence of Cost – Schedule – Performance

Source: *Adapted by the author based on [22] (Annex A3, page 110) (are illustrates the elements change effect)*

To explain it in simpler terms, if you make a change to the time taken to complete the project, one way or another, it will have an impact on either the cost or scope of the project or both. Likewise, changing the scope of the project will impact on the cost or the time taken or both and so on. This is a very important issue to remember because if you or another external force make a change to one of these elements during the life of the project there must be an understanding that the other elements will most likely be affected as well *[35]*.

Unbalanced influences create flaws and will produce a flawed system. These alone do not constitute failures; rather lead the system to failure. The factors which highly influence a system and then tend to create flaws and make hurdles in success of IS project are termed as risks. The factors, involved in creating imperfection, lead towards the increase of the complexity of IS and subsequently decrease its probability of success.

When we talk about the concept of project quality, we discover that human originality and creativity are closely related to it. *These aspects of quality are hard to measure, mainly because programmers see their work as a work of art rather than as a commercial product*. Another point is that the software quality is ***hard to define,*** but it is ***easy to recognize in its absence,*** it is ***transparent when present*** and**,** often, it is ***impossible to measure.***

Some challenges of IPs quality providing refers to the following factors, but not only to them:

- Software can’t be physically observed;
- The lack of knowledge of client needs at the start;
- The change of client needs over time;
- The rapid rate of change on both hardware and software;
- The high expectations of customers, particularly with respect to adaptability;
- Many of quality characteristic definitions have direct relations with the programming language, the environment for which a software product will implemented, etc.

The main challenge regarding the quality of IPs is related to the *appropriate choice of methodology and lifecycle model for product development*. **Changes in informational projects are inevitable** and generate a big amount of extra work:

(1) Getting agreement/approval of stakeholders;

(2) The necessity of frequent updating the plans, the schedules and the cost.

From the beginning of IPs development, the life model has evolved from the successive scroll methods „Waterfall” to Iterative, Incremental, and Evolutionary methods of product development, generically named ***Agile approach***. Shortly the difference, advantage and disadvantage for traditional and Agile approach will be discussed in section 2.2.

Within the software quality area, the need to provide a solution that matches user needs is often considered as design quality whilst ensuring a match to the specification is considered as manufacturing quality.

The second challenge regarding the quality of IPs: *Quality is a complex, multivalent concept, it means different things to different people, it is highly context dependent, and, as rule, it is a subjective estimation (Figure 1.5).*

David Garvin *[36]* described software quality from five different perspectives:

1) **Transcendental view,** excellence. It envisages quality as something that can be recognized but is difficult to define. Here quality is something that can be recognized through experience but is not defined in some tractable form. Quality is viewed to be something ideal, innate excellence, which is too complex to lend itself to be precisely defined.

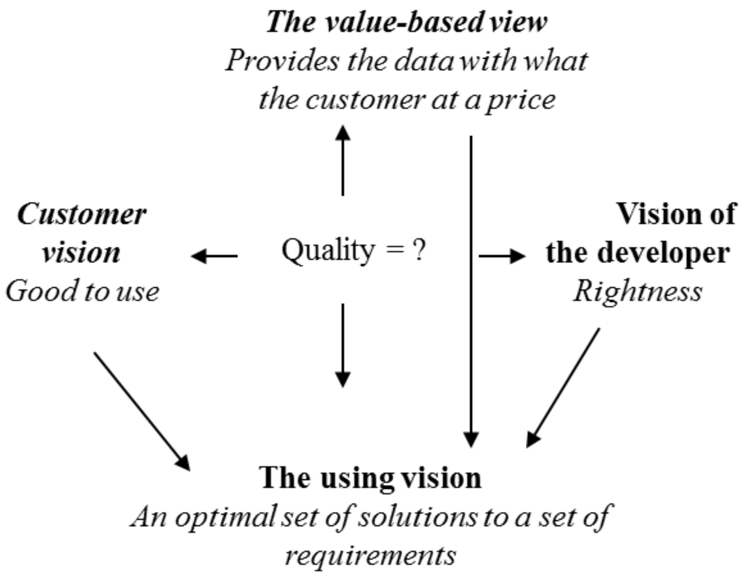

Figure 1.5. Different views of software quality

Source: *Developed by the author based on Garvin [36] (Page28)*

2) **User view,** fitness for intended use. It perceives quality as fitness for purpose. According to this view, while evaluating the quality of the product, one must ask the key question „Does the product satisfy user needs and expectations?" In this view, a user is concerned with whether or not a product is fit for use. Quality is not just viewed in terms of what a product can deliver, but it is also influenced by the service provisions in the sales contract.

3) **Manufacturing view**. Here quality is understood as conformance to the specifications. The quality level of a product is determined by the extent to which the product meets its specifications. Any deviation from the stated requirements is seen as reducing the quality of the product. The manufacturing view has its genesis in the manufacturing sectors, such as the automobile and electronics sectors.

4) **Product view,** quantities of product attributes. In this case, quality is viewed as tied to the inherent characteristics of the product (classical definition of quality). A product's internal qualities determine its external qualities. The product view is attractive because it gives rise to an opportunity to explore casual relationships between internal properties and external qualities of a product. An example of the product view of software quality is that high degree of modularity, which is an internal property, makes the software testable and maintainable.

5) **Value-based view,** quality vs. price**.** Higher the quality – higher the cost, greater functionality, greater care in development. Quality, in this perspective, depends on the amount the customer is willing to pay for it. The value-based view represents a merger of two independent concepts: excellence and worth where Quality is the measure of excellence and value is the measure of worth. Quality is meaningless if a product does not make economic sense. The value-based view represents a trade-off between cost and quality.

The third challenge regarding IPs quality is about various definition of software quality:

1. The totality of features and characteristics of a product or a service that bear on its ability to satisfy the stated and implied needs (ISO 8402 - Quality management and quality assurance - Vocabulary) *[37]*;

2. The quality of a system/software product is the degree to which it satisfies the stated and implied needs of its various stakeholders, and thus provides value (ISO/IEC 25023) *[38]*;

3.Capability of a software product to satisfy stated and implied needs when used under specified conditions (ISO/IEC 25000:2014) *[39]*;

4. Degree to which a software product satisfies stated and implied needs when used under specified conditions (ISO/IEC 25010:2011) *[33]*;

5. Degree to which a software product meets established requirements source (IEEE 730-2014) *[40]*.

As per ISO's definition of quality to a project end product implies that:

- The client's point of view should always prevail when assessing quality;
- Quality depends on a host of features and characteristics that contribute, to various degrees, to the client's needs and expectations;
- Quality is accrued progressively throughout the project lifecycle.

Corrective actions should be initiated as soon as significant quality deviations are detected.

Common understanding of all points of view (user view, customer satisfaction, manufacturing view, product view, etc.) is an issue, which can be solved by development of many documentations and bureaucratic processes.

The fourth challenge regarding the quality of IPs is a poor concept. Quality is not distributed only in one part. When we talk about software quality, we talk about assessing entire items which make part of the concept of quality. Quality as an objective value is dependent on sets of software attributes and customer's requirements, which are subjective. These attributes are explained as different level of characteristics and subcharacteristics in quality models, but the relation and impact of each characteristic and subcharacteristic should be distinguished. However, between quality characteristics, no matter what perspective they are viewed or grouped, there are multiple relationships of interdependence, subordination, hierarchy, composition or aggregation and the impacts of quality subcharacteristics on characteristics are not equivalent and it is hard to determine. *Models must be made more meaningful for different persons by using coefficients (factors), which relate characteristics and subcharacteristics.*

The solution can consist in a *categorization of the people who deal with software at different level by considering their need for software quality, and then to create tailored models for each group,* range of values which are acceptable for similar people and realize an original digital support for them. Using a good methodology and pattern *(such as design patterns, standardized processes, PDCA cycle, etc.),* we could increase the software quality.

PQM aims to identify the required project quality, assess and control it, and finally attain the optimum results through specific processes and activities. In order to obtain the desired result, a project manager must take care of the following three key concepts of quality management:

1. Customer satisfaction,
2. Prevention over inspection,
3. Continuous improvement.

These three targets can best be achieved in Agile; they help in accurately estimating what exactly a customer wants and what he actually needs. Once we have a clear understanding of these, we can effortlessly manage the project quality.

When we talk about IPs quality, we discover that human originality and creativity are closely related to it. *These aspects of quality are hard to measure, especially as programmers see their work as a work of art rather than as a commercial product.*

1.2. IT projects failure and success

According to CHAOS reports published by the Standish Group, we can observe some improvement in successful managing software projects during the last decade, but it is still remaining an important issue to be addressed when undertaking software development type projects. As given in the *Table 1.1*, *Figure 1.6,* below according to CHAOS reports, the failure rate of IT projects has increased compared to the success rate of such projects *[12-14], [16].*

Table 1.1

Projects resolution for years 2002 to 2015

Years	2002	2004	2006	2008	2010	2011	2012	2013	2014	2015	Average
Successful	34%	29%	35%	32%	37%	29%	27%	31%	28%	29%	31%
Challenged	51%	53%	46%	44%	42%	49%	56%	50%	55%	52%	50%
Failed	15%	18%	19%	24%	21%	22%	17%	19%	17%	19%	19%

Source: *Developed by the author based on [14](Page 3) , [16](Page 2)*

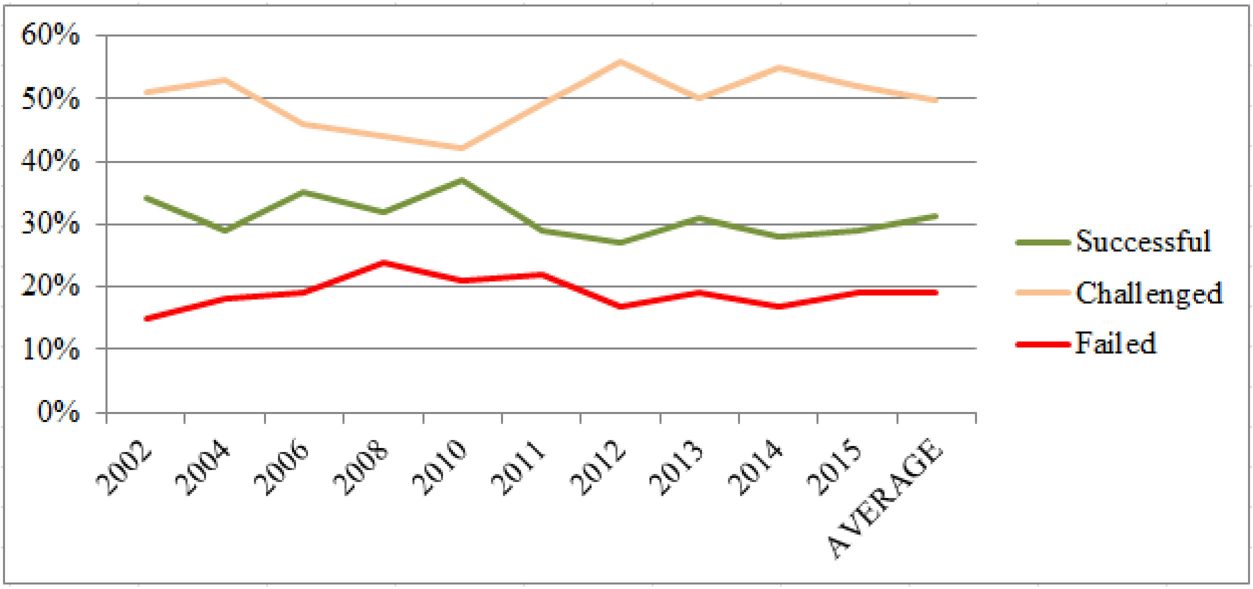

Figure 1.6. Projects resolution for years 2002 to 2015

Source: *Developed by the author based on [14] (Page 3), [16] (Page 2)*

The data comes from the new CHAOS data base in the fiscal years 2011-2015 with over 10000 projects. In the cells of the table above the bar there is the share by project type, and under the bar there is the share of the resolution. The definition of "Success Story" used in the Chaos Report 2015 *[16]* means that the project is timely, with a satisfactory result, and budget-oriented. This means that the project was done within reasonable time, stayed within the budget, and provided customer and user satisfaction, regardless of the initial requirements.

In general, the development and quality assessment processes of IPs are influenced by the following summarized indicators:

- Scope and purpose of project;
- The type of functional tasks to be solved;
- Volume and complexity of developed IS;

- The required composition and the required values of the quality characteristics of the IP and the amount of allowable damage due to their insufficient quality;
- The degree of connection of the tasks to be solved with the real time scale or the permissible length of waiting for the results of solving the problem;
- The predicted values of the duration of operation and the prospect of creating multiple versions of project;
- The estimated circulation of production and use of IS;
- Degree of required documentation of project and resulting IS.

Some more important factors are succinctly described as follows.

1.2.1. Dependency on quality of project size and development methodologies

The level of effort devoted to managing IPs projects should be commensurate with the size and scope of those projects, thus, projects are categorized so that appropriate project management processes and procedures can be applied to the various categories.

Categorize of the size of projects allow that appropriate project management processes and procedures can be applied to the various categories. The classification will assist the organization in describing the important characteristics of each project, and it will allow stakeholders to immediately associate project importance to the business, development team etc.

Detailed about IPs classification (e.g., project size, estimated total project value, degree of project complexity) can consult in *[41]*.

A trend from previous reports is how smaller projects have a much higher likelihood of success than larger ones, as shown in the *Table 1.2* and *Figure 1.7.* After analyzing the CHAOS report 2015 according to the size of the projects, the following tendency has been noticed: ***the resolution worsens with the increase of the size of the project***. Near 69% of all projects in 2015 failed or were challenged *(Table 1.1) and n*ear 86-87% they were medium and large projects *(Table 1.2).* In the cells of the *Table 1.2* the counter indicates the share of projects by types *(small, moderate, medium, large, Grand)* and the denominator - the share of *successful, challenged* or *failed* projects by types. Professionals in the field of software development are worried by this situation. As a result of insufficient resolution, in 2000, a new approach of Agile IPs development, the Manifesto and Principles of Agile Philosophy was approved. As it will be discussed below, this new *Agile approach is consistent with the principles of quality management, process approach, client satisfaction, leadership commitment, staff involvement, etc.*

Table 1.2

Projects resolution by size for years 2011 to 2015

Size of project	Successful	Challenged	Failed	Total
Grand	2/6	7/51	17/43	100%
Large	6/11	17/59	24/30	100%
Medium	9/12	26/62	31/26	100%
Moderate	21/24	32/64	17/12	100%
Small	62/61	16/32	11/7	100%
Total	100%	98%	100%	

Source: *Developed by the author based on [16] (Page 3)*

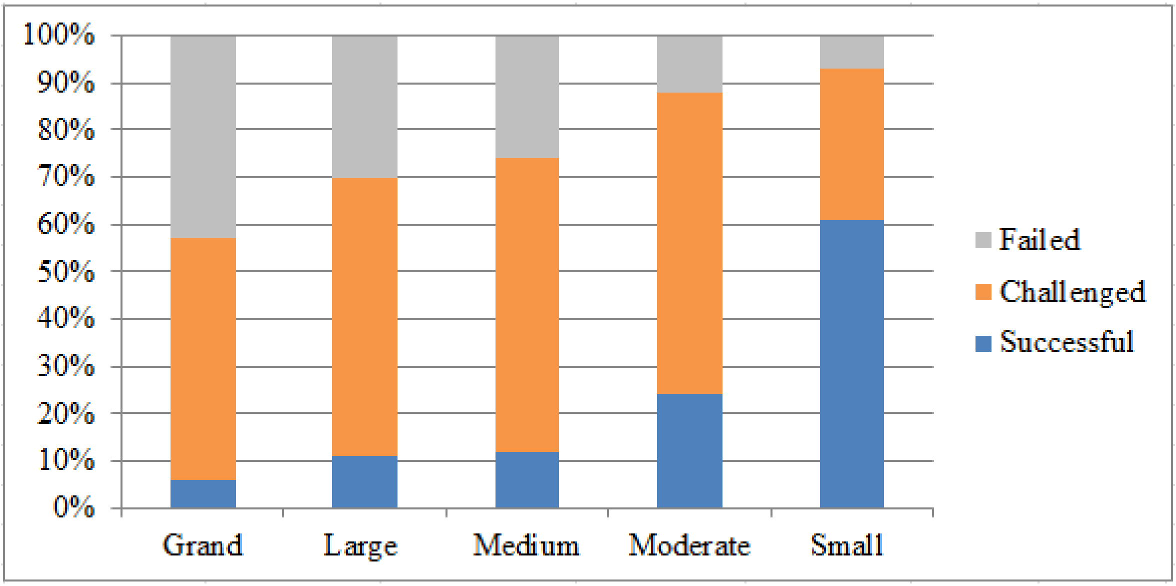

Figure 1.7. Projects resolution by size for years 2011 to 2015

Source: *Developed by the author based [16](Page 3)*

Although the research methodology has caused a strong scientific criticism, due to the impossibility to replicate the results with reliable methods, *(for example, [42])* many project managers in their experience confirm that the data is fairly accurate. The study also confirms the modern project management paradigm that all participants must be involved in the project and that the main negative impact is from large projects.

In the Chaos Report the classification of project by size was done with two definitions:

- Small project is defined as labor cost less than one million EUR;
- Large project is defined as labor cost over 5 million EUR.

But the more suggestive classification projects by size, presented in *Table 1.3*.

Table 1.3

A suggestive classification projects by size

Project size	Duration	Organization of the MP	Nr. of jobs	Nr. of subprojects	Work connectedness	Management methods
Small	<1 year	Manager	10-50	<10	Low	PMI, FMEA, PRINCE2, personal experience of the manger…
Medium	1-2 year	Manager + executive service	50-1000	10-100	Low-medium	Standard technique, PMI, SPICE, COBIT…
Large	3-5 years	Group of managers	>1000	Some thousands	High	Undefined

Source: *Developed by the author*

Small projects, have a maximum of one year, have low values, allow part-time employment, have modest or medium technology requirements and allow daily direct tracking.

Medium projects have terms ranging from 2 to 3 years, have average values, allow part/full-time employment, have medium technology requirements, and tracking is done through reports.

Large (grand) projects have long term, more than 3-5 years, high value, allow only full-time employment, high performance technology requirements, use specific tools and programs, tracking is done through control reports.

Large (grand) projects have three main characteristics:

(1) The first is a task change during the project. In large projects, the consequences can be particularly difficult, and for IPs, the initial formulation of the problem changes significantly during the course of the project. In addition, when implementing such projects, a substantial reorganization of the customer's activities is required, the functionality of the system being developed must be constantly refined and adjusted, and the customer's needs should be monitored when its business processes change. As a rule, any uncontrollable change in the functionality of the created IP or the project's work significantly affects the quality of design decisions and can generally lead to a project crash.

(2) Secondly, is the parallel management of work by different project groups, sometimes also by organizations. For example, *you can simultaneously work on the development of I&CT*

infrastructure, software development, integration of the existing enterprise IS with the system being developed, etc.

In combination with the forced permanent change these characteristics can cause an avalanche-like increase in labor costs for the project. The parallelism and multiplicity of work carried out in a large project for the development and implementation of IP greatly increase the likelihood of risks of project failure.

Thirdly, there are very big risks for the customer and for the project executor of the large project. Risk is the product of the magnitude of the damage (a change in the project time, labor costs, etc.) and the probability of its occurrence. The risks of the customer are associated with the incomplete achievement of the project objectives and the funds that are not effectively spent, and the risks of the contractor with the possibility of a sharp excess of the actual cost of work compared to the planned one. The reasons for the excess are just the first and second features.

Opportunities for reducing scale of the large projects and improving the likelihood of success include, but not only:

- Breaking large projects into incremental builds;
- Organizing teams into self-contained squads of 7-11 people;
- Restructuring the work to avoid scale and complexity.

With the take up of Agile development methods over recent years it was possible to compare project outcomes between Agile and traditional waterfall projects. Across all project sizes Agile approaches resulted in more successful projects and less outright failures.

1.2.2. Quality is dependent on type and complexity of IS

Complexity is another main reason for IPs failure. All the companies/organizations use information systems to help to perform the tasks they are specifically designed to do. The term „Information Systems” (IS) is a common definition of a wide variety of computer hardware communication technology and software designed to take care of information related to one or more business processes *(Figure 1.8).*

Information systems are computer-based infrastructures, organizations, personnel, and components that collect, process, store, transmit, display, disseminate, and act on information. So, quality of Information systems is a function of Infrastructure Quality, Software quality, Data quality, Information Quality, Process quality, Quality of organization, Quality of services. Quality management will be successful if all quality domains are under control. Obviously, treating all of them within a thesis is impossible, from which the thesis is based only on IPs product quality.

Figure 1.8. IS: a connection between environment, organization, processes, I&CT

Source: *Developed by the author*

Information systems generally provide computer-based assistance to people engaging their environment as illustrated in *Figure 1.9*, where engagements and environments are often too complex and dynamic to be handled manually *[43]*.

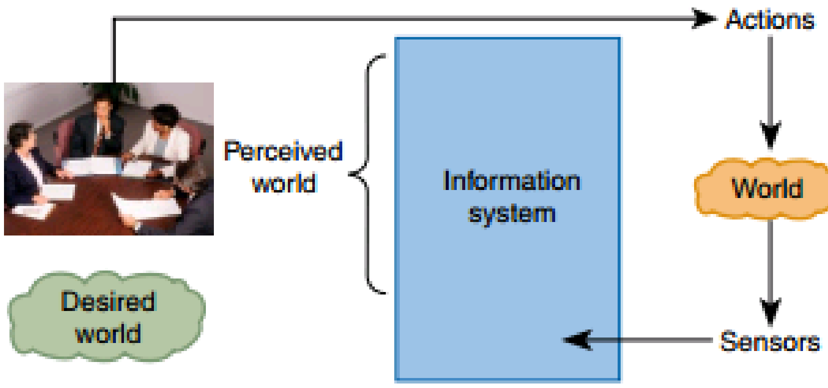

Figure 1.8. Information System context

Source: *Taken from [43] (Page 344)*

The business value of Information Systems for organization is very widespread in the literature on IT management. For more than 50 years, Information system has been controlling and managing the functioning of any different organizational tasks, from a back office administration support, to a company's strategic management tool *[44] (pp. 59-60)*. For the last twenty years, different kinds of information systems are developed for different purposes, depending on the need of the business. In today's business world, there are varieties of information systems such as transaction processing systems (TPS), or data processing System, office automation systems (OAS), management information systems (MIS), decision support system (DSS), executive information systems (EIS), Expert System (ES) etc.

The following diagram *(Figure 1.10)* illustrates the various levels of a typical IS in an organization, each with a different level of formalization. Application can be static ground breaking clearly defined or vaguely defined. The more information system is less formalized and more problematic and context-dependent structurally the information requirements are. The main domain of IS are: *TPS (Transaction Processing System), OAS (Office Automation Systems), MIS (Management Information System), DSS/ESS (Decision/Executive Support System).*

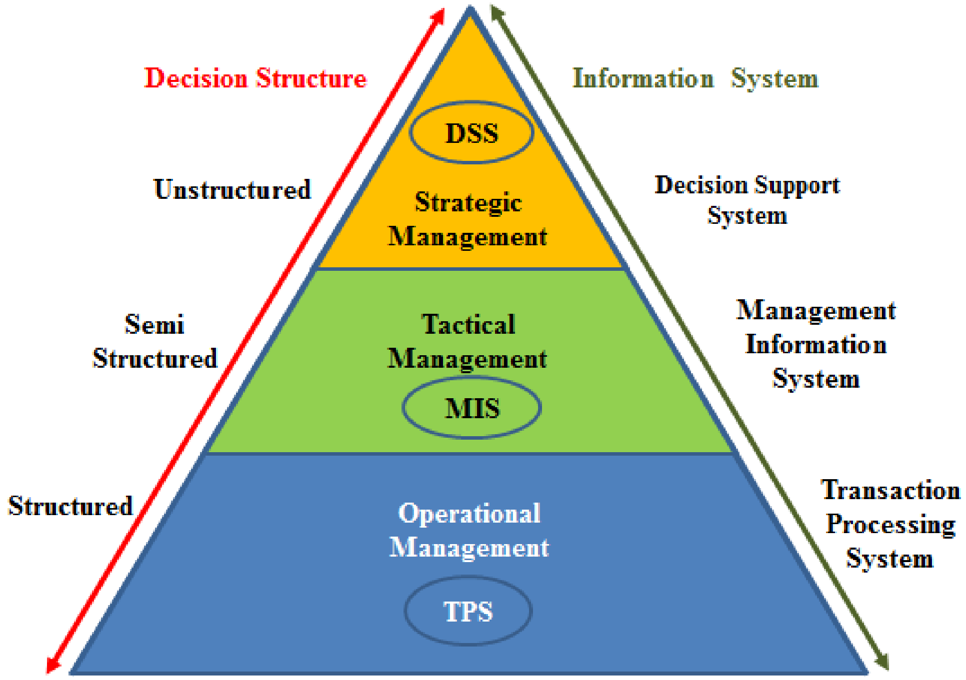

Figure 1.10. Domain of Information System and level of complexity

Source: *Adapted by the author based on [45] (Web-page) (emphasize the types of IS)*

The most popular integrated systems according to the basic function include the following ones, but are not limited to them:

- MRP I/II - Material Planning, Production Planning;
- CRP - Capacity Requirements Planning;
- ERP - Enterprise Resources Planning;
- CRM - Customer Relationship Management;
- CSRP - Customer Synchronized Resources Planning;
- ECM - Enterprise Content Management;
- SCM - Supply Chain Management, etc.

User requirements, the set of quality criteria, the weighting of the quality criteria for each of these systems can be very different. For example, the reliability requirements may be totally different for a bank transaction system and an atomic station or a space flight management application. Although IS are programmable systems where software quality is decisive, quality approach only from the perspective of software quality models does not solve the problem of quality. Different models are required for different types of IS, user contexts, etc.

Of course, the unique project products are classifiable to a certain extent, and they can impose general requirements on the products of this classifier. For this purpose, appropriate standards and guidance documents e.g. ISO 9000, ISO 9001, ISO 10006 *([26], [46])* have been developed, answering questions about the mandatory requirements of a particular class of products for the relevant conditions of use.

IS complexity is considered one of the major risk factors involved in project failure. Level of complexity and time duration of project are positively associated to failure. One way to reduce the level of project risk and failure is to reduce the level of complexity. Thus, it is obvious that in order to improve IS success rate and rate of return on investment, organizations must address the problem of complexity in IS and reduce it within limits.

In the traditional *(waterfall, cascade)* approach each IS project will grow in complexity once initiated. To meet this challenge, it is necessary to apply the Agile Philosophy of iterative and incremental development. Also, adaptation and modification of underlying organizational processes in such a way that they become conducive for automation is an issue deeply intertwined with project definition and has to be tackled in the very beginning. Once the processes have been reengineered, the scope of automation project can be fully visualized by all the stakeholders. This is the net benefit of the IS approach by Agile methods.

1.2.3. Project success and project failure critical factors

Sometime, if the long-term benefits outweigh the cost then the project is considered a success. Sometime, business value and speed to market have been tagged as the measure of project success while traditional triple constraints (time-cost-scope) are thought to have some flexibility. However, when large-scale projects fail the delays in schedule and enormous cost overruns this is considered as failure. But always the success/failure of project can and will be defined/aroused by the beneficiary. As per PMI's definition, a high-performing organization is a company that completes 80% or more projects on time, on budget, and meeting original goals. In a low-performing organization, only 60% or fewer projects hit the same marks.

Critical Success Factors of project (CSF's) are the criteria that define success. In project management, they are found in the project management plan and define what it means for the project to be successful. Time, Cost and Scope are almost always part of the CSF's, but most projects have other success criteria as well. The success of any project is determined by the balance between these three elements: time, budget, resources (people, technology, money) and goal (results) and customer/beneficiary satisfaction. At the end of the project there is a fourth factor: customer satisfaction in terms of the quality and expectations of the participants (financiers, project team members, project managers, stakeholders, public interest organizations, etc.).

Sometime, the project can be considered a complete success if it meets the technical specifications, it fulfills the mission for which it was developed, and the members of the parent organization, the client organization, the project team, as well as the end users are satisfied. This definition does not include compliance with the time and cost criteria as elements of success, because, once the activity has been completed, and if resulting product meets the needs of those who have requested it, costs and deadlines are losing importance.

Scope Statement and the CSF's derived from this should be SMART that is:

- **Specific.** If your goal is simply "to improve" I've got news for you. You probably won't.
- **Measurable.** Many wonderful goals are not easily measurable, and their success or failure gets drowned out by the debate.
- **Achievable.** There's nothing more demoralizing than being given promises that are outside of your abilities.
- **Relevant.** Ensuring the coffee is always hot and ready is a fantastic goal (in my office) but not relevant. That's an extreme example but suffice it to say that it's easy to set goals for secondary things. Keep them focused on the important performance metrics *[47]*.
- **Time-bound.** You can do everything else right, but if finish months or years behind schedule you just weren't successful.

Project management critical success factors fall into the following categories:

1. Time/Was the project completed within the allocated time period? (yes/no)
2. Cost/Did it stay within the budgeted cost? (yes/no)
3. Quality/Did it meet the proper performance or specification level? (yes/no)
4. Stakeholders/Is the result acceptable to the end user/owner? (yes/no)
5. Project Changes/Were the scope changes minimal and/or agreed upon?
6. Performing Organization/ Did it avoid unnecessary disturbance to the main work flow of the organization, or changes to the corporate culture?

The meeting of quality standards is quite often a critical success factor for a project. In fact, some projects have quality as their most critical element, such as space flight projects or nuclear reactor construction projects.

Stakeholders. It goes without saying that stakeholders need to be satisfied, but this is a surprisingly elusive goal to achieve in practice.

Project Changes. Very few projects are completed within the original scope of the project. Scope changes are inevitable and have the potential to destroy the entire project if not kept under strict controls. Scope changes must be held to a minimum and those that are required must be approved by both the project manager and the end user/owner.

Performing Organization. Also, a project should not alter the main work flow of the organization. It is not always possible to completely separate a capital project from the owner organization, and project managers should strive to manage within the strategy, policies, procedures, rules, and directives of the parent organization.

Failure of a project means failing to meet one of the criteria: cost, duration, performance/quality. Generally, the failure of a project is identified by not meeting the performance, cost, time, and usage criteria established at the time of its planning. This is true only if the criteria have been established in a realistic way. If a project manager sets unrealistic criteria under pressure from his supervisor, then the project will fail, even if these criteria have been met.

Therefore, the project manager has the obligation to set credible and realistic goals. Determining whether or not a goal is realistic is based on past experiences. In fact, the time spans of all project activities are probabilistic rather than deterministic. The workload for the project cannot be fully anticipated, the labor cost cannot be determined accurately, the forecasts become more accurate as the project is nearing completion. We must accept the variability of a project's parameters. This is inherent to any process. Over time, as the project approaches completion, variability can be reduced but can never be eliminated. The goal of a project manager is to try to eliminate the wide diversity of risks to which a project is subject or to minimize its effects.

A project cannot be considered successful if the deadlines, budget or customer requirements are violated. Sometimes a project cannot be considered successful, even if it is completed on time and within budget, but the quality of the final result does not satisfy the customer. Sometimes, on the contrary, although deadlines and budget are violated, the customer expresses satisfaction with the final result of the project. Moreover, the customer, key project stakeholders and the developer can assess the quality of the project in different ways.

Some 50 years since the inception of project management, more than 50% of IT projects still fail because they run out of time, resources, funds, etc. Detail about causes of project failure we can see in Chaos Report *([14], [15])*. Here are, for example, just a few statistics**.**

Companies that align their enterprise-wide PMO (project management office) to strategy had 38% more projects meet original goals than those that did not. They also had 33% fewer projects deemed failures. Source: (PMI's Pulse of the Profession Survey, 2017, *[48]*). According to the same research, a majority of companies either lack the skills or fail to deploy the personnel needed for strategy implementation.

Projects with effective communication are almost twice as likely to deliver project scope and to meet quality standards successfully, than projects without effective communication *(68% vs. 32% and 66% vs. 33% respectively. Source: PwC 15th Annual Global CEO Survey, 2012, [49]).* Poor estimation during the planning phase continues to be the largest (32%) contributor to IPs failures (ibid).

1.3. Project quality management and responsibilities for quality

IPs quality management is a part of general quality management. Without a QMS is impossible to solve the quality of IPs. For efficiency – all of management system (general, security, risk, quality etc.) must be integrated; all inputs and outputs – must be combined, database and quality policy – must be unique.

1.3.1. Understanding the contemporary concepts of IPs quality

In order to manage quality, one must first understand its meaning. The concepts of modern quality are codified in a single graphic image as seen in *Figure 1.11.* This graphic displays the three elements of customer focus, variation, and continuous improvement, showing the relationships and interactions among them. It also adds the essential elements of training and leadership in the Eight Principles of QMS *[50]*.

The graphic image of The Wheel of Quality discloses how all these elements interact. *Customer focus, variation, and continuous improvement are the central issues in contemporary quality*. Each is related to the others and shares a common boundary. Each is expressed through a more specific aspect of project work – respectively, requirements, processes, and controls. These aspects are not discrete, but exist as a spectrum between two extremes. Requirements may range from general needs to explicit specifications.

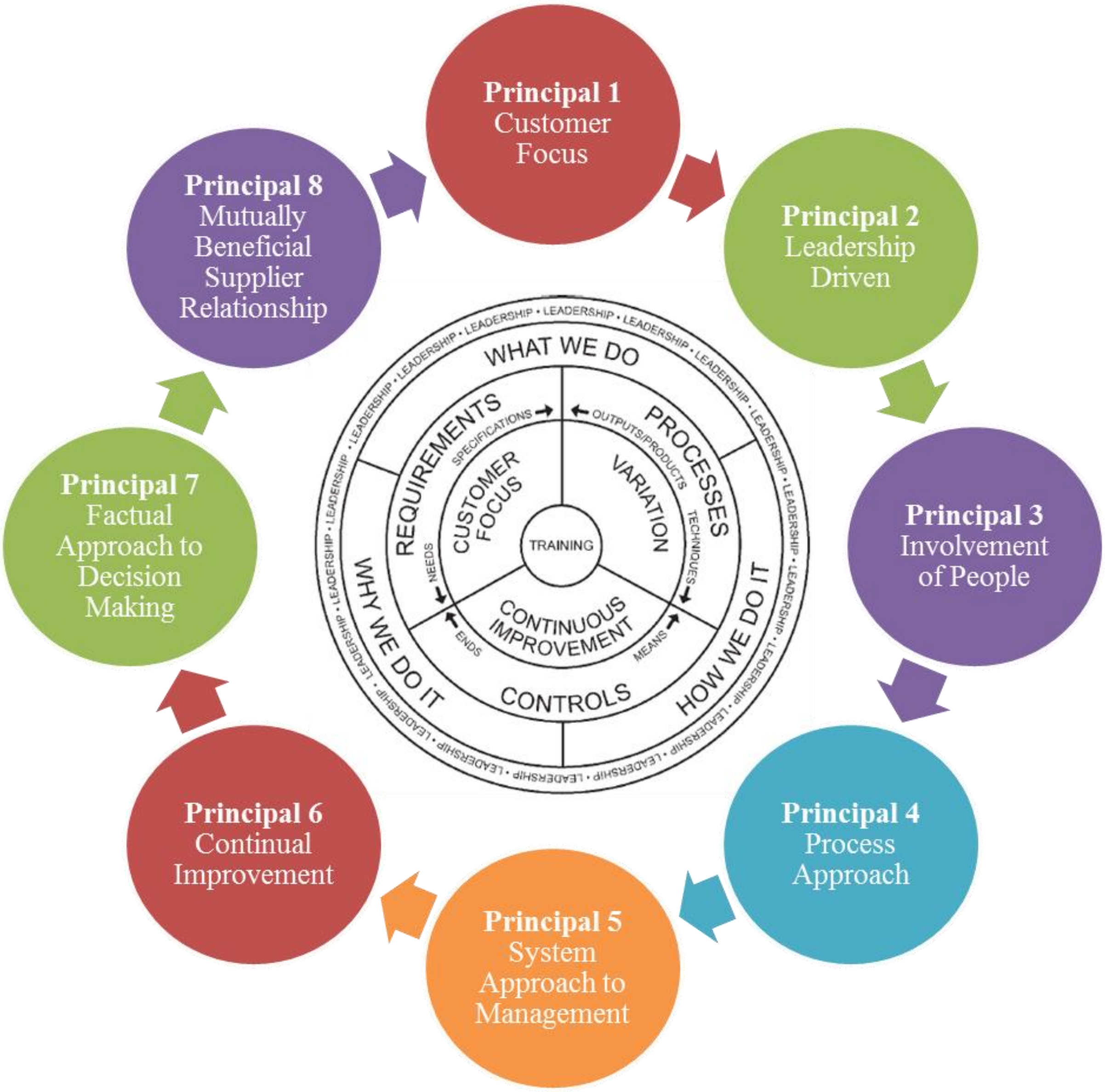

Figure 1.11. The Wheel of Quality and the Eight Principles of QMS

Source: *Adapted by author based on [34] (Page 19), [50] (Page 259) (integration The Wheel of Quality and the Eight Principles of QMS)*

Processes may be viewed from those focused-on outputs or products, which interface with the explicit specifications of requirements, to general techniques. Controls may focus on means of production, which interface with the techniques of processes, to ends of production, which interface with the general needs of requirements, completing the linkage of all three aspects. These aspects are further linked by higher level considerations in the organization that bridge the aspects two at a time. What we do bridges requirements and processes, how we do it bridges processes and controls, and why we do it bridges controls and requirements.

As the foundation of quality, continuous training is the hub of the wheel. Without training, project team members will be unable to employ the three elements effectively. Leadership holds it all together. Leadership encircles all elements, aspects, and considerations in a continuous outer loop that binds them in a unified whole.

The *main principle of project quality management* is to ensure the project will meet or exceed the stakeholder's needs and expectations. The project team must develop a good relationship with key stakeholders, especially the donor and the beneficiaries of the project, to understand what quality means to them.

Quality of IPs must be viewed in an equal level with scope, schedule and budget. If a project donor is not satisfied with the quality of how the project is delivering the outcomes, the project team will need to adjust scope, schedule and budget to satisfy the donor's needs and expectations. So, to deliver the project scope on time and on budget is not enough. And to achieve stakeholder satisfaction the project must develop a good working relationship with all stakeholders and understand their stated or implied needs.

Quality management is an essential component of the project management along with other processes. Growth and continuous improvement of performance of a project depends heavily on how to ensure proper management of quality. PQM consists of processes to ensure that the project will meet the requirements defined and planned, that quality planning processes, quality assurance and quality control. Project quality management includes all management activities that will ensure *the quality policy, objectives, and responsibilities and fulfill them through planning and improving quality through quality assurance and quality control*.

Project quality management not only refers to time and budget, but to specification of quality requirements. Prevention is preferred over inspection. It is better to design quality into deliverables, rather than to find quality issues during inspection. The cost of preventing mistakes is generally much less than the cost of correcting mistakes when they are found by inspection or during usage.

To manage quality, we need realize three processes: Plan, Management (Assurance), Control *(Figure 1.12)*.

Quality Plan. In many cases, quality is concerned with meeting the wants and needs of customers. For example Microsoft's *[51]* is conserved with „*Enhancing IT Business Value"* to „*meet customer needs*". According interchanged of data between administrations IDA is based on an experience showing that „*the most successful relationships between suppliers and purchasers are those which are defined precisely, clearly and completely (so that) the IDA project receives all items specified by the contract, to agreed standards of quality and timeliness" [52].*

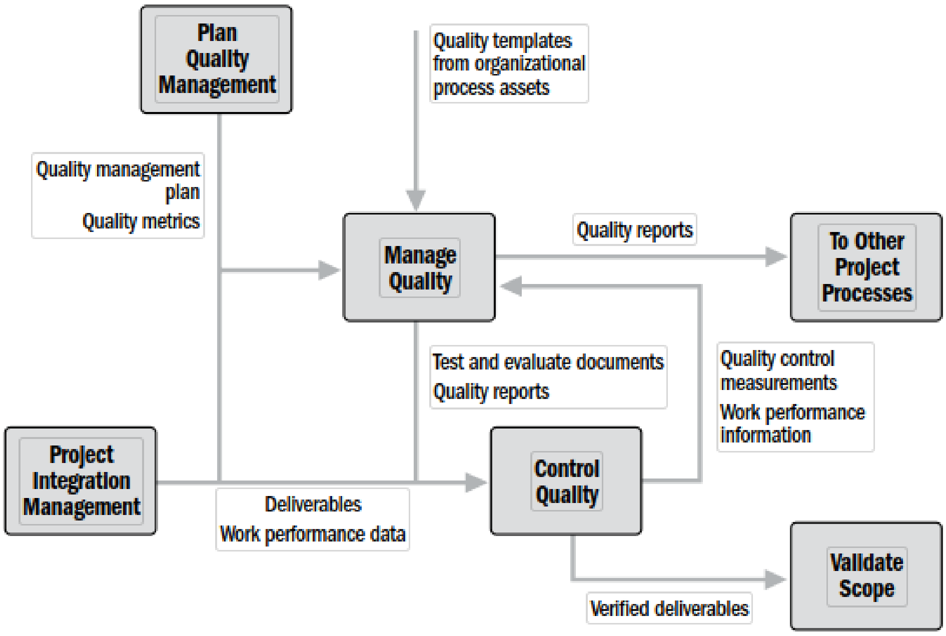

Figure 1.12. Major project quality management process interrelations

Source: *Taken form [22] (Page273)*

However, even when regarding customer concern, a more general and generic approach is usually applied using the ISO stating that quality is „*the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs*".

In short, the core meaning of quality is the fulfillment of expectations and needs, through adherence to specified requirements. However, one should note the definition of quality differentiate quality as a mean for assessing fulfillment of requirements from a *grade*, which is a mean for comparing „*products or services that have the same functional use but different technical characteristics"* (ISO). It should also be noted that this implies that the definition of quality differentiates accuracy, as a concept for measuring the difference of a measured value from the true value, from precision, which represents how the values of repeated measures are clustered *[31]*.

Perform Quality Assurance. Quality Assurance reefers to the process used to create the deliverables, and can be performed by a manager, client, or even a third-party reviewer. Examples of quality assurance include process checklists, project audits and methodology and standards development. In our discussions up to this point, we have been talking about product quality, i.e., the quality of the delivered software products. Quality assurance challenges in information project management throughout their lifecycle (from idea to implementation), require the study of development methodologies and processes for the most suitable choice with the organization's policy and elaborated project types. However, the ISO/IEC/IEEE-12207 *[29]* states that the purpose of the software quality assurance process „is to provide assurance that work products and processes comply with pre-defined provisions and plans". Quality Assurance deals in preparation of the Quality Plan and formation of organization wide standards.

Perform Quality Control. The quality assurance tasks in software development are not conducted as one-time undertakings' rather they are conducted continuously. According to literature, a continuous quality control cycle is best described as a feedback loop, of the type familiar from control systems. The object under control is the software system. The output is the quality of the software product, i.e., the degree of conformance of the software product to the requirements. The control loop measures the degree of conformance to requirements (analytic quality assurance). Based on the deviations (i.e., quality deficiency) corrective action is taken, which requires communicating the desired requirements (constructive quality assurance) and the deviations. Perform Quality Control includes also, verification if we follow the quality standards and if we meet the quality standards. In this context, a standard means requirement, procedures, records, documentation, etc.

1.3.2. Cost of quality and continues improvement

Cost of quality is the cost of a product that can be saved if all the performers work flawlessly. Much misunderstanding exists about quality in spite of the various definitions in circulation. Quality is many things to many people, but quality is also not some things that have been assumed over time, the quality level and the cost.

PMBOK *[22]* describes five levels of increasingly effective quality management as follows:

1) Usually, the most expensive approach is to let the customer find the defects. This approach can lead to warranty issues, recalls, loss of reputation, and rework costs.
2) Detect and correct the defects before the deliverables are sent to the customer as part of the quality control process. The control quality process has related costs, which are mainly the appraisal costs and internal failure costs.
3) Use quality assurance to examine and correct the process itself and not just special defects.
4) Incorporate quality into the planning and designing of the project and product.
5) Create a culture throughout the organization that is aware and committed to quality in processes and products.

QM standards talk about continuous improvement in the quality of processes, products, etc. But in essence, it is about *balancing between the required quality level and the cost of quality.*

According to the classic Juran model as shown in *Figure 1.13* an increase in expenditures on prevention and appraisal ensures a decrease in the percentage of defects, that is, a higher quality level.

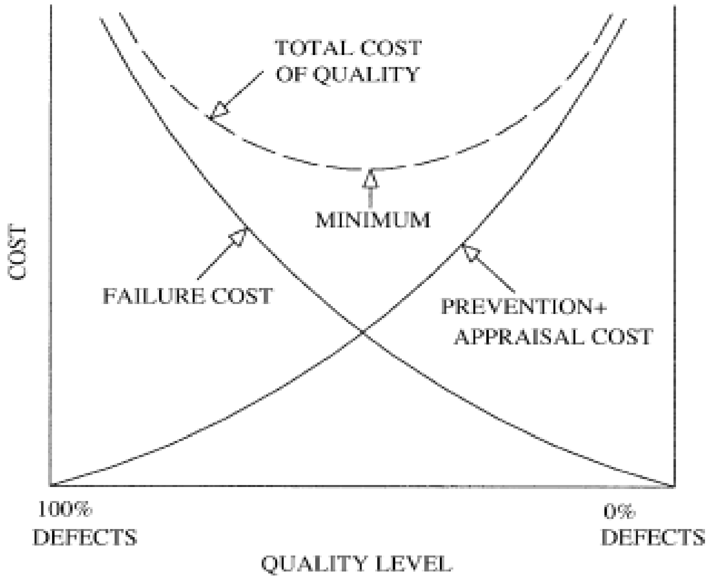

Figure 1.13. Cost versus quality level - classic model of Juran view

Source: *Taken form [53] (Page 1359)*

In contrast, an increase in the percentage of defects is accompanied by increased failure costs. Accordingly, there is an inverse relationship between prevention and appraisal effort and failure cost (Brown and Kane 1984) *[53]*.The resulting total quality cost curve *(Total cost of quality)* drops initially to the level of an optimal cost *(the intersection points of Prevention + appraisal effort and Failure cost)*, and then begins to increase as the quality level increases.

Optimal quality is to ensure the utility of the product under conditions of total cost optimization *(both to the producer and the beneficiary).* Exclusive consideration of intrinsic qualitative features, with neglect of the economic aspect, refers mostly to luxury products or those that require a very high reliability.

Except for all products, there is an *optimal quality defined by the maximum difference between the overall economic effect resulting from the increase in the quality level and the expenses incurred to achieve this level of quality.* Or, conventional wisdom, perhaps better called „conventional ignorance" in this case, has it that *better quality costs more.*

In times of cost control and cost cutting, the answer to quality improvement can be an unwise „We can't afford that". Philip B. Crosby, another quality pioneer, addressed this in a book entitled „*Quality is free" [54].* Briefly, his point of view was that quality does not cost, it pays. When you improve the quality of a process, you reduce the defects that result from that process.

While the new process may be more expensive – it may be less expensive, too – the resulting reduction of defects is something that pays back over and over and over. So, if the payback is more than the cost, as it often is, quality is essentially free.

Expensive product. This may be the greatest misunderstanding of all because of the tendency to view quality in price of products.

An automobile with leather seats and little mechanical wipers on the headlight costs more than one without these features. A fine „writing instrument" costs more than a plastic ballpoint pen. But *price does not confer quality.* If the definitions of quality are reviewed, we can observe that none of them mentions price. Quality arises from an ability to satisfy customer needs. If a customer's goal is to spend a lot of money, then an expensive product may be viewed as top quality. *Customers generally seek the lowest price for a product that meets their functional needs, not the highest.* Considering accuracy and maintenance, an inexpensive digital watch from a drugstore provides better quality than a more expensive mechanical watch from a jewelry store. A customer may want the jewelry item, but only because it serves a purpose other than timekeeping, not because it is a better-quality watch.

Time consuming. „We don't have time" is the response that condemns an organization to poor quality. Urgency prevails and shipping dates or field requirements rule. The reality is that we always have time; we just choose not to use it wisely. The old adage „There's never enough time to do it right, but always enough time to do it over" is not just a clever collection of words; it is the truth. *Poor quality in production leads to rework.* Delivery of poor-quality products leads to replacement, warranty charges, lost customers, and loss of reputation.

In the long run, quality saves time and much, much more [34]. Improving the process of developing and implementing software significantly reduces the relative inconsistent cost of quality while keeping the agreed cost at the same level.

The goal of the thesis is providing quality of information projects, considering the high-quality requirements at low costs, in according with the specific of the software development organizations and the specific of the user organizations.

The thesis assumption is to develop a new approach, based on a generic, extensible, flexible and adaptable quality metamodel, which includes the best knowledges *about* quality factors *(quality characteristics and metrics),* extracted from known basic quality models and quality standards, and which permit realize a tailored quality models for any concrete type of informational project.

At baseline, the research examines the software development framework Agile, the existing software quality models and standards, including but not only: *current ISO/IEC 9001:2015,* *Quality management systems, Requirements; Total Quality Management philosophy, principles; ISO/IEC 25010:2013, Quality in use model, Quality of Product model; ISO/IEC 25012:2008, Quality of Data model etc.*

The measurement of *Information quality products, Data quality* and *Quality in use* of information products, are based on the best practices, such *Systems and software quality requirements and evaluation (SQuaRE),* which include *ISO/IEC 25022:2016, Measurement of quality in use, ISO/IEC 25023:2016; SQuaRE, Measurement of system and software product quality and ISO/IEC 25024:2015, SQuaRE, Measurement of data quality.*

The research objectives include:

- Critical analysis of the existing IPs quality assessment frameworks, models, standards and identify the quality approach framework based on the best practices presented on the actual international management standards, considering the specific context of the organization and the most appropriate IS/software development methodologies to provide the quality of IPs.
- Developing a reliable and efficient approach that permit manage IPs quality *(measure, assess, improve)* through the development life cycle taking into account internal context of development and the modern development approach (Agile), the best practices recommended by SQuaRE standards, CMMI and others.
    - ➢ Developing a generic quality metamodel with the best knowledge about quality factors;
    - ➢ Identifying the most important quality characteristics for some more widespread classes of IPs, based on literature analysis, performing on-site research (survey) to verify the relevance of the selected quality characteristics;
    - ➢ Building a tailored quality models for quality assessment in according with concrete type of system and concrete organization context;
    - ➢ Developing the original software application for support of new approach, with the extraction of some initial data directly from the Agile PMO, or from measurement templates.

The main research problem is the different quality conception/framework by different organizations, for different types of information projects/systems and their outputs. Existing basics quality standards/models are too abstract and a high level of abstraction, which eliminates their practical usage. On the other hence, the author/particular models with high degree of detail have a narrow applicability and eliminates their widespread usage. An others problem of repeatedly evaluating and improving of quality over the life cycle – is time and resources consuming task.

Research problem solving.

Firstly, existing quagmire of IPs quality frameworks, standards, models, objective and subjective quality assessment characteristics and metrics have been reviewed, have been discussed and gathered in the generic metamodel. Thus, the first decision related as appropriate choice of the quality assessment framework of IPs, can be easily solved.

Secondly, realized extensible, flexible and adaptable quality metamodel, which integrates the knowledge about known quality models, quality factors, and the best practices, presented on the actual international standards is universal, and allows the generation of particular quality models for any type of project and any type of organization.

Obviously, the construction of particular quality models requires prior study and understanding of the internal context of the developer and the requirements of the beneficiary. Taking this into account and based on literature analysis, were identified the most important quality characteristics and their weights for seven more widespread classes of IPs: *ERP & CRM, GIS & Map Library, Enterprise Portal & Knowledge Management, Business Intelligence & Big Data, Internet Site & Web Applications, Document Management Systems, Mobile Applications.*

The relevance of the selected quality characteristics for these seven tailored quality models was confirmed by on-site research (survey of expert in these domains).

The assessment of the quality of IS/Software depends essentially on the initial data – the quality of the measurements. But this is a bulky routine. In order to solve this problem, in Excel three templates of collating of the initial data were made, with the definitions of the quality characteristics and of the calculation formulas incorporated, based on the respective quality metrics. These three developed templates are in according with basics quality models: ISO/IEC 25010:2011 Quality in use and Quality of Product, ISO/IEC 25012:2008 Quality of Data.

A new developed approach permits continuous assessment of the quality of IPs along the lifecycle that can be systematically measured, calculated, managed and improved. But even in the new approach the process of repeatedly evaluating quality over the life cycle is time consuming and requires a lot of resources. To manage quality of project along lifecycle, assess and visualize measurement values and trends, was developed the software application IPMS (Information Project Management System), which automate the routine work.

The new assessment approach and the realized software tool IPMS are implemented in „WGS", Israel and in the study process of the Moldova State University, and could be easily implemented in others organizations. Thus, the research provides a complete solution, which consists of realized framework and developed software tool IPMS for repetitive quality assessment and improvement along development lifecycle of IPs.

Conclusions on chapter I

Two main needs arise from background analyze of IPs development in KBS:

- *Provide project managers a better understanding and methodologies for assuring information projects quality.*
- *Increasing the quality of information projects along lifecycle, focusing on information systems and application software.*

The quality of IPs as minimum includes three aspects: quality of project management (processes), quality of resulting product (IS, software applications) and quality of information, resulting by data processing. Quality of product is a complex and multivalent concept, it means different things to different people, it is highly context dependent, and, as rule, it is a subjective estimation.

*The **thesis assumption** is to create a metamodel, which will permit realize a tailored model for each group, or range of values which are acceptable for similar people and realize an original digital support for them.*

There are many other **challenges of IPs quality providing**:

- *Software can't be physically observed;*
- *The lack of knowledge of client needs at the start, but often this is impossible;*
- *The rapid rate of change on hardware, software and of client needs, which are* inevitable*, generate a lot of extra work, and improvement of quality has a cost, etc*.

Thus, **the first decision regarding the quality of IPs** is related to the appropriate choice of methodology (approach) for IS/software development, which determine, to a great extent, the success of the project. The project organizations (managers) will need to understand, first of all, the development environment with its relation to the project management and the quality processes and tailor all of this to determine the way project quality management processes are applied for continuous improvement a quality of deliverables. And project quality management aims to identify the required project quality, assess and control it, and finally attain the optimum results through specific processes and activities.

In accordance with the above facts and trends, there is a strong need **to develop and/or choose the right approach for project management and good tailored model for quality improvement**, especially for medium and large IPs/IS, as more prone to risks of project failure.

Order saves time. (J. Goethe)

Order frees the thought. (S.P. Korolev)

Chaos always conquers order, for it is better organized. (Terry Pratchett)

II. METHODOLOGICAL APPROACHES OF PROJECT QUALITY MANAGEMENT

There are many approaches to Project Quality Management (PQM). Understanding and proper selection of project and quality management methods, models, and tools is one of the key aspects in the management and quality assurance of IPs. This chapter describes the methodological approaches of project quality management (1) in accordance with the PMI and ISO/IEEE standards requirements and (2) treatment PQM as a part of corporate-wide quality management system (QMS), which is based on ISO 9000:2015 and ISO 10006:2017standards. The advantages and disadvantages of each approach in treating of IPs quality are examined.

2.1. General framework of quality and important bodies in the domain

The term „framework" refers to models and standards that have a variety of issuing bodies, scopes, architectures, and rating methods. These frameworks include General Total Quality Management (TQM) philosophies such as those of Deming [55; 56], Juran [57], and Crosby [54], many standards and quality models, some of which will be briefly examined further. In this section we try to bring more understanding to the use of standards appropriate to the company's needs.

Don't judge a book by its cover (proverb)

2.1.1. PQM in accordance to the ISO/IEEE standards and PMBOK from PMI

Today quality orientated process approaches and standards are maturing and gaining acceptance in many companies. One of the best quality management solutions is to use the best international experience, which is reflected in *industry, national and international management standards. Using standards means forces analysis of quality management activities*. In the absence of a disciplined form of management, quality can be one of those things assumed to be achieved. It documents all aspects of the quality management system again, no assumptions or promises, only facts. It focuses on prevention, not inspection. But the main benefits of implementing an ISO Management Standard and of ISO certification includes giving your organization credibility, making a business more **efficient**, **reducing costs,** improving your **customer satisfaction** levels. In many cases, without an ISO certification, it is impossible to participate in **tenders** or realize a good potential customer. *The importance of standards consists in encapsulation of best practice, avoidance repetition of past mistakes.*

The main problem with standards lies in the fact that it is difficult to relate standards of many bodies to business needs and to justify the application of the international standards in their operations.

PMBOK from Project Management Institute (https://www.pmi.org/), is the standard de facto in PM**.** PMBOK® Guide, shortly PMBOK *[22]*, is an acronym for *„A Guide to the Project Management Body of Knowledge".* PMBOK is the sum of project management expertise, the most important publication of PMI in the field of PM, with chapter 8 entirely dedicated to quality management. Since the first edition in 1996, *in* 2017 PMBOK *has reached the 6th edition*.

This PMBOK edition cover more content on **Agile project management** as Agile has fast become one of the fastest growing project management methodologies these years. **Agile** „*A practical guide to using Agile" [23]*, is highlighted in a separate book and fully correlates with both the 6th and 5th (already previous) version of PMBOK. *These are two fundamental books for PM, including quality management, regardless of the field and the thymus of the project. For any field, project management is no different, differs only resulting project products.*

The American National Standards Institute (ANSI, https://www.ansi.org/) is the U.S. member of ISO. The American Society for Quality (ASQ, https://asq.org/), the global voice for quality, is a member of ANSI and is responsible for QM standards. It publishes standards in the ANSI/ISO/ASQ-Q9000 series that are the U.S. equivalent of standards published by ISO.

International Electrotechnical Commission (IEC, https://www.iec.ch/), develop International Standards and Conformity Assessment for all electrical, electronic and related technologies, inclusively I&CT. The IEC works with its sister international standardization organizations, ISO and ITU (International Telecommunication Union), on a bilateral basis in specific technical areas, and under the tripartite World Standards Cooperation.

European Telecommunications Standards Institute (ETSI, https://www.etsi.org/) provides members with an open, inclusive and collaborative environment. This environment supports the timely development, ratification and testing of globally applicable standards for I&CT-enabled systems, applications and services.

The ISO (International Organization for Standardization (ISO) of quality standards), (https://www.iso.org), an international standard-setting body composed of representatives from various national standards organizations, founded on 23 February 1947, promotes worldwide proprietary, industrial and commercial standards. ISO is the world's largest developer of voluntary international standards and facilitates world trade by providing common standards between nations. Over twenty thousand standards have been set covering nearly everything. The organization’s short title „ISO” is not a fractured only as acronym, but rather an adaptation of the Greek word „isos”, which is translated into English as „equal”.

ISO Quality standards try to focus your organization on saying what you do and then doing what you say, while at the same time keeping the customer in mind. *For many standards exists checklist.* The purpose of the checklist is to define clearly all the *policy, procedure, plan, records, document, audits or reviews that the underlying standard calls out, either required or suggested*.

The main idea of all standards is to eliminate variation within your organization, and checklist help to achieve of this. However, the situation is compounded by the many complex regulations and versions that make it difficult to implement them in concrete context (*Fig.2.1).*

The theoretical background of the thesis poses some main needs regarding *regulatory framework of the projects and quality management, quality models of software products*. It is because there are numerous quality frameworks around the world. Considering the specifics of the fields of activity such telecommunications, transport, health and/or the specific platforms of activities, application areas etc., it is estimated that *the number of standards is several hundreds*.

Here the natural question arises, *which family of standards is current and is necessary to ensure the quality management of software systems and software projects*. What standard is it worth to rely: on ISO 9004, ISO 9000-3 or ISO 9001; on the ISO 9126 or ISO 25000 series; on CMM (Capability Maturity Model), CMMI (CMM Integrated), etc.?

Quality assurance challenges in information project management throughout their lifecycle (from idea to implementation) require the study of development methodologies and processes for the most suitable choice with the organization's policy and elaborated project types.

ISO/IEC/IEEE 12207:2017, *Systems and software engineering [29] Software lifecycle processes*, help ensure the effective application of the organization’s Quality Management process to the project, describe QM activities and tasks such *Plan QM, Evaluate QM, perform corrective and preventive action etc.*

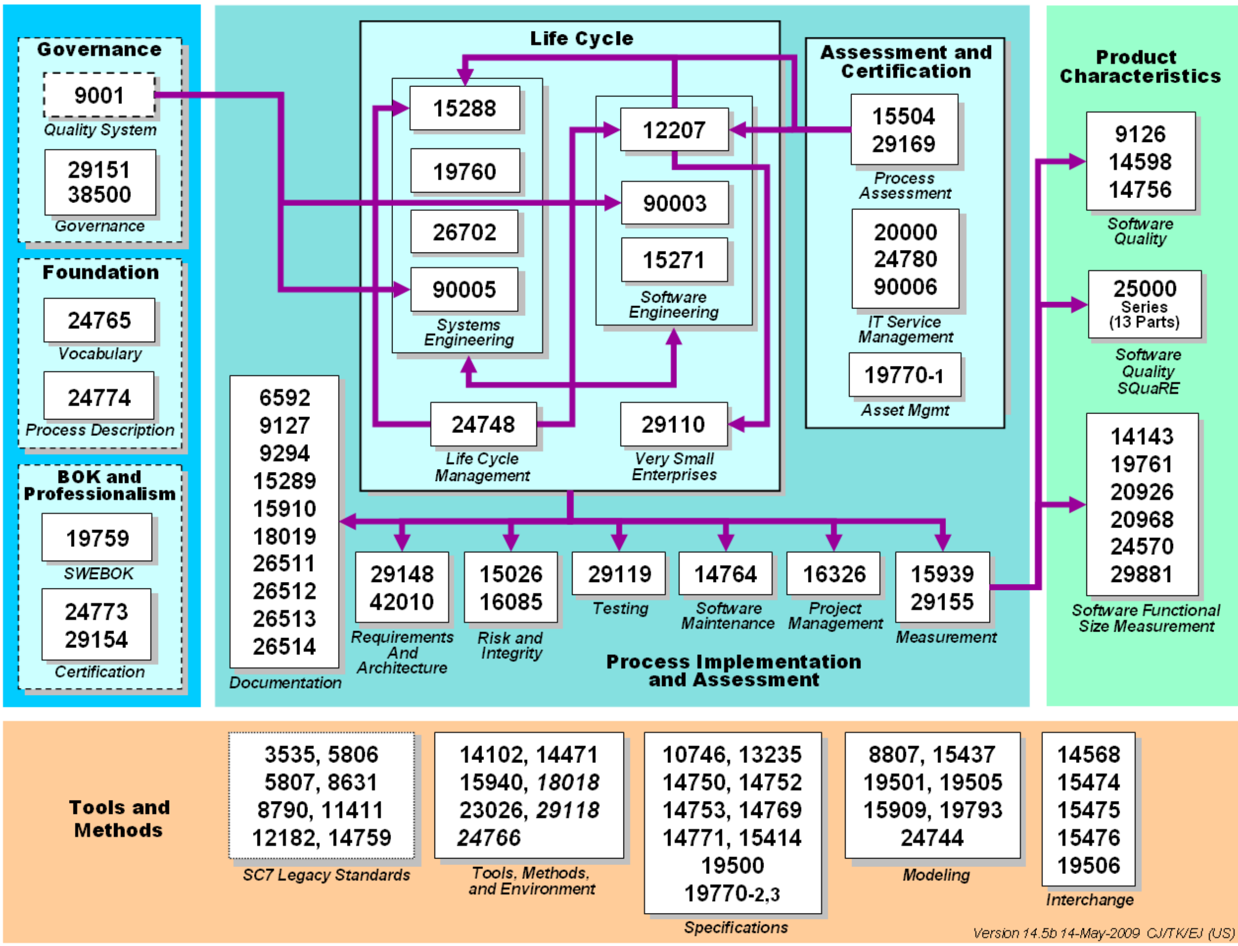

Figure 2.1. The quagmire of SC7/I&CT quality standards

Source: *Taken from [58] (Presentation, page 23)*

New and revised standards for Quality include, but not only:

IEEE 730:2014 IEEE Standard for Quality *Assurance Processes*

- ISO 10006:2017, Quality management. Guidelines for quality management in projects
- ISO 14001:2015, Environmental management system, is similar to ISO 9000, both refer to the process of production of a product, and not to the product itself.
- ISO 21500:2012, Guidance on project management, PMBOK 6th Edition closely follows this standard;
- ISO 45001:2018, ISO/IEC 25040, SQuaRE, Evaluation process, composed of five activities. Provides a process description for evaluating quality of software product and states the requirements for the application of this process;
- ISO 90003:2018, Software engineering, Guidelines for the application of ISO 9001:2015 to computer software, to the acquisition, supply, development, operation and maintenance of computer software and related support services, originally published as ISO 9000-3;

- ISO 9001:2015, Quality Management System, Requirements.
- ISO/IEC 15504:2012 *[59]*, also known as SPICE (Software Process Improvement and Capability Determination), is a framework for the assessment of processes *(part 5)*. The methodology for creating quality metrics in this way is approved in SQuaRE;
- ISO/IEC 25022:2016, SQuaRE, Measurement of quality in use *[60]*, includes measures of effectiveness, efficiency and satisfaction), replacing ISO TR 9126-4;
- ISO/IEC 25023:2016, SQuaRE, Measurement of system and software product quality *[38]*, includes measures for usability attributes, replacing ISO/IEC TR 9126-2 and ISO/IEC TR 9126-3;
- ISO/IEC 25066:2016, SQuaRE, Common industry Format for Usability (CIF) *[61]*;
- ISO/IEC 27001:2017, Information security management systems, Requirements *[28]*;
- ISO/IEC 29110:2016, Guides for Very Small Entities (VSEs)/for Software/System Engineering *[62]*.

2.1.2. Crossroads of Project management and Quality management

The main principles of the Project Quality Management (PQM) described in PMBOK:2017 *[22]*, ISO 9001:2015 *[25]* and ISO 10006:2017 *[26]* are very close. They are based on the theory of management, which has been developed over the past 30 years, concentrating the best practices of QM for last three decades.

Project Quality Management *[63]* is the process required that ensures that the project meets requirements and expectations of the beneficiary involved in the project consists *of identification of relevant quality levels for the project and how to meet them, planned activities implemented quality system intended to ensure that the project will be within the parameters of quality planning, monitoring results of project activities and assessing their quality standards, ways to eliminate the causes which led to unsuccessful and continuous improvement.*

PQM addresses both the management of the project and the product (deliverables) of the project. PQM applies to all projects, regardless of the nature of their deliverables. Quality measures and techniques are specific to the type of deliverables being produced by the project. Failure to meet quality requirements in either dimension can have serious and negative consequences for any members, executors or all of the project stakeholders. PQM includes the processes and activities that determine *quality policy, objectives & responsibilities* to ensure that the project satisfies the needs for which it is undertaken.

The main principles of PQM in PMBOK *[22]*, in ISO 9001:2015 *[25]* and in ISO 10006:2017 *[26]* consist in:

- Orientation of the Company's activities to the client needs;
- Management's responsibility to create an enabling environment for quality and continuous improvement of the QMS;
- Project presentation as a set of planned and interrelated processes;
- Focus on the quality of products and services as a necessary condition for meeting the objectives of the project;
- Process presentation of all activities;
- Systematic approach to project management in general.

Crossroads: Project Management Institute, PMBOK, Chapter 8, Project Quality Management; American Society for Quality (ASQ), QBOK – Quality Body of Knowledge, section III C, Project Management; ISO Standard 10006:2017 for Quality in PM, Guidelines for quality management in projects, is a subsidiary standard to ISO 9001:2015, cannot be used for certification, but it can be used to audit a project.

PMBOK of PMI focuses the functional scope of the PQM on the standards of the 9000 and 10000 series. *Figure 2.2* provides an overview of the PQM processes.

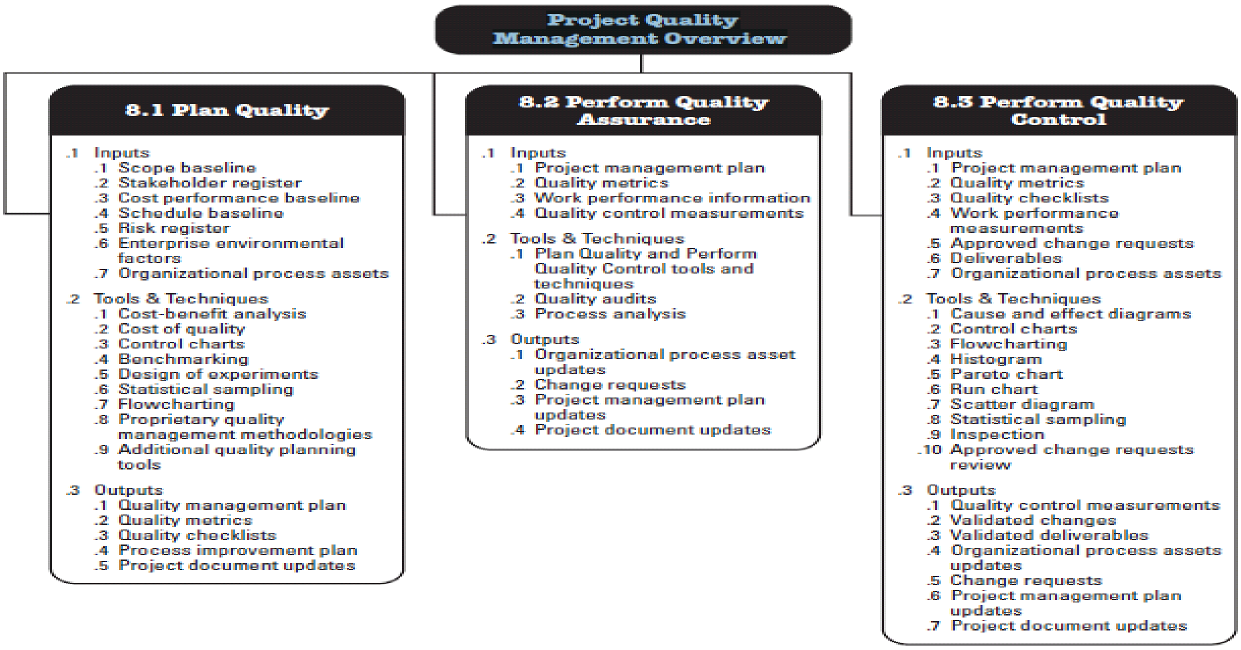

Figure 2.2. Project quality management overview

Source: *Taken from [31](Page 191)*

In areas of intersection with groups of management processes it is recommended to organize the implementation of three processes: *quality management planning*, *quality provision* *and quality control*. According to requirement of ISO standards and to the PMBOK, „PQM includes the processes and activities of the performing organization that determine quality policies, objectives, and responsibilities so that the project will satisfy the needs for which it was undertaken” *[31]*.

- Plan Quality: The process of identifying quality requirements and/or standards for the project and product, and documenting how the project will demonstrate compliance.
- Perform Quality Assurance: The process of auditing the quality requirements and the results from quality control measurements to ensure appropriate quality standards and operational definitions are used.
- Perform Quality Control: The process of monitoring and recording results of executing the quality activities to assess performance and recommend necessary changes.

Although the processes are presented as discrete elements with well-defined interfaces, in practice they may overlap and interact. Over the years several methodologies for quality management were presented among which are Total Quality Management/TQM, Six Sigma, Failure Mode and Effect Analysis/FMEA, design reviews, voice of the customer, Cost Of Quality/COQ, and continuous improvement, all of which are compatible with the International Organization for Standardization (ISO) principles regarding quality management *[31]*:

- Project must provide **stakeholders satisfaction**. This is achieved by understanding, evaluating, defining, and managing expectations so that requirements are met.
- **Prevention over** inspection *(Figure 2.3)*. The cost of preventing mistakes is generally much less than the cost of correcting them after *inspection.* Hence, quality is planned, designed, and built in-not inspected
- **Continuous improvement** as an implemented vision of all processes. In the case of TQM and Six Sigma continuous quality improvement initiatives focus on project’s management project’s product while other models, such as Organizational Project Management Maturity Model (OPM3) or Capability Maturity Model Integrated (CMMI) focus on process improvements.
- **Management responsibility**. Success of the projects is hard to achieve without management commitment and participation.

We need to prevent failures. The cost of a single defect can be significant; the cost of several defects can be catastrophic. Thanks to the fact that the majority of bugs are introduced during coding, but not discovered until a later phase, it becomes important to understand the difference it costs to fix defects at each phase of development. The early detection of defects is important for the successful execution of an IP. However, the detection and prevention of defects is a significant challenge in the software industry. The reworking process costs more than the initial process so early detection of defects during the design and requirements phase is necessary to avoid this extra expense. A large number of defects usually occur in the initial stages of a project and early defect detection will lower the overall cost of the project *(Fig. 2.3)*.

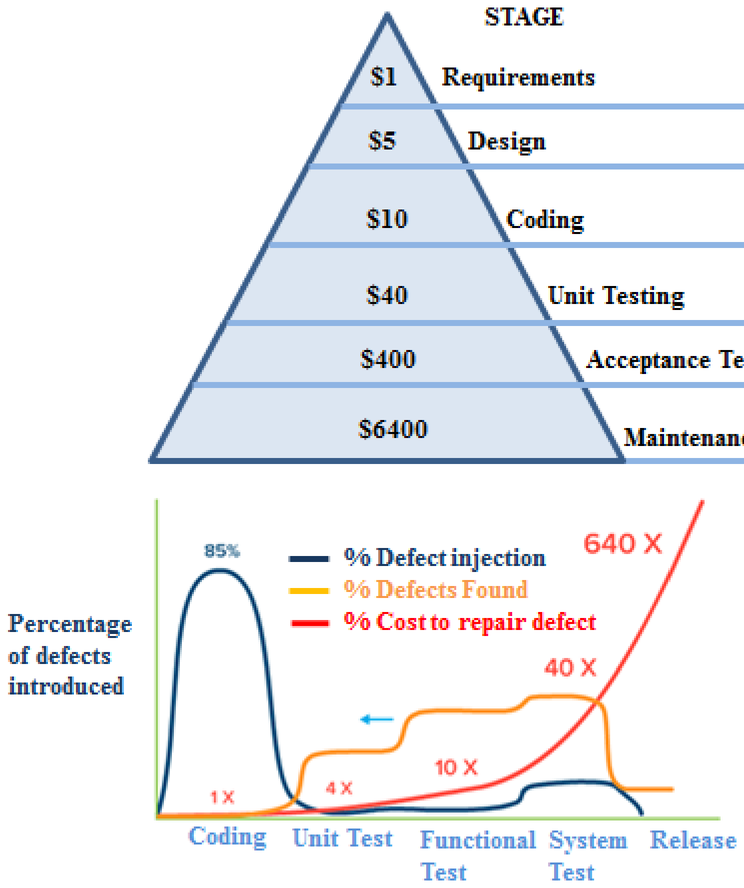

Figure 2.3. Relative cost to repair a defect at different project lifecycle stages

Source: *Combined design based on [64] (First Edition, page 21), [65](Web-page)*

Also, the Systems Sciences Institute at IBM has reported that the cost to fix an error found after product release was four to five times as much as one uncovered during design, and up to 100 times more than one identified in the maintenance phase *[66]*. According to the data gathered by the Hewlett Packard company, the cost of correcting a bug detected at the last stage of the project life cycle (deployment and maintenance) is 30-100 higher than the` cost of the same bug detected at the first stage *[67]*.

The ISO standards clearly separate the concepts of *quality* and *variety*. The PMBOK provides an interpretation of the concepts of *accuracy* and *precision*. In addition to the generally accepted ISO principles of TQM, PMI also introduces elements: *management responsibility of quality* and *cost of quality*. It is about the potential of a separate investment mode with respect to ensuring the required characteristics of the project product. For this purpose, a somewhat broader interpretation is given of the application of basic tools to different process management groups.

ISO 10006:2017 addresses the concepts of both „quality management in projects” and „quality management systems in projects”. In accordance with the standard, quality in project management consists of the quality of the project processes and the quality of the products produced.

ISO 10006:2017 vs. PMBOK (6th edition, 2017):

- Similar definitions of project,
- ISO – customer focus, but PMBOK – stakeholder focus,
- ISO – seven process groups, but PMBOK – five process groups.

ISO emphasizes alignment with strategy, Quality planning more succinct in ISO, mutual benefit supplier relationships, emphasis on management involvement in the quality of the project PMBOK is cumbersome for that purpose. ISO standard are complementary to other standards, such projects IN Controlled Environments (PRINCE2 *[68]*), IPMA Competence Baseline (ICB3 *[69]*).There are similarities between the standards, but there are also differences *[70]*.

Modern trends and good practices for PQM, as a part of ISO standards and PMBOK, *seek to minimize variation and to deliver results that meet defined stakeholder requirements, include*, but are not limited to:

- **Customer satisfaction**, which requires a combination of conformance to requirements and fitness for use. *In Agile environments, stakeholder engagement with the team ensures customer satisfaction is maintained throughout the project.*
- **Continual improvement,** aided for plan-do-check-act (PDCA) cycle, is the basis for quality improvement as defined by Shewhart and modified by Deming *[71]*.
- **Management responsibility**, for project success requires the participation of all members of the project team.

All of this are compatible with ISO 9000 and ISO 10000 series of standard guidelines, and proprietary approaches to quality as recommended by Deming *[55]*, Juran *[57]*, Crosby *[54]* and nonproprietary approaches for continuous improvement such as TQM. New approaches to quality management including the TQM movement are developed by Dr. W. Edwards Deming. Many of the underpinnings of Agile come from lean manufacturing and TQM. In a manufacturing environment, companies learned many years ago that running a manufacturing plant like a sweatshop and forcing workers to work an excessive number of hours under poor conditions does not often result in high-quality products.

Taking a broader view, the **PMBOK** *[22]* describes *three processes of quality management*: *quality planning, quality assurance, and quality control*. The Juran Trilogy *[72],* also describes *three slightly different elements*: *quality planning, quality control, and quality improvement*. Juran's view includes *assurance* and *control* activities within quality control. It also adds the essential element of quality improvement, which the PMBOK does not include as a distinct process. *A better approach is to combine between these three views to include quality planning, quality assurance, quality control* and *quality improvement*. This is in according with the requirements of PMBOK, ISO 9001 and ISO 10006 standards *Figure 2.4.*

Figure 2.4 illustrates the combined approach model of the quality management system in projects: *Quality Planning (PLAN), Quality Assurance (DO), Quality Control (CHECK)* and *Quality Improvement (ACT).*

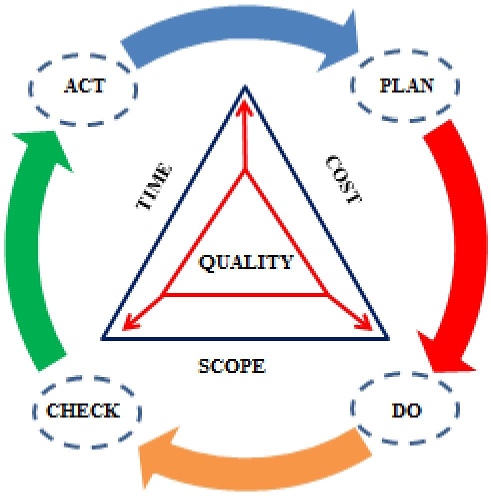

Figure 2.4. Project quality management as continuous PDCA process

Source: *Adapted by the author based on [63] (Chapter 3, page 43) (emphasize the quality element in the triangle)*

„Quality in a service or product is not what you put into it. It is what the client or customer gets out of it" – Peter Drucker. However, there lies a possibility where this statement could be polished. It could be argued that if the customers will get a quality service or product, it is due to the reason that quality is put into it. In order to put quality in any product, one needs to intensely identify and analyze the requirements from various level of the product development. Product development usually starts from the initial market study till the final product support with various requirements. Based on these requirements, the implementation has to be made. Also, ***the product has to be evaluated over different measures of quality, known as quality attributes*** /***quality factors/quality characteristics***. There are several factors, by the help of which the quality could be enhanced. As in general, the definition of quality for any service or a product is similar.

However, *the way of evaluating the quality for products depends on the requirement of its customers as well as the product itself*. Quality management is the process for ensuring that all project activities necessary to design, plan and implement a project are effective and efficient with respect to the purpose of the objective and its performance.

Project quality management is not a separate, independent process that occurs at the end of an activity to measure the level of quality of the output. It is not purchasing the most expensive material or services available on the market. *Project quality management is a continuous process that starts and ends with the project.* It is more about preventing and avoiding than measuring and fixing poor quality outputs. It is part of every project management processes from the moment the project initiates to the final steps in the project closure phase.

PQM focuses on improving stakeholder's satisfaction through continuous and incremental improvements to processes, including removing unnecessary activities; it achieves that by the continuous improvement of the quality of material and services provided to the beneficiaries. It is not about finding and fixing errors after the fact, quality management is the continuous monitoring and application of quality processes in all aspects of the project *[73]*. Agile development, shortly examined in section 2.2, best meets these all the needs.

Software Quality Management (SQM), as well as PQM, includes all processes that ensure that software products, services, and lifecycle process implementations meet organizational software quality objectives and achieve stakeholder satisfaction. SQM defines *processes, process owners, and requirements for the processes, measurements of the processes and their outputs, and feedback channels throughout the whole software lifecycle*.

SQM also comprises four subcategories: software quality planning (SQP), software quality assurance (SQA), software quality control (SQC), and software process improvement (SPI). SQP includes determining which quality standards are to be used, defining specific quality goals, and estimating the effort and schedule of software quality activities. In some cases, SQP also includes defining the software quality processes to be used. Software quality assurance activities define and assess the adequacy of software processes to provide evidence that establishes confidence that the software processes are appropriate for and produce software products of suitable quality for their intended purposes.

SQA is a process to provide confirmation based on evidence to ensure to the donor, beneficiaries, organization management and other stakeholders that product meet needs and expectations from **International Organization for Standardization (ISO) of quality standards** and other requirements. SQA means performing auditing the quality requirements, ensuring appropriate quality standards and operational definitions which are used. It assures the existence and effectiveness of processes and procedures tools, and safeguards are in place to make sure that the expected levels of quality will be reached to produce quality outputs. Assurance is the activity of providing evidence to create confidence among all stakeholders that the quality-related activities are being performed effectively; and that all planned actions are being made to provide adequate confidence that a product or service will satisfy the stated requirements for quality. SQA occurs during the implementation phase of the project and includes the evaluation of the overall performance of the project on a regular basis to provide confidence that the project will satisfy the quality standards defined by the project. One of the purposes of quality management is to find errors and defects as early in the project as possible. Therefore, a good quality management process will end up taking more effort hours and costs up-front. The goal is to reduce the chances that products or services will be of poor quality after the project has been completed. *More succinctly – SQA make audit quality requirements and measurement results to assure requirements are met; use tools identified in the plan; evaluate the results.*

SQC activities examine specific project artifacts (documents and executables) to determine whether they comply with standards established for the project (including requirements, constraints, designs, contracts, and plans). The goal of quality control is to improve the quality and involves monitoring the project outputs to determine if they meet the quality standards or definitions based on the project stakeholder's expectations. Quality control also includes how the project performs in its efforts to manage scope, budget and schedule. *More succinctly, quality control defines corrective action; take corrective action; revise project management plan.* SQC evaluates intermediate products as well as the final products. The fourth SQM category dealing with improvement has various names within the software industry, including SPI, software quality improvement, and software corrective and preventive action. The activities in this category seek to improve process effectiveness, efficiency, and other characteristics with the ultimate goal of improving software quality. Although SPI could be included in any of the first three categories, an increasing number of organizations organize SPI into a separate category that may span across many projects *(the Software Engineering Process Knowledge Area).*

Risk management can also play an important role in delivering quality software. Incorporating disciplined risk analysis and management techniques into the software lifecycle processes can help improve product quality (see the Software Engineering Management KA for related material on risk management).

Quality improvement means realizing recommended changes. It's the systematic approach to the processes of work that looks to remove waste, loss, rework, frustration, etc., in order to make the processes of work more effective, efficient, and appropriate. Quality improvement refers to the application of methods and tools to close the gap between current and expected levels of quality by under-standing and addressing system deficiencies and strengths to improve, or in some cases, re-design project processes. A variety of quality improvement approaches exists, ranging from individual performance improvement to redesign of entire project processes.

In conclusion, *quality management is a continuous process that starts and ends with the project. It is part of every project management processes from the moment the project initiates to the final steps in the project closure phase. ISO 9001:2015 is one of the top-quality certifications that an organization can attain.* Project Quality Management is a part of this standard.

It has also changed responsibility of IPs quality. *In past times, the quality department was responsible, but no more. Quality departments have been significantly reduced and functions have been transferred to the performing level or eliminated altogether. Nowadays, everyone is responsible for quality. Organizational management is responsible for the quality system. Project managers are ultimately responsible for project and product quality. Project teams are responsible for the quality aspects of their part of the project, and individual team members are responsible for quality in everything they do to contribute to project completion. No one has the luxury of off-loading quality responsibility to someone else or some other function. Everyone associated with a project is responsible in some way, with the project manager bearing the burden or obligation of ensuring quality in everything the project does [34].*

2.1.3. Quality management system according to ISO 9001:2015

Various editions of the ISO standards include dates in the reference number. The ISO 9001 standard may be listed as ISO 9001:2015 to indicate the 2015 edition. This fifth edition cancels and replaces the fourth edition (ISO 9001:2008), which has been technically revised, through the adoption of a revised clause sequence and the adaptation of the revised quality management principles and of new concepts. Soon the 6th edition was expected.

A quality product cannot be created if the quality system is not implemented in the enterprise (division). Very briefly – the quality system (*Figure 2.5, Figure 2.6)* is:

(1) A package of developed and approved regulatory and methodological documents *(instructions, methodologies, directives, etc.)* for each selected technological process of designing, manufacturing and distributing a product;

(2) Monitoring the gradual implementation of technological processes in accordance with the approved plan.

Achieving the desired quality is to monitor processes and methods to ensure quality. To ensure a desired quality level, a system is required. This system is usually called the Quality Management System (QMS), based on seven principles: *customer focus; leadership; engagement of people; process approach; continuous improvement; evidence-based decision making; relationship management*. The process approach, known as PDCA cycle, seems to be a little bureaucratic, and in many situations it's very bureaucratic, with a lot of requirements.

The ISO 9001:2015 approach is prevention based, an approach proven to be more effective in the long run than identifying and fixing accepted defects as they occur. Finally, it is a framework for quality improvement. Continual improvement, not satisfaction with the status quo, is an essential part of the ISO 9001 approach. But the scale of the QMS should be consistent with the quality assurance objectives.

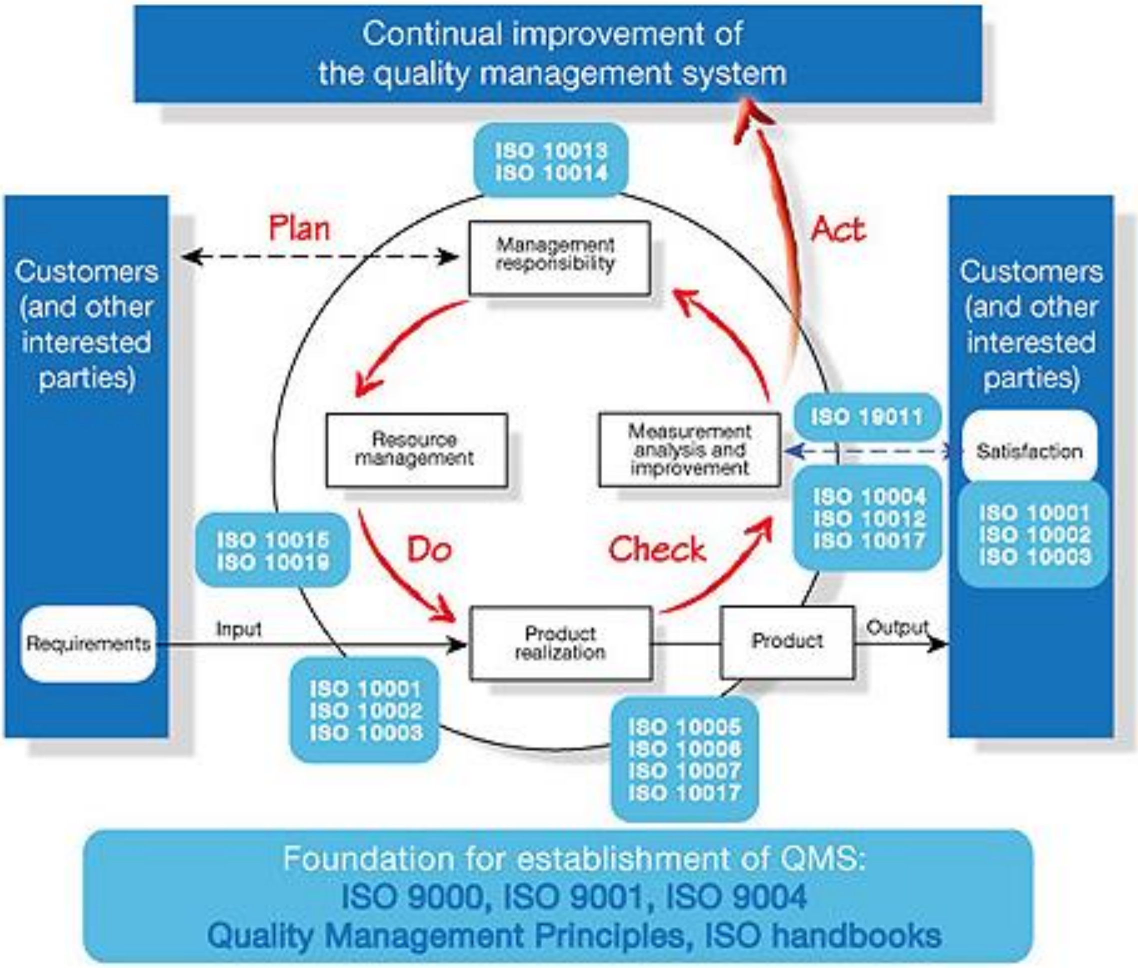

Figure 2.5. ISO 9000 Process model and standards requirements

Source: *Taken from [58] (Page 33)*

Figure 2.6. Structure of ISO 9001:2015 in the PDCA cycle

Source: *Adapted by the author based on [74] (Page 9) (emphasize in the circles numbers, which refers to the clauses in ISO 9001:2015 standard)*

A general model of QMS for a modern software development company, in according with ISO 9001 standard, consisting of three parts, as is sown in *Figure 2.7*.

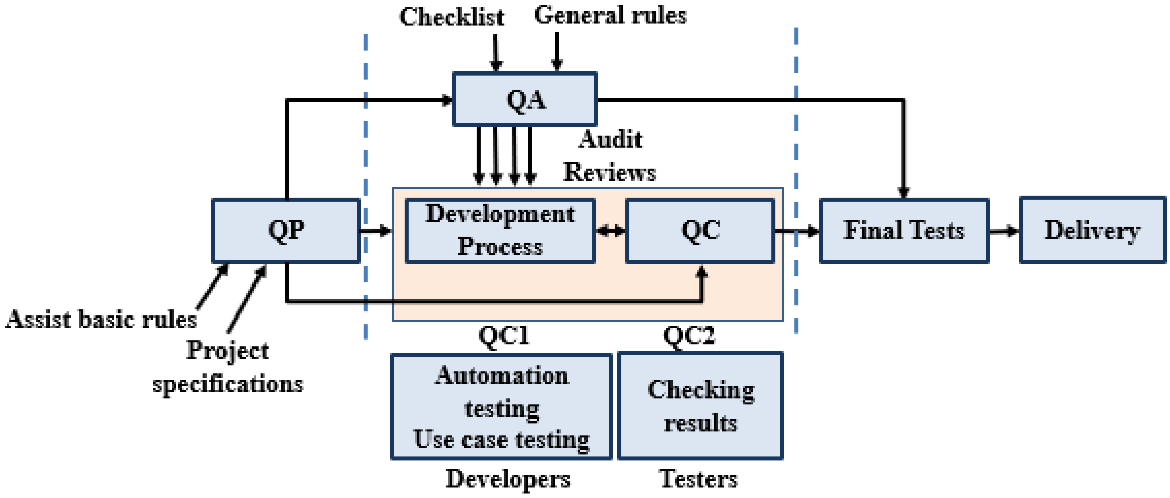

Figure 2.7. QMS model for a modern software development company

Source: *Taken from [75](Web-page)*

Legend: QA = Quality Assurance, QP = Quality Plan, QC= Quality Control

In according to ISO 9001, QMS must be „born”, „grow” and „mature” inside the company.

ISO 9001 is a brief document. It contains many prescriptive paragraphs that indicate what an organization „shall” do. QMS reflects a set of activities that need to be undertaken in order for a product to be of high quality.

In general, verification can be viewed as *the assessment process using decisional algorithms, based on specific product observations*. Often, verification is confused with testing. Testing is much more; it seems to be a craft. It is described as *exploration, discovery, investigation and learning*. In opposition it should be said that almost anyone can follow a list of punctual checks. The requirements can be analyzed, check lists can be constructed based on them, and verification for each control can be performed. And the quality *assurance of the project is not a quality assurance control, but the management of its production process*.

QMS for a software development organization covers: (a) the actual engineering activities (analysis, design, design, coding, (b) revisions applied to each step of the project, c) testing strategies, including automated methods and tools d) control of the software documentation and its maintenance, (e) compatibility with standards, if these are applicable, (f) measurement and reporting mechanisms (e.g. internal quality).QMS example: Quality management oriented to defining and standardizing processes, procedures, templates, and more. Quality assessment in this approach uses maturity models, such CMM *[76]*, CMMI *[77]*, Agile *[23]*.

2.1.4. CMMI Methodology for PQM and software developments

Capability Maturity Model Integration *(CMMI, replace the old model CMM)* describes a set of features related to the organization's ability to follow repeatable processes in conducting its activities. It is a set of models (methodologies) for improving processes in organizations of different sizes and activities. CMMI contains a set of recommendations in the form of practices, the implementation of which, according to the developers of the model, allows realizing the goals necessary for the full implementation of certain areas of activity.

The CMMI set (*CMMI Institute Resource Center*, https://cmmiinstitute.com) includes three models: CMMI for Development (CMMI-DEV), CMMI for Services (CMMI-SVC) and CMMI for Acquisition (CMMI-ACQ).

The most famous is the CMMI-Dev model, which focuses on organizations involved in the development of software, hardware, and integrated systems. *Currently, there are two versions of the models: three separate models of version 1.3 (released in November 2010) and version 2.1 (released at the end of March 2018). „Capability area is a group of related practice areas that* *can provide improved performance in the skills and activities of an organization or project" [77].* Between the benefits of the new CMMI V2.0 we can mention:

- Support implementation →Collaborate with stakeholders for informed decisions, support product integrity and deliver committed outcome
- Ensure quality → Individuals, team and organization to define and maintain superior standard of deliverables for customer and business

At the bottom of the CMMI scale are organizations that do not carry out repeatable processes, where the activity takes place ad hoc and chaotic. At the top of the scale are companies that use defined and repeatable processes, collect performance indicators to support continuous process improvement, and continually identify creative methods for doing business. *Figure 2.8* shows the old CMM levels, and *Figure 2.9* shows the new structure of the model and the Practice areas for each maturity level.

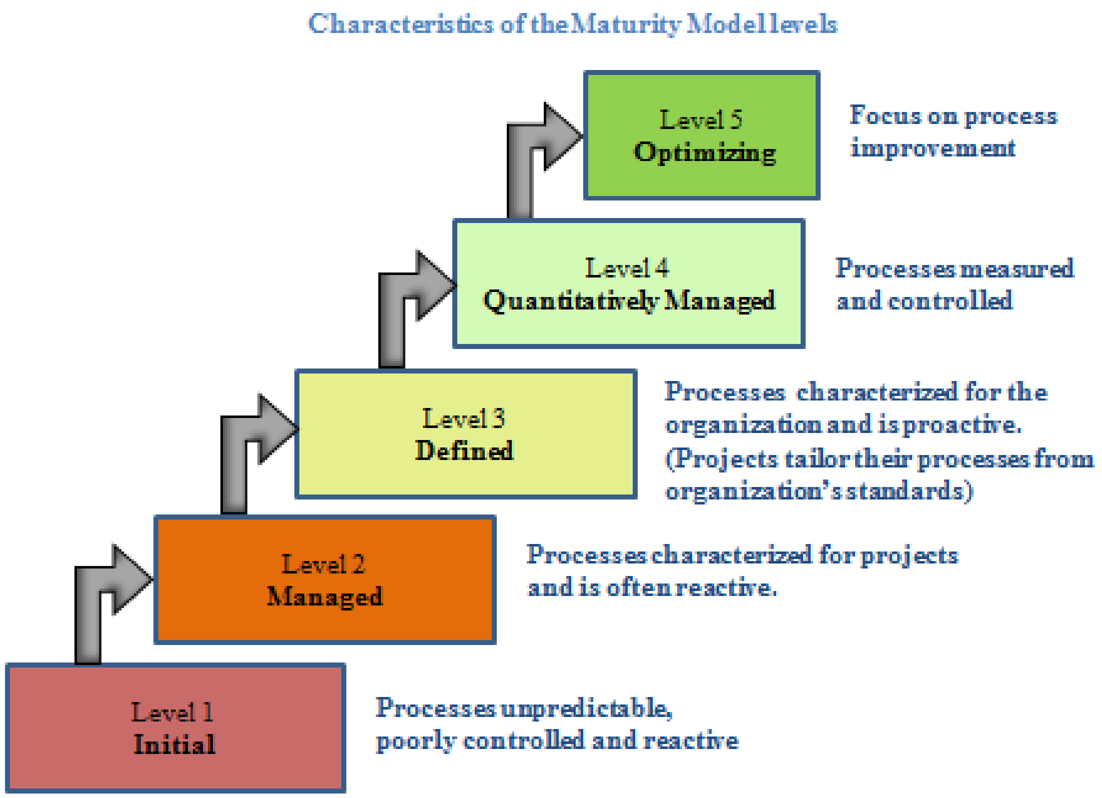

Figure 2.8. Characteristics of the Maturity Model levels

Source: *Taken from [78](Web-page)*

Maturity levels represent a staged path for an organization's performance and process improvement efforts based on predefined sets of practice areas. Within each maturity level, the predefined set of practice areas also provides a path to performance improvement. Each maturity level builds on the previous maturity levels by adding new functionality or rigor.

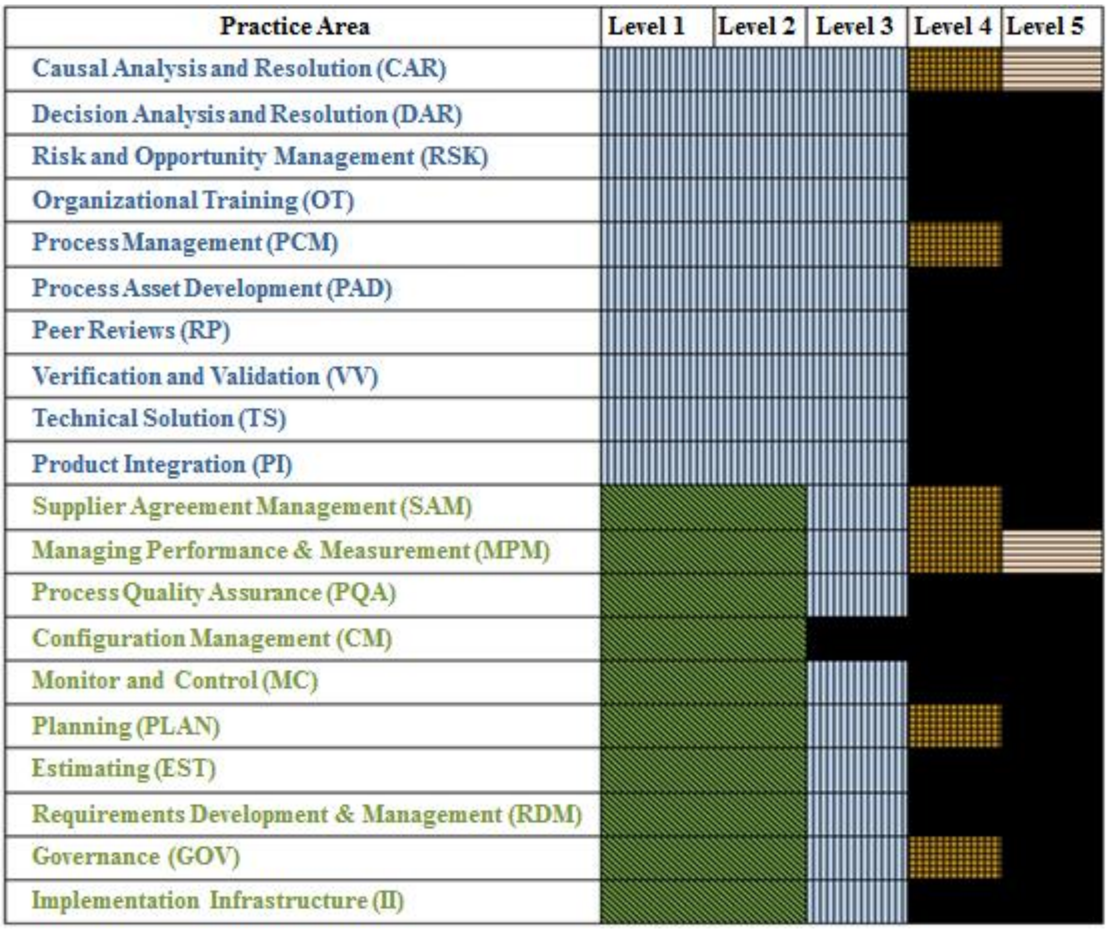

Figure 2.9. CMMI V2.0 Practice Areas - Development View

Source: *Taken from [79](Web-page)*

Maturity Level 1, Initial *(chaos, crises, ad hoc and unknown)*, unpredictable and reactive**.** Work gets completed but is often delayed and over budget. The organization has a few common processes (are specialized and non-organized), the success of projects depending exclusively on the qualities and abilities of the people. The organization does not provide a common environment for increasing the success of the projects. Today most organizations are at this level (they do not have a culture of PM).

2. Maturity Level 2, Managed *(Managed on the project level)***.** Projects are planned, performed, measured, and controlled. The trials are repeated on a regular basis. An attempt is made to establish a reference for future improvements to be made. Most of the companies that start on the CMMI road are trying to reach this level.

3. Maturity Level 3: Defined ***(****Standard repeatable, descriptive processes by standards).* Proactive, rather than reactive. Organization-wide standards provide guidance across projects, programs, and portfolios. There have been made attempts to achieve the same level of process standardization, similar to Level 2 project management processes. Repeatable processes, deliverables, tools, etc. are included here.

4. Maturity Level 4: Quantitatively Managed (Manageable, measurable, controlled reaction). There have been collected indicators for project and development management processes. This information is structured in the form of knowledge warehouses on past projects in order to be used in future projects. Organization is data-driven with quantitative performance improvement objectives that are predictable and align to meet the needs of internal and external stakeholders.

5. Maturity Level 5: Optimizing (Stable and flexible, continue optimization, enhancement)**.** Organization is focused on continuous improvement and is built to pivot and respond to opportunity and change. The organization's stability provides a platform for agility and innovation. Characteristic of this level is the existence of closed loop execution of processes - measurement and continuous improvement. Measurement results are used to continuously identify creative processes for improving processes, which is a permanent concern.

2.1.5. Other alternative approaches for addressing project management quality

Today, quality is widely regarded as a necessary requirement for a company's success and competitiveness in the market. According to Turner *[80]*, definitions of good quality on a project often concern the output and how well and to what extent it: meets the specification, is fit for purpose, meets the customer's requirements or satisfies the customer.

The perceived quality should at least match the expectations. The stakeholders' expectations, especially the end users ,determine whether the project is regarded as successful or not. It is important to manage the stakeholders and make sure that they have an accurate image of the project and what is to be delivered. *Research has shown that if a project manager, the project team and other stakeholders agree before they start how they are going to judge the project's success, then they maximize their chance of success.* By the end of the project, what the customers think they want, what they actually need and what you think they need should be the same thing. With that said, Turner states *[81]* that the widely accepted definition of good quality is taken as delivering project objectives that are fit for purpose, i.e. they achieve the desired result.

A more methodological approach which represents a more professional-oriented or business-oriented approach was formulated by Tiwana and Keil *[9]*. They found that in many cases the difficulty to assure quality was a result of intricacy and complexity of the project along its progress which was often dealt with *by using informal practices and intuitions* („educated guesses"). These were due to conventional management concept, which saw the project as an outcome-yielding process rather than an integrated knowledge solution involving technical skills, management experience and business understanding of customer needs. According to Tiwana and

Keil this view sees risk and quality as inherent in the process of knowledge creation, design and incandescent into a coherent solution. Tiwana and Keil *[9]*. and Tiwana and Bush *[82]* main contribution is by offsetting the focus from organizational and managerial orientation for providing quality of IS projects to a more knowledge orientated approach. Their rationale suggests that since knowledge constitute the bulk of the added value created by information systems then knowledge creation and knowledge management should be applied for quality and quality assurance.

In view of the dynamic characteristics of the software industry and development processes, this approach led the way for the development of a more relaxed and flexible approaches for project managers. For example, the *[83]* qualitative review show that while "conservative" indicators for projects' success often relate to the most limited aspects of time, budget and system requirements, a broader perspective must be embraced, especially one that will combine and evaluate different stakeholders' perspective regarding quality. This perspective should lead to the development of different tools and metrics for measuring and assessing quality. For the most part, this approach is illustrated in *[10]*, which present a multi-dimensional relating not only to the different stages of the project but also to different stakeholders of the project.

Perception of quality acknowledge not only the existence of risks along a project's lifecycle but also that these risks must be prioritized differently along the project's lifecycle according to the exposure of the project's aspects to these risks (e.g. the potential damage that quality prevails. This can lead to a multi-processes approach as well as a multi-system approach (*Figure 2.10.*).

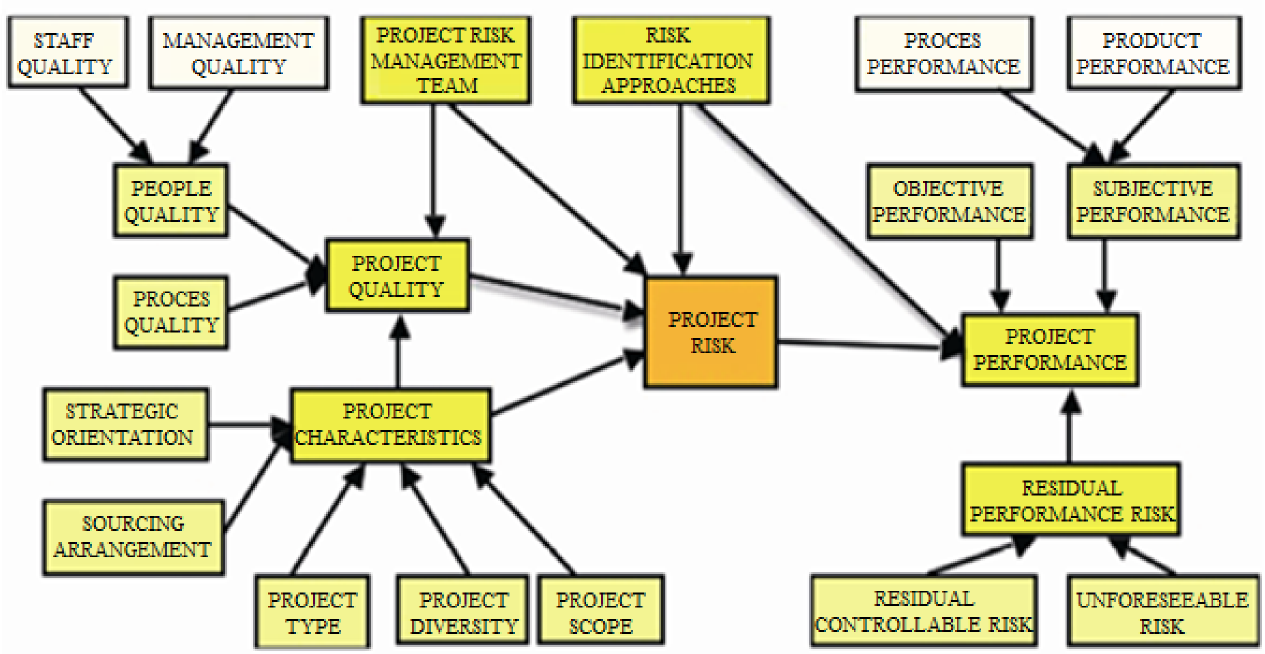

Figure 2.10. Common conceptual framework for PM and QM

Source: *Taken from [84](Page 298)*

This approach combines quality aspects associated with the project itself and the given reality as it is reflected by management perception as well as the project's objective characteristics such as the quality of human resources and management quality.

The Information Technology Infrastructure Library (ITIL) *philosophy suggests that IT service management should be viewed as a service providing business.* This reflects both on the mission statement of the service and on the way in which quality conforms into an interactive term between the service provider and the customer, ensuring the customers get what they want. These also mean that quality improvements, using the PDCA circle in an ongoing process in which customer services quality in continuously examined. ITIL and its constituent modules were scoped and developed within an overall framework *(Figure 2.11)*.

ITIL gained wide popularity since its establishment in 1989 and is now considered to be the industry standard used by many governments. Known frameworks, such as Microsoft Operations Framework (MOF) or HP's ITSM Reference Model are based on ITIL (for instance, *[51]*) along with other known and useful frameworks such as COBIT (Control Objectives for Information and Related Technologies), CMM (Capability Maturity Model), Six Sigma and Balanced Scorecard *[85]* and other international industry standards as BS15000, ISO20000 *[86] etc*.

Figure 2.11. The ITIL Framework

Source: *Taken from [87](Cover page, page 10)*

Cob IT (Control objectives for Information and related Technology, *Figure 2.12,* one of the most known frameworks was created by the ISACA as a management tool for bridging the gap between control requirements, technical issues and business risks.)

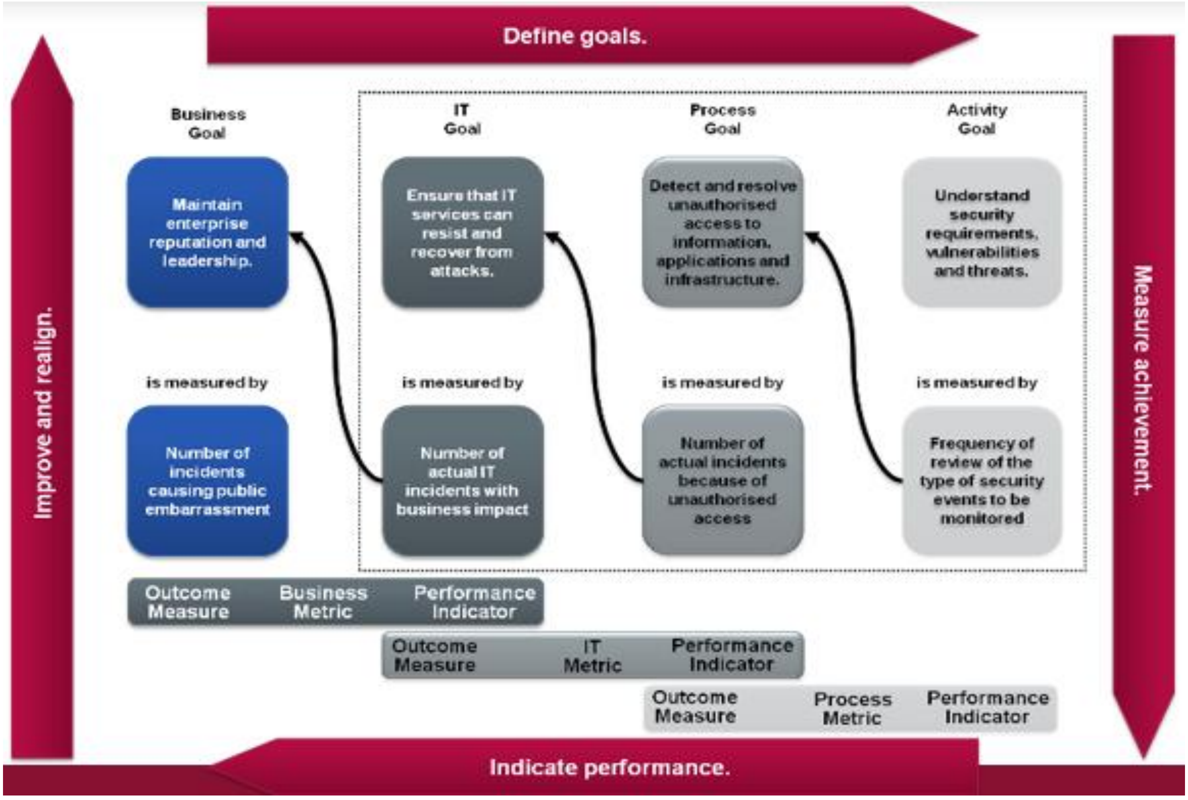

Figure 2.12. COBIT processes, goals and metrics

Source: Taken from [90] (Presentation, page 12)

COBIT offers a set of tools, which is based on standards such as ITIL, ISO 27001, ISO 27002, and PMBOK, is often used at the highest level of IT governance improving these standards alignment with business needs with relation to IT-related activities *[88]*, *[4]*. It provides better governance using a continuous improvement process which combine defining business goals, measure achievement, indicate metrics and improve and realign. One of COBIT's key elements is decomposition of the processes into activities and which are monitored by different functions in the organization. This is usually achieved by using a **RACI** chart, identifying is **R**esponsible, **A**ccountable, **C**onsulted and/or **I**nformed with regard to each activity.

Other standards and templates. Many of the standards reviewed above evolved within the other frameworks. These can be categorized into three main categories:

- *Institutional standards* – these standards can be viewed as an evolution or an adjustment of the „common”" standards so that projects' quality assurance demands will be accommodated within a specific business or operational framework. Among these standards are US OCIO policy no.*132 (Providing Quality Assurance for Information Technology Projects)*, the CDC project quality management plan template, IDA PMQP *(Project Management and Quality Plan Checklist),* OWASP *(Open Web Application Security Project)* and many others.

- *Private templates* – these templates are internal frameworks for planning and managing quality of an IS PM. These templates often rely on public standards, often simplifying them and adjusting them for specific uses or fields of expertise *[89]*.
- *Private tools:* independent private initiatives developments of PMO software that incorporate industry standards in a framework aimed at providing specific end-users with IS PM quality assurance tools. Among these tools are MS Project, „Super Project" (a private initiative of Semantics for utilizing PM both within and between enterprises), Redmine - an open source flexible tool (web application), Jira Software etc.

I'll Know It When I See It (IKIWISY)
Uncertainty is inherent an inevitable in software development processes and products.
Hadar Ziv (University of California)

2.2. From Quality Assurance to Agile Quality

The technological boom in the early 1990s has led to many new approaches to managing software development projects. In addition to the traditional „Waterfall/Cascade" step-by-step development method, new methods have emerged, such as Prince 2, RAD, V-Model, etc. But these have not responded to the ever-increasing need for quality software assurance and the rapid pace of technological advances, many projects being abandoned, canceled or terminated with failure because the software that was developed was often outdated***. One of the most possible solutions for project success today is considered Agile.*** This is because Agile development gives a possibility to see and know growing product *iteratively, incrementally and evolutionary,* in short sprints, according ***IKIWISY***.

Agile is a pretty hot topic, and most developers get pretty excited about giving it a try. Agile holds the promise of creating teams of empowered individuals; teams full of people working on the highest priorities of the business with a shared sense of purpose. When Agile is done well, it creates really fun places to work, there is nothing quite like being part of a team of people working hard toward shared goals. Today there are numerous publications, surveys, statistics that eloquently demonstrate the growing and overwhelming application of Agile: *20 Surprising Project Management Statistics (2016, [91])*, then 15 *Incredible Agile Project Management Statistics for 2018*, both published by Rachel Burger in Project Management *(2017, [92])*, *How Agile are companies in Luxembourg (2017 [93])*, etc.

Thousands of professionals share experience with Agile in the largest, longest-running Agile survey and reasons for adopting Agile, annual executed by Version One (https://www.collab.net/). The 13th annual report features new themes, shifts in priority and expanded areas of focus12 key reasons companies are adopting Agile *[94]*.

The 13th annual report Version One, among many reasons for adopting Agile, the reasons stated for adopting Agile were less about increasing productivity *(51% compared to 55% last year)*, and more about improving team morale *(34% compared to 28% last year)* and less about reducing project risk *(28% compared to 37% last year)*, and more about reducing project costs *(41% compared to 24% last year)*.But the trend of miraculous growth is kept constant.

A recent study by Agile Journal (https://Agile-journal.com) shows that *over 88% of evaluated software companies (many with over 10, 000 employees) already use or plan to use Agile methods and practices for projects on which implements them.*

2.2.1. The reasons for adopting Agile philosophy

According to many sources and surveys, Agile projects as a whole are 28% more successful than traditional projects *(Table 2.1, Figure 2.13)*. For example, this fact is confirmed by State of Agile Report (Survey 2019) *[94]*, Chaos report and other statistics *[16]*.

Table 2.1

Agile versus Waterfall resolution projects

Size	Method and Size	Successful	Challenged	Failed
Small Size Projects	Waterfall small size	44%	45%	11%
	Agile small size	58%	38%	4%
Medium Size Projects	Waterfall medium size	7%	68%	25%
	Agile medium size	27%	62%	11%
Large Size Projects	Waterfall large size	3%	55%	42%
	Agile large size	18%	59%	23%
All Size projects	Waterfall all size	11%	60%	29%
	Agile all size	39%	52%	9%

Source: *Adapted by author based on [16] (Page 7) (the selected data sort order)*

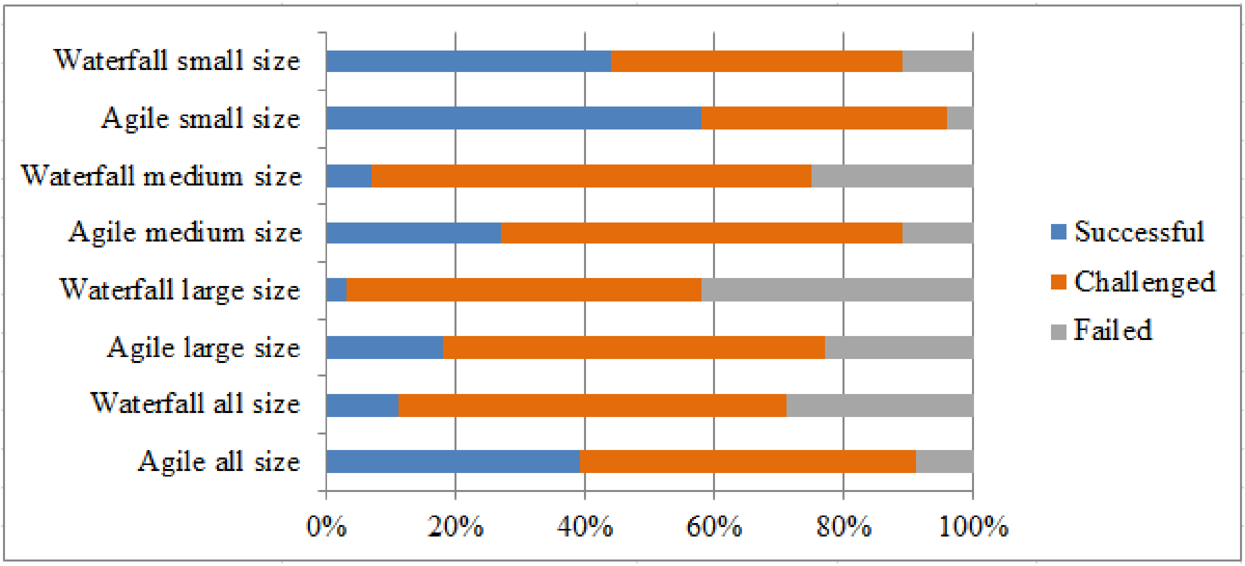

Figure 2.13. Agile versus Waterfall resolution projects

Source: *Developed by author based on [16] (Page 7)*

The average share of successful projects in Agile is more than 3.5 times higher than in Cascade/Waterfall (39% vs. 11%), large projects – 6 times (18% vs. 3%), medium-sized projects 3.9 times, small projects – 1.3 times. And the share of abandoned projects is lower for all project categories: in the medium – 3.9 times; large – 1.3 times; average – 2.3 times; small – 3.8 times. So, figures denote a superior net efficiency of Agile vs. traditional methodologies for IPs development.

Cost reduction has gained importance. Last year (2018) saw a 71% increase in those selecting "Reduce Project Cost" as a reason for adopting Agile. There was also a 27% increase in "Project Cost Reduction" as a reported benefit of implementing Agile.

Benefits of Adopting Agile. World continue to see many benefits realized by companies adopting Agile, some of the ones shown below.

Faster time to market. Lots of folks that decide to go Agile are pretty fed up with 18-month delivery cycles that quite often deliver the wrong products to market. One's that our customers just aren't interested in buying. The idea of two-week delivery cycles and quarterly release cadences is pretty appealing.

Better quality. Developers are generally tired of building crap and our customers are universally tired of getting crap. When businesses fix time, cost, and scope – the only thing developers have left to manage is quality. Agile fixes time, cost, quality and gives us the tools to vary the business and technical scope of the solution. You might not get everything you hoped for, but you can trust what was delivered.

Early ROI. Even though you may have thought it was less efficient to splitting stories, it makes a real difference to the business. We can show the output of this sprint to an external customer and sell business based on this.

Efficiency, or „reducing waste", Agile holds the promise of helping us eliminate the stuff we don't need and get down to the business of building working software**.**

Customer satisfaction. Building products our customers can use makes them happy. Being able to frequent add new features based on their feedback makes them happy too. Agile helps us build of partnership with our customers, one where we are working together to grow customer satisfaction.

Emergent outcomes. Some folks don't know what they want to build or how to build it. Some people are building products for markets that don't exist yet using technologies that are brand new and cutting edge. Agile is a great way of building software when you have to explicitly account for the fact that you'll have to learn as you go. Build a little product, learn something from your customer, adapt your vision, build a little more software, and ultimately create something that is better than you could have ever planned in a vacuum.

Systematic feedback from real customers **allows early risk reduction.** By delivering early and getting feedback, we reduce the risk of building the wrong product. By focusing on architectural risk in the early sprints, we reduce the risk that we won't have a solution that can be built in time etc. (For more details see referred surveys).

2.2.2. Quality management in Agile vs traditional methods

Three styles of Agile is **Iterative, Incremental,** and **Evolutionary** allow to providing quality of developed product. Because the quality of software products can be improved through preventative processes or an iterative process of continual improvement, which requires management control, coordination, and feedback from many concurrent processes:

(1) The software lifecycle processes,

(2) The process of fault/defect detection, removal, and prevention, and

(3) The quality improvement process.

Building the quality through the prevention and early detection of defects, continual improvement, and stakeholder focus are the main tasks for Agile. These concepts are based on the work of experts in quality who have stated that the *quality of a product is directly linked to the quality of the process used to create it.*

Approaches such as the Deming improvement cycle PDCA, evolutionary delivery, quality function deployment (QFD) and others offer techniques to specify quality objectives and determine whether they are met are characteristics of Agile development.

Agile is a collection of 4 values, 12 principles and some frameworks (methods and practices) that can be applied on an software development projects *[95]*. The 4 values of Agile

1. **Individuals and interactions -** OVER processes and tools
2. **Working software -** OVER comprehensive documentation
3. **Customer collaboration -** OVER contract negotiation
4. **Responding to change -** OVER following a plan

The Agile model is more flexible than traditional methods, making it a better fit in a fast-changing environment *(Table 2.2, Figure 2.14)*.

Table 2.2

Waterfall vs Agile: inversed triple constraint

Waterfall	**Agile**
Fixed scope	Fixed time and costs (resources)
Provide the quality upon delivery	In time quality is increasing
The time and costs are increasing	The variable is the Scope (in short time)

Source: *Developed by author*

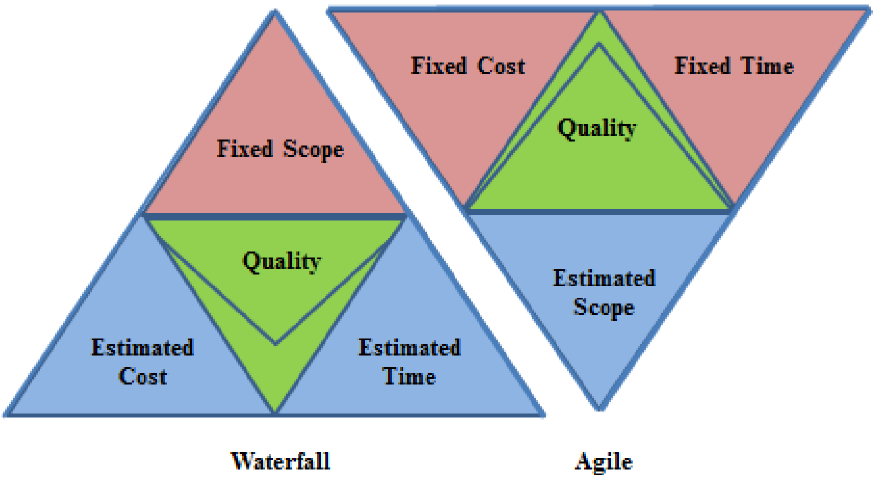

Figure 2.14. Waterfall vs Agile: inversed triple constraint

Source: *Adapted by the author based on [96] (Web-page) (emphasize the quality element in the triangle)*

As *Figure 2.15* suggests, the efficiency of traditional development (ca. 30/70) is inverse to efficiency in Agile (ca. 70/30), and is close to Pareto's magic numbers (formula 80/20). And confirm that *„for a new software system, the requirements will not be completely known until after the users have used it"* *(Watts Humphreys, IBM).*

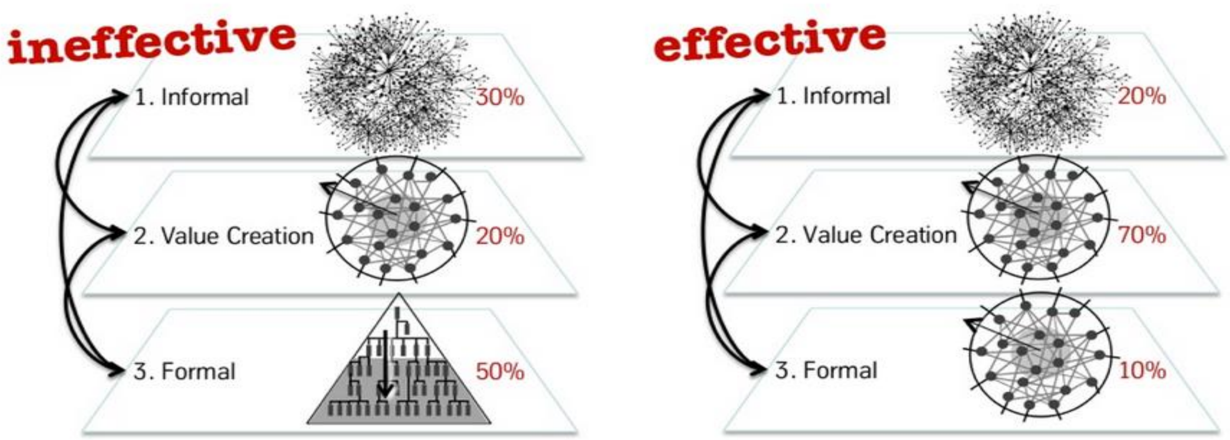

Figure 2.15. Inversed Value creation Waterfall vs Agile

Source: *Taken from [97](Presentation page 19)*

Development Agile cycle is fully in line (in according) with management standards, TQM and allows for continuous improvement of product *(Figure 2.16).*

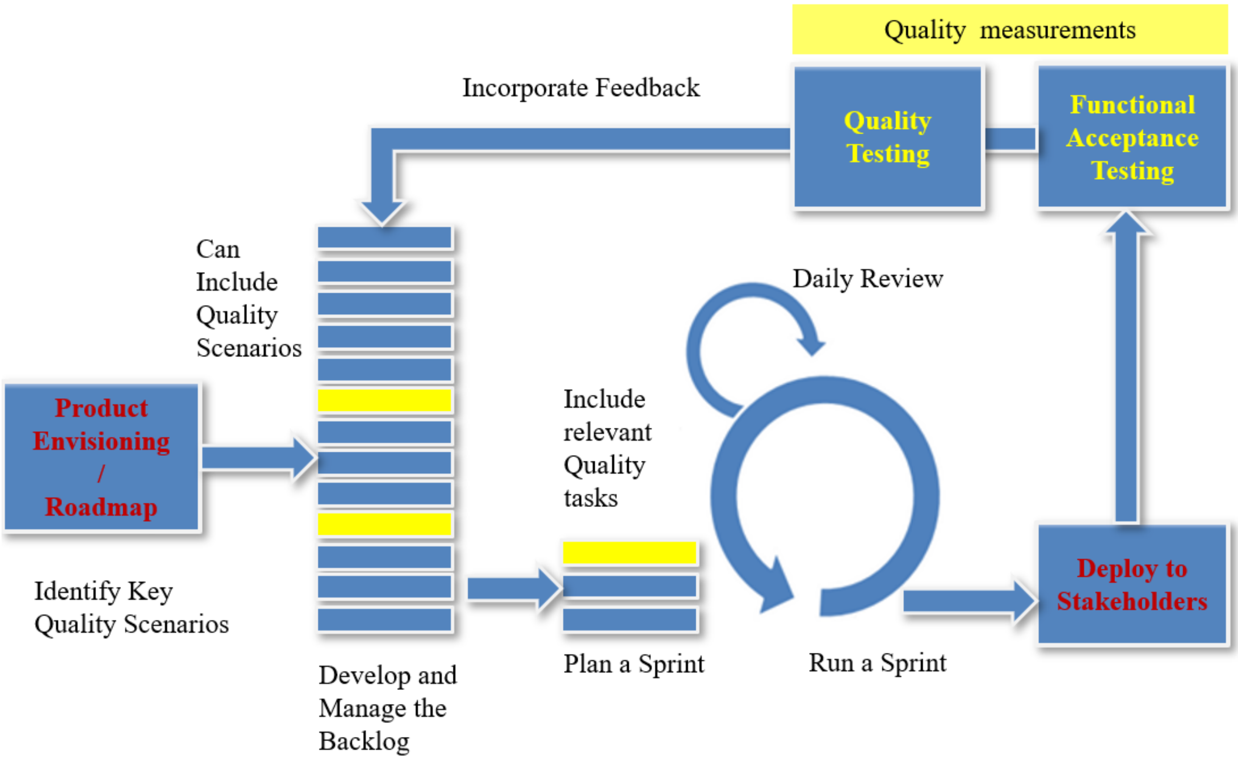

Figure 2.16. PDCA and Quality cycle in Agile (SCRUM)

Source: *Adapted by the author based on [98](Page 5)(emphasize quality measurements)*

Being Agile it is about the people and teams, about customer and delivering software, about continuous improvement, and constantly applies PDCA cycle:

- By the process (Retrospective analysis, Scrum, Kanban etc.);
- By the user feedback (Sprint review, demo Scrum);
- By the team itself (peer review, pair programming etc.).

Measurement is a key to process improvement. The needs for improvements can be investigated after performing measurements. In many cases this is impossible until the final delivery of the product. In Agile software processes development, it is possible along life cycle.

Agile fits perfectly with the requirements of quality management standards – continuous improvement and focus to customer: „*Our highest priority is to satisfy the customer* through early and continuous delivery of valuable software" *[95]*.

Joseph W. Yoder and all in *[98]* describes twenty-four patlets organized into four categories: (1) Knowing where quality concerns fit into your process, (2) Identifying system qualities, (3) Making quality visible, and (4) Being Agile at quality assurance. A patlet is a brief description of a pattern, usually one or two sentences. Additionally, this paper describes **six of these patlets, written as patterns**: *Integrate Quality, Agile Quality Scenarios, Quality Stories, Fold-Out Qualities, Whole Team and Quality Focused Sprint.* For details see cited source.

Many of the Agile principles related to quality have their roots in the philosophy of total quality management *(Figure 2.17)*

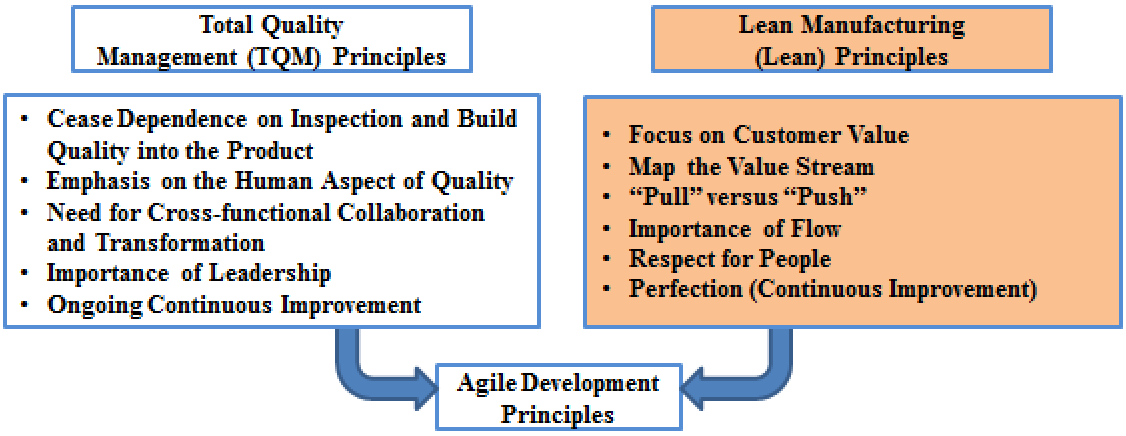

Figure 2.17. The roots of Agile practices regarding quality

Source: *Taken form [24](Page 168)*

2.2.3. Scaled Agile Framework for medium and large projects

In the previous chapters we mentioned that the main negative failure impact is from medium and large projects: near 87% of large and medium project in 2011-2015 failed or were challenged *(Table 1.2)*. We also stressed/emphasized the need to develop and/or choose the right approach for project management and quality improvement, especially for medium and large IPs, as more prone to risks of project failure.

Agile Methodology also faces these challenges, providing new methods such as Scrum of Scums and/or SAFe *(Figure 2.18)*: Scaled Agile Framework *(SAFe®)*, (https://www.scaledAgile.com) for Lean Enterprises, as a knowledge base of proven, integrated principles, practices, and competencies for Lean, Agile, and DevOps, *Large Scale Scrum (LeSS,* https://less.works/less/framework/index.html), DAD, https://disciplinedagileconsortium.org/Disciplined-Agile-DAD).

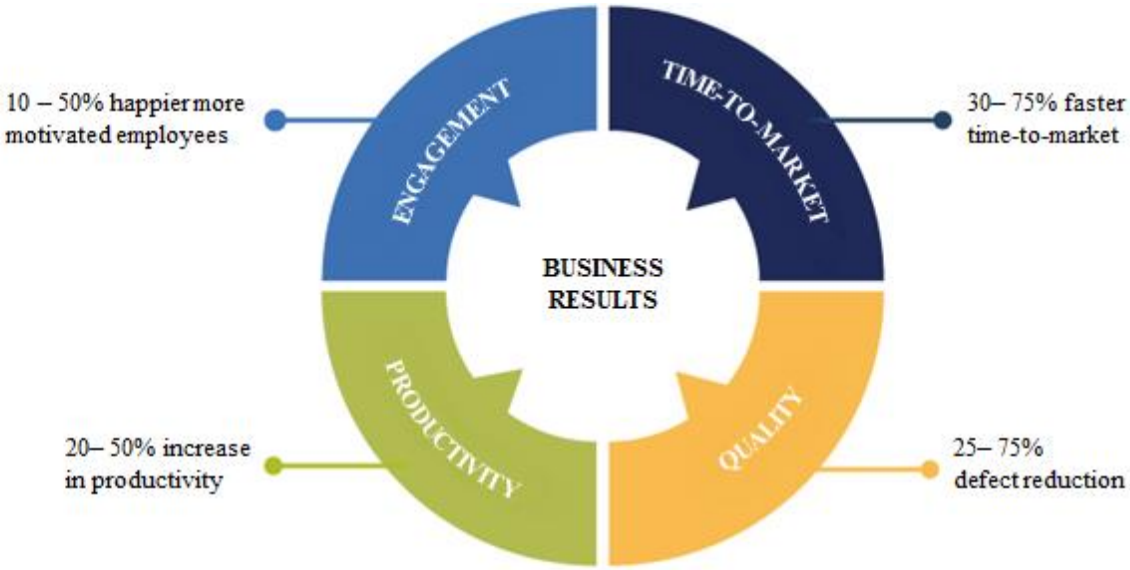

Figure 2.18. Business results delivered from SAFe

Source: *Taken form [99](Web-page)*

The practices of SAFe are freely available of the framework website (https://www.scaledAgileframework.com/). These have been proven to work in an integrated fashion in many organizations. SAFe transformation requires developing *new „enterprise competencies" that enable a new style of leadership, new ways of thinking and working, and a culture focused on value delivery and continuous improvement.*

SAFe, as well as other Agile methods, such LeSS, DAD etc., offer **Built-in Quality**:

- Ensures that every increment of the solution reflects quality standards;
- Enables high velocity and a sustainable development pace;
- Software practices include continuous integration, test-first, refactoring, pair-work, collective ownership, and more;
- Hardware practices include exploratory iterations, frequent system integration, design verification, model-based systems engineering, and set-based design.

SAFe permit fast feature delivery with architectural runway, relentlessly improve without inspect, without the system demo etc.

The latest version, SAFe 4.6, 2018 introduces the Five Core Competencies of the Lean Enterprise that are critical to achieving and sustaining a competitive advantage in an increasingly digital age, for details see https://www.scaledAgileframework.com/about/, https://europe2018.safesummit.com/presentations/, https://www.scaledAgileframework.com/ advanced-topics/ etc.

The Framework is supported by the SAFe Community Platform, Customer Success Team, and an extensive network of 250+ Scaled Agile Partners providing platform, training, consulting, and implementation services in almost every region of the world. And while every business situation is unique, we have found that the SAFe Implementation Roadmap always delivers results.

These are some pretty impressive results and may sound like something that's tough to believe or marketing hyperbole. But the results of Agile implementation really are amazing, especially in quality assurance.

Conclusions on chapter II

The whole ideology of the PQM is built around the recommendations of PMI, TQM and ISO standards. The project-specific methodology is fully replicated in the main international standards for project and quality management. The main concepts of project quality in PMBOK and in the ISO mentioned standards are similar. PMBOK Guide 6th edition in quality management issues has incorporated the best and most advanced of ISO standards. Also, we can consider the quality management of the project as part of a corporate-wide QMS, which, in turn, is based on the standards of ISO 9000:2015 and ISO 10006:2017.

Lifecycle models such as the Waterfall, Agile, etc. prescribe a detailed process for developing software. **Process maturity models** such as CMMI and SPICE provide approaches for assessing and improving the processes of organizations.

Building a QMS in accordance with the ISO 9001 is an issue, which can be solved by development of many documentations and bureaucratic processes. But customers don't want better project management. Customers want better product delivery. *All the Agile tools and techniques exist solely to that end.* Obviously, each project is best suited to a methodology that takes into account *the level of maturity of the organization, quality policy, types of projects, requirements, criteria, metrics and quality procedures.* This means that Quality Management Systems are not the same, even for enterprises of the same type, and **each QMS „grows" into an organization, and it can be „raised" only by the employees themselves,** possibly with third-party support.

Lot of effort is invested in the process quality improvement. When a project is undertaken, the aim is to deliver the right product at the right time with the right functionalities. It is a common scenario that the one at the receiving end always desires/expects the best to be delivered to them. The developers and testers must to ensure that they are able to meet the expectations of their clients.

The **other major topics for improvement of IPs** quality are the *comprehensive definition/establishing baselines for quality, using a right development methodology, involving personal etc*. Today one of the most recommended, globally recognized and aggressively implemented in various fields is Agile philosophy, which treats software quality in a more efficient and effective way. *Different views of Software quality, Quality factors, Quality criteria, Quality models, measurement of user's view and of manufacturing view etc. are examined in ISO 25010, 25012, 5022, ISO 25023 etc*.

Project management standards have greatly evolved during the past 10 years. These standards provide project managers as well as other executives with better tools for managing projects and achieving business objectives, as well as better understanding of business processes, while improving quality through Quality assurance methodologies.

This suggests that either that **PM and QA methodologies are limited with regard to IS PM** or that information projects' practices are handled improperly, both resulting in PM inability to handle projects' risks that lead to high failure rates. As stated above, *this pose a major concern for investors, organizations and practitioners which face high risks and unnecessary costs during IS projects and to minimize the project risks through Quality Assurance*. Some of these issues have been addressed in this section; others are addressed in the following sections.

Give me a point of support and I'll move the Earth out of the way!

Archimedes

III. SOFTWARE QUALITY MODELS AND TOOLS

In chapter II was examined processes and resource-oriented project quality models (e.g. ISO 10006, ISO 9001, TQM, CMMI). These standards and recommendations specify requirements, say what do you need to do, but they do not say how to do it. In literature a large number of software quality models have been proposed for this purpose. In chapter III we focus on product/software quality models, not on the processes that lead to its construction, even though they are closely related. Also, there will be examined the issues of models, assumptions and possible solutions, including one of the most current software quality models – ISO 25010:2011.

3.1. Overview of system/software quality models

One of the main aspects of the software is the quality. To predict and to develop software with a high-level quality at lower cost it is necessary to use a quality model. The evolution of the software quality models starting with the massive development of software systems. There are various models known in the literature, to provide software quality. *Figure 3.1* shows the general evolution of quality models from the McCall model in 1977 until 2013, some of which are listed in *Table 3.1* show the most famous quality models.

The first McCall model appeared in 1977. The complexity of the development and maintenance of certain types of software with special requirements for their quality justifies the importance of developing other models that use other characteristics, such Boehm, FURPS, FURPS+, Dromey. Since 2000 the construction of software started to depend on generated or manufactured components and gave rise to new challenges for assessing quality. These components introduce new concepts such as *configurability, reusability, availability, better quality* and *lower cost*. Consequently, the models are classified in **basic models**, and those based on components called **tailored** or **private** or **authors quality models** *[100] (pp. 126-130),* enriched with other characteristics, specific to certain areas, e.g. onboard software, navigation systems. Simultaneously with the development of the Internet and Web as open systems, in 2003 a new subclass of tailored models - **open source quality models** started, e.g. Cap Gemini, Open BPR etc.

Table 3.1

The most famous quality models

№	Model and source	Year of publication	Number of model levels	Number of subcharacteristics	Authorship
1	McCall *[101]*	1977	2	11/35	McCall J. A.
2	Boehm *[102]*	1978	3	3/8/18	Boehm B. W. et all
3	Ghezzi *[103]*	1991	1	8	Carlo Ghezzi
4	FURPS *[104]*	1992	2	5/25	Grady R. B.
5	IEEE *[105]*	1993	2	6/19	IEEE
6	Dromey *[106]*	1995	2	4/13	Dromey, G. R
7	SATC *[107]*	1996	2	4/13	NASA, SATC
8	ISO 9126 (1-4) *[108]*	2001-2003	3	6/19	ISO
9	QMOOD *[109]*	2002	2	6/19	Bansiya, J.
10	ISO 25010 *[33]*	2011	2	8/31	ISO

Source: *Developed by the author in based on literature analysis*

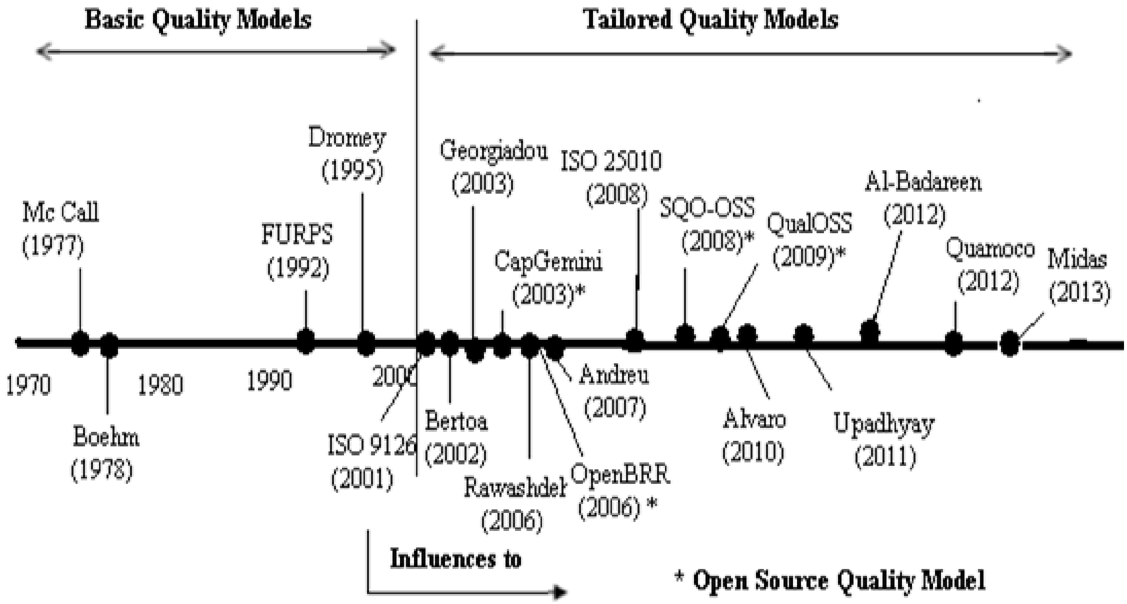

Figure 3.1. Evolution of quality models from the McCall until 2013

Source: *Taken from [110](Page32)*

The models to evaluate the quality of software have been **constructed** defining the fundamental factors (also called characteristics), and within each of them the subfactors (or subcharacteristics). Metrics are assigned to each subfactor for the real evaluation. So, quality model *is a set of selected quality characteristics with the assigned measures and the relationships between them relevant to a context that provide the basis for specifying quality requirements and* *evaluating quality of an entity*. The software quality models are designed to *allow developers a clearer understanding of the relationships between internal and external quality, ways to reduce the number of defects in software development, increase efficiency,* etc.

A quality model specifies *which properties are important for a product (e.g. usability, traceability,* etc.) *and how these properties are to be determined*. For each attribute, one or more quantitative or qualitative metrics can be defined in order to assess its value.

In addition, the quality model describes additional functional properties, such as *„how the software was created"* and *„how it works"*. The quality required for a software product must be defined in the software requirements definition document. Also, the definitions of quality attributes, measurement methods, and attribute acceptance criteria must be specified.

The detailed examination of these models is not the subject of this section, since they have been continuously refined, improved and incorporated, partially or totally, in the new models. *For this reason, just to exemplify we will give brief explanations some basic models, which are used for global assessments of a software product.*

3.1.1. McCall and Boehm quality models

At the end of the 1970's, the first hierarchical quality models were published by McCall *[101]* and Boehm et al. *[102]*. **The basic models** are used for global evaluations of software products. The basic models *are hierarchical;* they *can be adjusted to any type of software product* and are oriented to the evaluation and improvement. The most important hierarchical models are: McCall, Boehm, FURPS, FURPS+, Dromey, family ISO 9126 (internal quality, external quality and quality in use) and ISO 25010.

The main problem is that *these models are too abstract and too general for specific areas*, such as particular domain of application or Component-Based Systems Design or concrete type of IS *[100] (p. 128)*. In this sense, some authors have started to propose particular, tailored models and metrics for software components *[111]*. The different models use different quality characteristics and the main problem is that these models are too general for specific project type, so the main challenge is to find the required quality characteristics for each one of the information project types *[112] (pp.11-12)* .

Tailored quality models began to appear since 2001. The main characteristic is that they are specific to a particular domain of application and the importance of characteristic may be variable in relation to a general model. These models are built from the basic models, especially the ISO 9126, and lately on ISO 25010, with the adding or modification of subfactors and the goal to meet needs of specific domains or specialized applications. Identifying and quantifying the quality of software products is a first task in determining the quality of applications and ensuring the desired quality level *[100] (pp.120, 126)*. Based on them, the quality model is being built – an operational system of features designed for quality control and management in accordance with established objectives.

Estimation of IS quality will be correct if it is based on the „lifecycle - criterion quality - metric of quality" relationship. Due to this approach, it is appropriate to judge not only the quality criteria nomenclature, but also the dominant role and the content of each criterion in relation to the development stage of the IS. The suitable quality characteristics and their importance for these types of IS was selected based on the comprehensive literature review and normalized as result of highest frequency score, obtained from survey of professionals. Established value of corresponding weights of these quality factors allows a more accurate quality assessment for these type of IS.

The Mc Call model established product quality through several factors at the top level of the hierarchy, which are refined to criteria. These were grouped into three perspectives: *Product Operation (correct, reliable, efficient, integrity and usability), Product Review (maintenance, flexibility, and testing), and Product Transition (portability, reusability and interoperability).* The criteria are then quantified by metrics (at the 3^{rd} level, on the right).

The major contribution of the McCall model was to considerer relationships between quality characteristics and metrics *(Table 3.2)*. This model was used as base for the creation of others quality models. **The main drawback** of the Call Mac model is the accuracy in the measurement of quality, as it is based on responses of *Yes* or *No*.

Boehm model which is similar to McCall model, establishes large-scale characteristics that constitute an improvement over the McCall model, because it adds factors at different levels and shows the decomposition provided by Boehm model.

The general utility of a software product is decomposed to *portability, as-is utility,* and *maintainability*. These quality characteristics are further decomposed to low-level quality characteristics, such as *consistency, structuredness, and conciseness*. Boehm highlights the importance of measuring the low-level characteristics. He introduces several measures to this end, which are conductible through expert judgments.

Boehm and *McCall* models use a several primitive features that determine more high-level features *[100] (p.129)*. The decomposition of the higher-level features into primitive characteristics determines the displacement of interdependencies and redundancies between the characteristics of the high-level structure to the lower level structure. The McCall model explains the relationships between the 11 main considered characteristics.

Table 3.2

The relationship characteristics-subcharacteristics of the McCall model

Subcharacteristics \ Characteristics	Correctness	Reliability	Efficiency	Integrity	Usability	Maintainability	Testability	Flexibility	Portability	Reusability	Interoperability
Traceability	+					+	+	+		+	
Completeness	+	+			+						
Consistency	+	+				+	+	+		+	
Accuracy		+	X		+						
Error tolerance	+	+	X		+						
Simplicity	+	+	+			+	+	+	+	+	
Modularity			X			+	+	+	+	+	+
Generality		X	X	X				+		+	+
Expandability			X					+		+	
Instrumentation			X		+	+	+				
Self-descriptiveness			X			+	+	+	+	+	
Execution efficiency			+						X		
Storage efficiency			+				X		X		
Access audit			X	+							
Access control			X	+	+			X			X
Communicativeness			X		+					+	
Operability					+					+	
Training			X		+	+	+	+		+	
SO independence			X					+	+	+	+
Machine independence			X					+	+	+	+
Communications communality			X								+
Data communality				X						+	+
Conciseness	+		+			+	+				

+ = complementary relationships; X = conflicting relationships

Source: *Adapted by the author based on [101](Page 38)*

3.1.2. ISO 9126 and ISO/IEC 25010:2011 quality models

ISO/IEC 9126 is the international standard for software quality that has been agreed upon by a majority of the international community and upon which some countries, such as Japan, have decided to standardize. It defines a common language relating to software product quality and is widely recognized as such, at least in Europe, where a survey indicates that it is known by at least 70 percent of the IT community *[113;114]*. The fundamental objective of the ISO/IEC 9126 standard is to address some of the well-known human biases that can adversely affect the delivery and perception of a software development project. These biases include changing priorities after the start of a project or not having any clear definitions of „success”.

The ISO/IEC 9126 standard *[108]* is divided into four parts: *Quality model; External metrics; Internal metrics; Quality in use metrics.* ISO/IEC 9126 tries to develop a common understanding of the project's objectives and goals. The ISO 9126 Quality Model *[108]* is a hierarchical model similar to the McCall model, identifying six key features in delivering software quality: *functionality, reliability, usability, efficiency, maintainability and portability.* The quality in use aspects are referred to the *effectiveness of the product, productivity, security offered to the applications and satisfaction of users*. *Figure 3.2* shows a view of the relationship between internal, external and quality in use attributes.

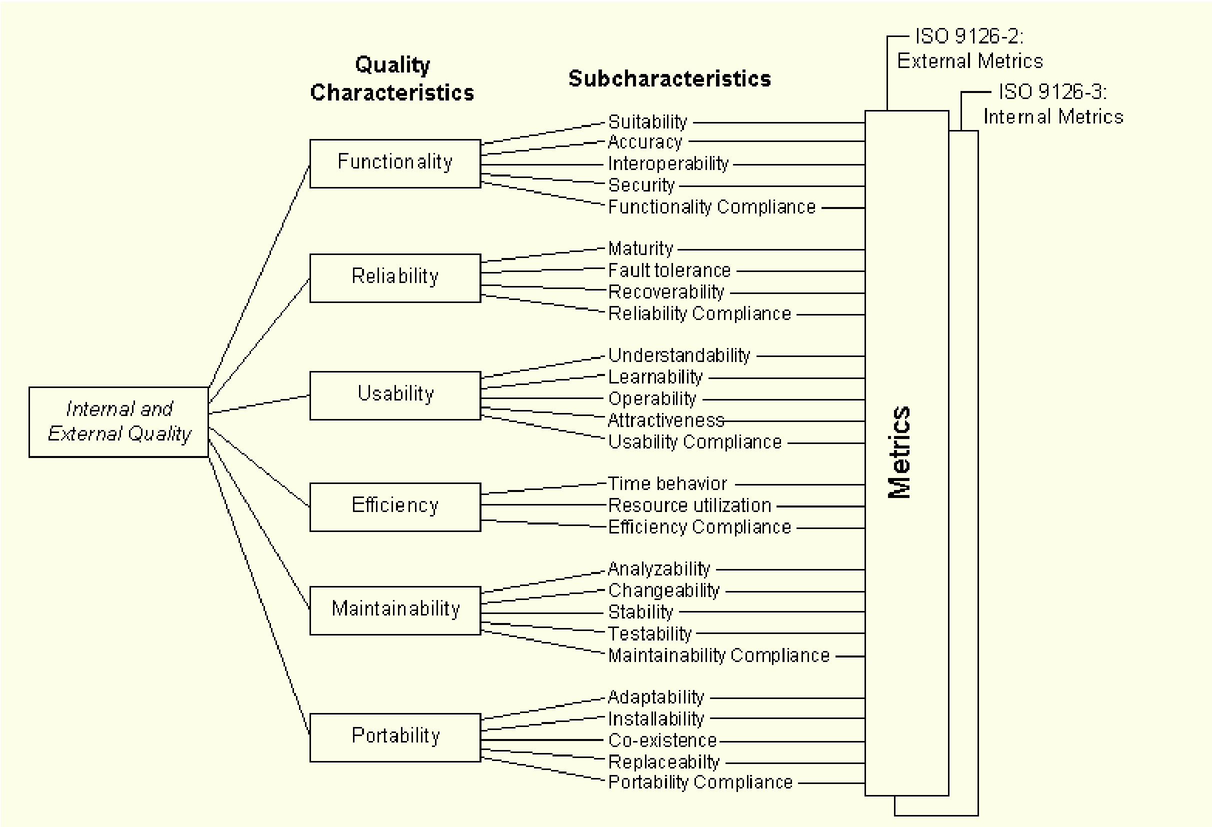

Figure 3.2. ISO 9126 External and Internal quality model

Source: *Taken from [115](page 10)*

The ISO 9126/1 model *[108]* and ISO 25010 model are more coherent in this content, because although there are *interdependencies* between features, they are not integrated into the model. In fact, for model ISO/IEC 9126, the lower-level structure of the model is not part of the standard, being only a proposal from ISO and the International Electro technical Commission.

After the publication of the ISO 9126, various adaptations and amendments have been proposed to it *[100] (p.130)*. Some authors adapt the quality attributes of ISO 9126 and add measures for quantifying them.

The ISO/IEC 25010:2011 model. This standard emerged in 2007 updating the ISO 9126 model, in 2011 has a second edition. Similar to ISO 9126 it distinguishes the model of **quality in use** *(5 characteristics and 11 subcharacteristics)* and the **product quality model** *(8 characteristics and 31 subcharacteristics). Table 3.3* illustrates the components of ISO 25010 models.

Table 3.3

ISO/IEC 25010: 2011: Quality Models, characteristics and sub-characteristics

Quality in use	Product quality	
1. Effectiveness **2. Efficiency** **3. Satisfaction** • Usefulness • Trust • Pleasure • Comfort **4. Freedom from risk** • Economic risk mitigation • Health and safety risk mitigation • Environmental risk mitigation **5. Context coverage** • Context completeness • Flexibility	**1. Functional suitability** • Functional completeness • Functional correctness • Functional appropriateness **2. Performance efficiency** • Time behavior • Resource utilization • Capacity **3. Compatibility** • Co-existence • Interoperability **4.Usability** • Appropriateness recognizability • Learnability • Operability • User error protection • User interface aesthetics • Accessibility	**5.Reliability** • **Maturity** • **Availability** • **Fault tolerance** • **Recoverability** **6. Security** • **Confidentiality** • **Integrity** • **Non-repudiation** • **Accountability** • **Authenticity** **7. Maintainability** • **Modularity** • **Reusability** • **Analyzability** • **Modifiability** • **Testability** **8. Portability** • **Adaptability** • **Installability** • **Replaceability**
5 characteristics and 11 subcharacteristics	**8 characteristics and 31 subcharacteristics**	

Source: *Developed by the author based on ISO 25010 [33](Pages 3-4)*

The ISO/IEC 25010:2011 model presents the *quality of use* covering the external characteristics and *Product Quality Model.* This model considers as new characteristics the *security* and *compatibility* that groups some of the former characteristics of portability and those that were not logically part of the transfer from one environment to another. It uses the term transferability as an extension of portability.

The first edition of this standard emerged in 2007 updating the ISO 9126 model. It is subdivided into 8 sub key features and characteristics. Constitute a set of standards based on ISO 9126 and one of its main objectives is to guide in the development of software products with the specification and evaluation of quality requirements. This model considers as new characteristics the *security* and *compatibility* that groups some of the former characteristics of portability and those that were not logically part of the transfer from one environment to another. It uses the term transferability as an extension of portability.

The quality models in this international standard can be used in conjunction with *ISO/IEC 12207* and *ISO/IEC15288*, particularly the processes associated with requirements definition, verification and validation with a specific focus on the specification and evaluation of quality requirements. ISO/IEC 25030 describes how the quality models can be used for software quality requirements, and ISO/IEC 25040 describes how the quality models can be used for the software quality evaluation process. This international standard can also be used in conjunction with ISO/IEC 15504, which is concerned with software process assessment.

The software quality models are used in following four steps.

The first step corresponds to the definition of the characteristics (indicators) of software quality, each of which reflects a user's individual point of view on quality. For example, ISO 9126 quality model of product includes six characteristics (quality indicators): *Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability*; ISO 25010 includes eight characteristics: *Functional Suitability, Performance efficiency, Compatibility, Usability, Reliability, Security, Maintainability, and Portability.*

The second step corresponds to the attributes/subcharacteristics for each quality characteristic, which detail the different aspects of a particular characteristic. A set of attributes of quality characteristics is used in assessing quality.

The third step is designed to measure quality using metrics. Each of them according to the standard is defined as a combination of an attribute measurement method and an attribute value measurement scale. To assess the quality attributes at the lifecycle stages (when viewing documentation, programs and program test results), metrics with a given estimated weight are used to level the results of a metric analysis of the total attributes of a particular indicator and quality in general. The quality attribute is determined using one or several assessment methods at the lifecycle stages and at the final stage of software development.

The fourth step is the evaluation element of the metric (weight), which is used to estimate the quantitative or qualitative value of an individual attribute of the software indicator. The selected attributes and their priorities are reflected in the requirements for the development of systems or the corresponding priorities of the software class to which the software belongs are used.

3.1.3. Enriched, tailored and open source models

Enriched models arise due to the shortcomings of hierarchical models. After the publication of ISO 9126 researchers proposed more elaborate quality models, to over-come known shortcomings of the previous quality models. The richer models mostly focus on single problems of the hierarchical models and improve them.

Dromey *[106]* introduced a distinction between product components, their properties, and externally visible quality attributes. As product components he uses the artifacts of the software or rather constructs of a programming language, such as expressions, variables, and modules. These components are attributed with properties, such as computable, precise, and structured. For each component property, such as computability of expressions, its influence on a quality attribute (taken from ISO 9126) is defined and explained by a rationale.

Tailored models began to appear the year 2001 with Bertoa model *[116]*, followed by Georgiadou Model in 2003 *[117]*, Alvaro Model in 2005 *[118]*, Rawashdesh Model *[119]*. These non-basic models include *Component Based Software Development/CBSD* and the software development concentrated on the use of *Commercial Off-The-Shelf Components/COTS.*

The main characteristic of tailored models is that they are specific to a particular domain of application and the importance of characteristics may be variable in relation to a general model. These models arise from the need of organizations and the software industry for specific quality models capable of doing specialized evaluation on individual components. They are built from the basic models, especially the ISO 9126, with the adding or modification of subfactors and the goal to meet needs of specific domains or specialized applications.

Open Source Models. Actually, free software products have much popularity for the diverse characteristics and freedoms they offer and because they are used in different contexts. Many of them are directed to perform the same or similar applications than traditional products. For example, they can be Free Software Operating Systems *(such Linux, Solaris, Free BSD),* middleware technologies/databases *(e.g. Apache Web Server, MySQL)* and products for the end user *(e.g. Google Chrome, Mozilla Firefox, Open Office).*

Models for assessing the quality of free software products adapt models like ISO-9126, adding some particular aspects of free software. It is noteworthy that although there is a distinction between models of first and second generation, an ideal model that captures all aspects of quality in a free software product has not been defined yet *[120]*. According to *[121]*, *[122]* the open source models started in 2003 and all of them emphasizes about the open source.

3.1.4. Critical analysis of quality models

There are different groups of characteristics included in the quality models:

- **Constructive** characteristics – are defining the adopted solutions, for example the characteristics of the main components of the program product.
- **Functional** characteristics – defining performance results, such as execution speed, yield, productivity, reliability, etc.
- **Economic** characteristics – a system of indicators that define the effort to achieve the product through costs, e.g. design, realization, exploitation, maintenance, elimination of potential errors, etc.;
- **Technological, identification, aesthetic, ergonomic, ecological** characteristics, etc.

Between quality characteristics, no matter what perspective they are viewed or grouped, *there are multiple relationships of interdependence, subordination, hierarchy, composition or aggregation*. The relationships between quality model characteristics are complex – **providing an aggregation approach for measurement values and multicriterial assessment producing comprehensible results is a big challenge.**

For the quality model to be operational in the sense of selecting a set of characteristics on which to build a metric system with which the quality of one or more software products is evaluated, the selected set has the following properties:

- To be fully appreciated by the evaluators in the sense of capturing all aspects of the quality that the evaluators are interested in. Evaluators (internal, beneficiaries, external auditors) are authorized and well-informed individuals interested in the qualitative assessment of the product, organization or the system.
- To be hierarchical in the sense that the main features are decomposed into factors which are quantified using metrics.
- To be consistent (non-contradictory). For example, complexity runs counter to reliability, efficiency to portability etc. In this case, the values of certain features may

be fixed, but others may vary within acceptable limits or left to the discretion of the developers, explicitly setting only the priority system.

The internal characteristics of the product process or resource, are those that are measured by examining the product *(e.g. number of code line)*, processes *(e.g. backup, restore)*, or resources *(e.g. processing time, memory volume)*, separate from their mode of behavior. **Internal quality** is the unobserved direct part of its product, reflecting the program's structure, coding, testing, and maintainability practices. From the point of view of the business, the low level of internal quality determines loss of reputation and competitiveness of the company.

External characteristics of the product, process or resource, are those that are measured only with reference to how the product, process or resource reacts with the environment. External characteristics of software quality directly affect product value to the user. **External quality is** observed by the user, is tested and serves as the acceptance or refusal criterion of the product. The low level of external quality can be manifested by system failure, unexpected behavior, data corruption, low performance, etc.

A fundamental axiom of software quality is the fact *that tangible internal product characteristics determine the quality of external characteristics, and for improving software quality it is necessary to improve internal quality*. *Establishing a link between the tangible properties of the product and the levels of the quality characteristics, as shown schematically in Figure 3.3 need the following steps:*

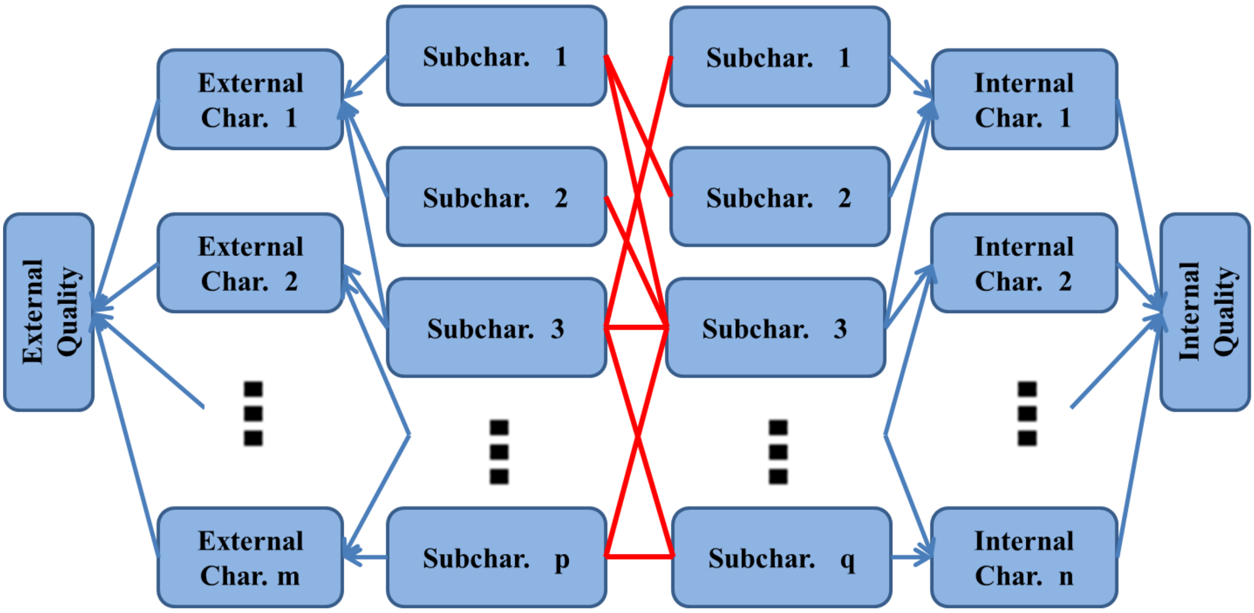

Figure 3.3. The relationship between three parts of model characteristics

Source: *Developed by the author*

- Definition and decomposition of external quality features, user-oriented step;

- Definition and classification of internal quality characteristics, stage- oriented towards the developer;
- Making detailed links between external and internal subcharacteristics.

Relationships of interdependence between quality characteristics, both in the same category and between different categories, are complex and difficult to quantify. This involves identifying conflicts between desired quality requirements and performance of activities to ensure a satisfactory equilibrium *(Table 3.4, Table 3.5, adapted by the author, based literature study).*

Table 3.4

Relationship between quality characteristics

Name of characteristics	Adaptability	Usability	Efficiency	Maintainability	Portability	Correctness	Completeness	Security	Robustness	Reliability	Reusability
Adaptability	0	0	0	0	0	0	0	0	0	0	0
Usability	+	0	0	0	0	0	0	0	0	0	0
Efficiency	-	-	0	0	0	0	0	0	0	0	0
Maintainability	0	+	+	0	0	0	0	0	0	0	0
Portability	+	+	0	0	0	0	0	0	0	0	0
Correctness	+	+	-	0	+	0	0	0	0	0	0
Completeness	0	0	-	0	0	+	0	0	0	0	0
Security	+	+	-	0	+	0	0	0	0	0	0
Robustness	+	+	-	0	+	0	0	0	0	0	0
Reliability	-	+	-	0	0	0	0	+	0	0	0
Reusability	+	0	-	0	0	+	+	0	-	0	0
Adaptability	+	+	0	0	+	+	0	+	0	0	0

Source: *Adapted by the author based on literature analysis*

***Legend:* + =** *positive influence, - = contradiction; 0 = indifference.*

Solving conflicts between quality characteristics involves:

1. Identifying conflicts and negotiating compromises between characteristics;
2. Diagnosing conflicts between characteristics based on early information.

A technique for establishing a compromise on quality characteristics that satisfies all participants in the development, marketing, and use of a software product is the negotiation technique based on the win-win spiral model *[123] (Figure3.8)*. A risk in engineering software requirements is to increase the level of a quality characteristic, for example *performance*, to the detriment of another at least or as important as *portability*. Many software projects have been abandoned because they had a poor set of quality requirements, even though they had well-specified interface and functional requirements.

Table 3.5

The correlation between the quality factors and the application domain

Quality characteristics / Domain of application	*Adaptability*	*Usability*	*Efficiency*	*Maintainability*	*Portability*	*Correctness*	*Completeness*	*Security*	*Robustness*	*Reliability*	*Reusability*
Generalized systems (e.g. SO)	C	C	X	X	C	C	X	C	X	X	C
General programs (e.g. compiler)	X	C	X	C	C	C	C	C	X	X	C
Systems with critical safety	X	D	X	C	C	X	X	X	C	X	C
Process control (e.g. automatic)	X	C	X	C	C	X	X	C	C	X	C
Interactive systems	C	C	X	C	X	X	X	C	X	X	X
Data protection	X	D	X	C	X	X	X	X	C	X	X
Real-time systems (e.g. atomic stations)	X	D	C	X	X	X	X	X	X	X	X
CAD, CAE, CAQ	C	C	X	C	C	X	X	C	X	X	X
Distributed systems (e.g. e-booking)	X	X	C	C	C	X	X	X	C	X	C

Source: *Adapted by the author based on literature analysis*

Legend: *C - required quality factor; D - desired factor, but not considered; X - irrelevant factor.*

The technique of spiraling negotiation consists in taking the negotiation to a higher level where it „starts at the end" and then, after the conditions are agreed, again transfers to a higher level and resumes the request to offer more advantageous conditions.

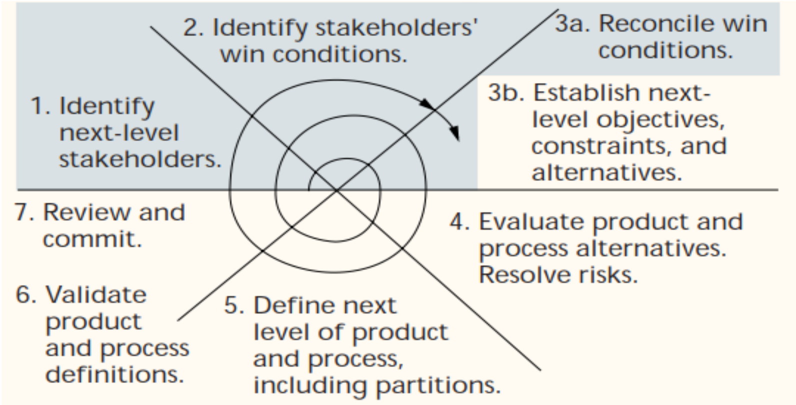

Figure 3.4. WinWin spiral model for solving conflicts between quality characteristics

Source: *Taken from [123](Page 34)*

As a result of the comparative analysis of the models *(Table 3.6)* we conclude, the major unsolved topics with regard to quality models and identify solutions for these as following:

1. Different models use different set of characteristics, and the models, in essence, cannot be compared with each other. The first problem is *providing a clear definition of high-level quality attributes and clear decomposition criteria.* It is **necessary to find the required quality characteristics for each one of the information project type**.
2. Standards quality models are too general for specific project type, and the second challenge is **relating the high-level quality attributes to concrete product characteristics that are measurable.**
3. **Lack of terminological consistency, Ambiguity, Incompleteness, and Overlapping.** The narrow specialization of the models gave rise to the use of different terms to describe equivalent phenomena. Some concepts appearing in the standards (and models) are not clearly defined. Some quality attributes of models are „ambiguously defined, incomplete with respect to quality characteristics and overlapping with respect to measured properties" *[124]*. A similar problem with the definition of the quality attributes is the lack of rationale for their selection and their hierarchical composition *[125]*. For example, the difference between adaptability and maintainability is unclear, because both refer to the modification of the product to satisfy certain needs *[126]*.
4. **The level of detail back affects the level of applicability**. Existing models are either abstract, widely applicable, or detailed and narrowly applicable. A high level of abstraction eliminates widespread use, and narrow applicability eliminates a high degree of detail.

Table 3.6

Comparison between some quality models by the characteristics

Quality Characteristic	McCall	Boehm	Ghezzi	FURPS	IEEE	Dromey	ISO 9126	QMOOD	SATC	SQuaRE
1. Accuracy			X				X			
2. Ambiguity										X
3. Availability/Reliability	X	X	X	X	X	X	X		X	X
4. Correctness	X									
5. Efficiency	X	X		X	X	X	X		X	X
6. Expandability								X		
7. Feasibility									X	
8. Flexibility	X		X					X		
9. Functionality				X	X	X	X	X	X	X
10. Human Engineering		X								
11. Integrity			X							
12. Interoperability	X						X			
13. Maintainability	X	X	X	X	X	X	X		X	X
14. Modifiability		X						X		
15. Modularity								X		
16. Performance				X		X				
17. Portability	X	X	X		X	X	X		X	X
18. Process Maturity						X				
19. Reusability	X		X			X		X		
20. Security	X						X			
21. Supportability				X					X	
22. Testability	X	X								X
23. Understandability		X		X				X		X
24. Usability	X		X	X	X	X	X	X	X	X

Source: *Adapted by the author based on literature analysis*

All of these problems can be solved through tailored models, obtained from generic, adaptable, flexible and extensible, but unique quality metamodel, which includes knowledge of the best known's quality models and QM best practices, set in standards. In old approach user is felt compelled to define clearly each concept of the standard and the relationships among them, when he intended to use a model. In a new approach, when user intended to use some basics or tailored models, is not felt compelled to redefine each concept of the quality factors and the relationships among them, because these are defined in knowledge quality metamodel.

Despite the enriched quality models discussed here address single issues of the hierarchical models, a comprehensive model solving all issues is still missing. Both the main challenges, the definition of quality attributes and the operationalization of a quality model, are still unsolved. For instance, the activity-based quality models provide a clear way of defining quality attributes. However, they have so far only been applied to maintainability, usability, and security.

Regarding the usage of software measures for quality assessments there are three major critiques:

1. Despite the large number of software measures, it is still unclear how they relate to quality in general or to specific quality attributes. Hence, their gainful application in quality assurance is still unclear.

2. Additionally, measures are often defined based on available data, instead of actual measurement goals *[127]*. Thus, most measures are defined by source code, because source code is usually available and easily accessible.

3. A third topic is the validity of software measures. Generally, the validity depends on the objective pursued with the measure. As we have seen, for the prediction of faults, software measures are of limited use. Though fault prediction is just one single topic in the wide area of quality assessments. In general, for assessing the quality of software, it must be assured that the measures conform to the measurement goal. *To achieve conformance, measurement frameworks have been introduced.* The most prominent of them, the Goal-Question-Metric, is discussed in the following subsection.

„Significant benefits can be achieved from the integration of information management, knowledge management and quality management in the form of improved profitability and improved customer satisfaction" [128]. Quality management of IS, at the operational phases, isn't possible without permanently feedback about the user's satisfaction on the quality (usefulness, relevance, timelines, etc.).

3.2. Quality tools

A large number of quality tools have been developed. In the literature, different categorizations and collections of such tools have been published [22], [34]. The full description of the quality instruments is not an objective of the present work. We will mention shortly only some of them, more main or recommended for the intended application.

3.2.1. Seven classical quality tools

The classical tools of quality are those that allow us to work on visible and measurable facts, starting from numerical data and in a curative approach. They also allow to some extent the analysis of the causes. Traditional/classical tools of quality management are statistical analysis tools that are designed primarily to treat established numerical data, starting from visible or measurable facts. They are:

1. Data collecting form (Tracking sheet, Inspection sheet)
2. Histogram
3. Pareto chart
4. Scatter diagram, called also Correlation Diagram
5. Control Charts/Check list
6. Cause-effect diagram (ISHIKAWA)
7. Brainstorming, Mind Mapping

For example, techniques that can be used for data representation according *PMBOK [22]* include, but are not limited to:

Affinity diagrams – can organize potential causes of defects into groups showing areas that should be focused on the most.

Mind mapping consolidates ideas created through individual brainstorming sessions into a single map to reflect commonality and differences in understanding and to generate new ideas. Mind mapping is a diagrammatic method used to visually organizing information. A mind map in quality is often created around a single quality concept, drawn as an image in the center of a blank landscape page, to which associated representations of ideas such as images, words, and parts of words are added. The mind-mapping technique may help in the rapid gathering of project quality requirements, constraints, dependencies, and relationships.

Flowcharts are also referred to as process maps because they display the sequence of steps and the branching possibilities that exist for a process that transforms one or more inputs into one or more outputs. Flowcharts show a series of steps that lead to a defect, the activities, decision points, branching loops, parallel paths etc. When flow charts are used to represent the steps in a process, they are sometimes called process flows or process flow diagrams and they can be used for process improvement as well as identifying where quality defects can occur or where to incorporate quality checks.

Histograms show a graphical representation of numerical data. Histograms can show the number of defects per deliverable, a ranking of the cause of defects, the number of times each process is noncompliant or other representations of project or product defects, values of characteristics etc. A histogram is a common data analysis tool in the business world. It’s a column chart that shows the frequency of the occurrence of a variable in the specified range. It’s similar to a Bar Chart, but a histogram group’s numbers into ranges. There are many graphical methods of representation numerical data (i.e. Column chart *Fig. 3.5*, Radar chart *Fig.3.6*) for various uses.

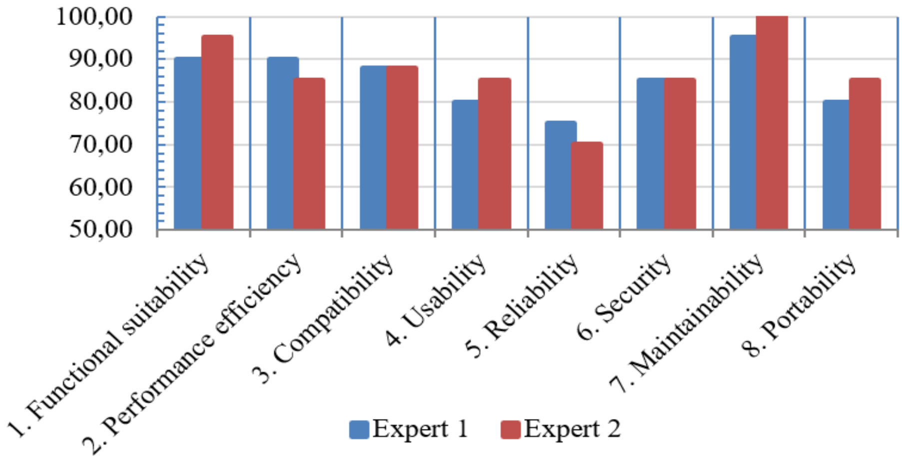

Figure 3.5. Quality of product according ISO 25010 (two evaluations, Column chart)

Source: *Developed by the author*

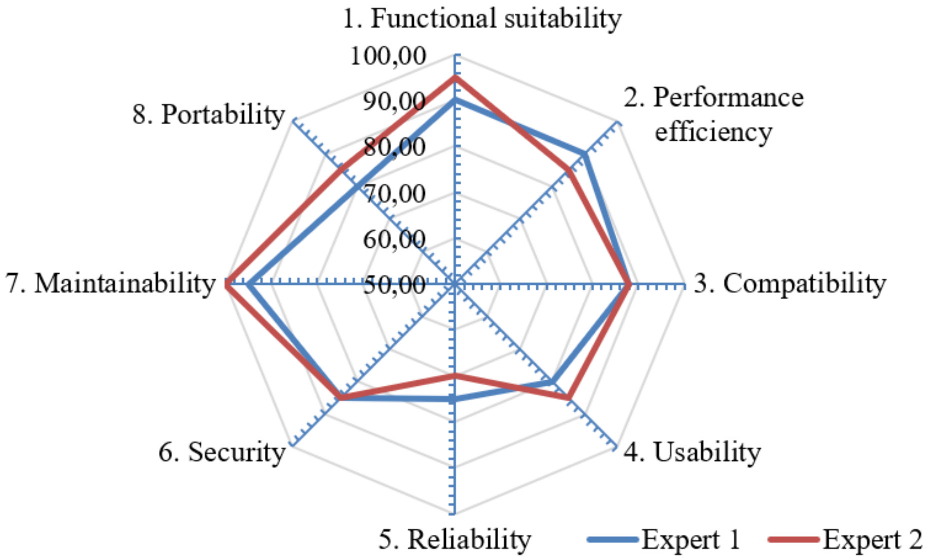

Figure 3.6. Quality of product according ISO 25010 (two evaluations, Radar chart)

Source: *Developed by the author*

A scatter diagram *(Figure 3.7)*, is a graph that shows the relationship between two variables (X, Y). Scatter diagrams can demonstrate a relationship between any element of a process, environment, or activity on one axis and a quality defect on the other axis. Y can be dependent on X but X not dependent on Y. So, there are many types of correlations such as positive correlation (proportional), negative correlation (Inverse) or pattern of no correlation (Zero Correlation). An example of positive correlation is the weight of the human and its relation to his age (between one year and 40). We find that the weight of the human depends on and is affected by age (increase in age will lead to increase in weight but the reverse is incorrect). So, we will consider the age is X axis and the weight is Y axis. In a negative correlation, when X is increased, Y decreases.

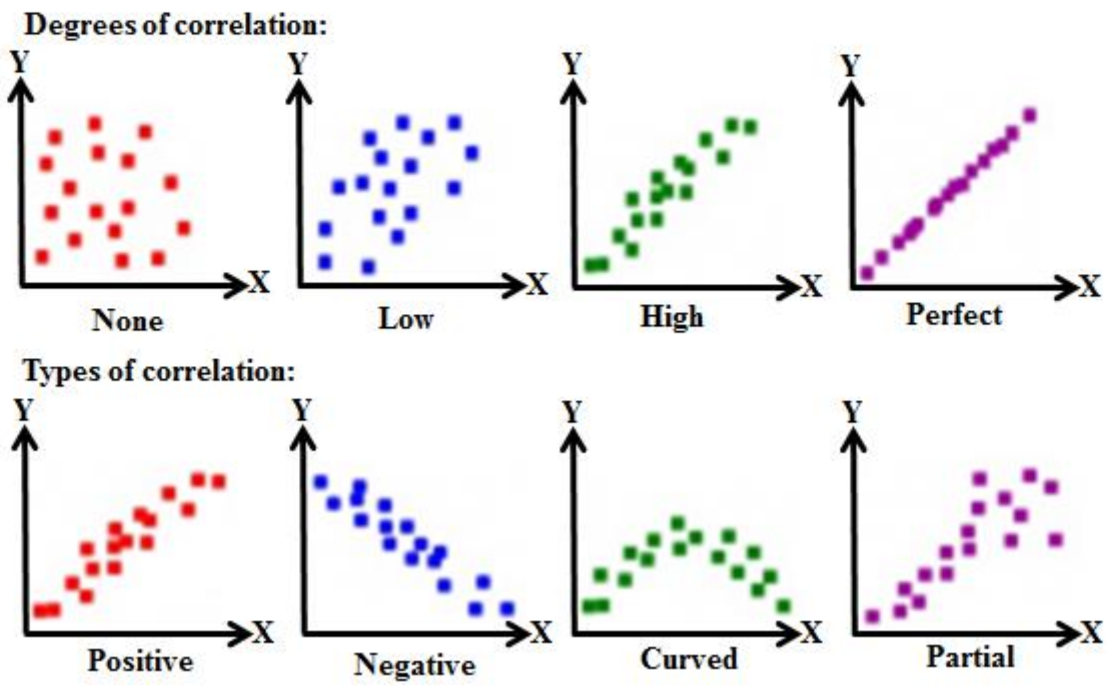

Figure 3.7. Example of Scatter diagram

Source: *Adapted by the author based on [129](Web-page)(format axis)*

Control Charts *(called also Check list, Check sheets or Tally sheets.* Check sheet is a sheet that contains items of inspections, tests and the attribute that each test can result in. For example, the acceptance criteria of each test must be listed on the sheet to be a guide for determining if the inspected item of the sample such as- a piece of code in the software project has passed a test item (such as a unit test). Then we gather the frequencies of each defect and represent it in Pareto Chart.

Pareto chart *(Figure 3.8).* The purpose of the Pareto chart is to highlight the most important among a (typically large) set of factors. In quality control, it often represents the most common sources of defects, the highest occurring type of defect, or the most frequent reasons for customer complaints, and so on. In fact, that's called the 80/20 rule and applies to many other disciplines and areas. It goes on to say that 80 percent of the problems are usually caused by 20 percent of the causes.

This particular Pareto chart in *Figure 3.8* divides problem areas into nine categories, which are always ordered from most to least faults found. The vertical axis on the chart shows the cumulative percentage and the horizontal –the problems. We observe that most of the product problems are caused by *Cause #1* and *Cause #2*. Solving these 2 issues we can improve quality near 80%. *The main idea of the diagram does not consist in the magic numbers of 80/20. They can be different, for example, 70/30. The main idea is to prioritize tasks, problems etc.*

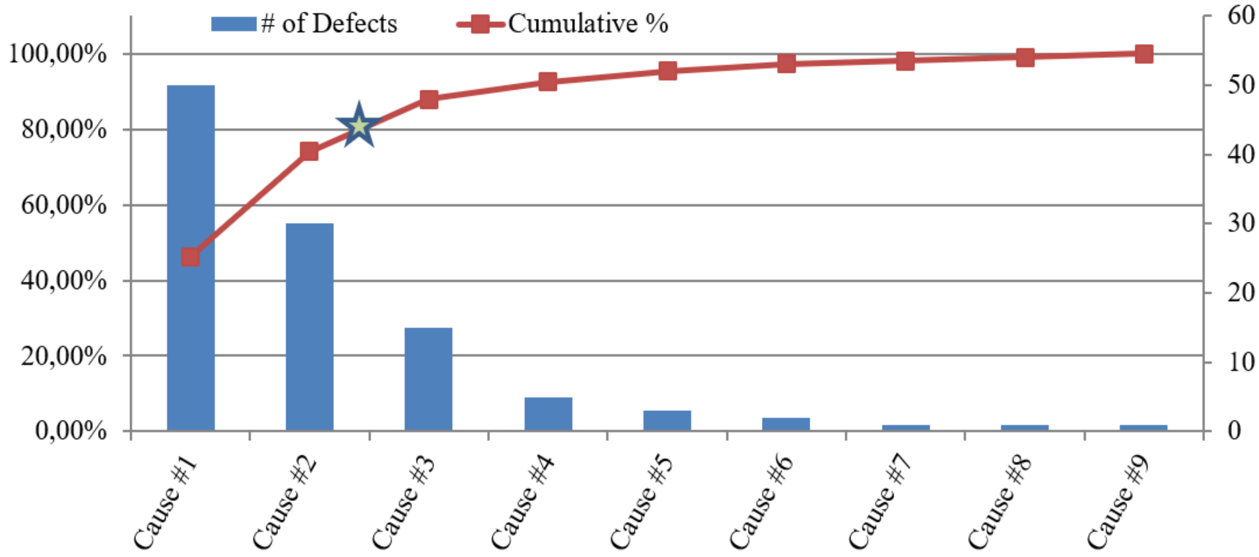

Figure 3.8. Example of Pareto diagram

Source: *Developed by the author based on [130](Web-page)*

Cause-and-effect diagrams *(Figure 3.9)* are also known as fishbone diagrams, why-why diagrams or Ishikawa diagrams. This type of diagram breaks down the causes of the problem statement identified into discrete branches, helping to identify the main or root cause of the problem.

The given diagram illustration simply helps us list out the possible causes of the cost over-run. Once the root cause is determined, it could be handled with appropriate resolution and may help in damage control. It could help the project manager save some cost during the future phases of the project.

It is worth mentioning, that many organizations have created their own methods of dealing with a problem in order to integrate it into their own culture. This results in seemingly distinct presentations of generally identical methods.

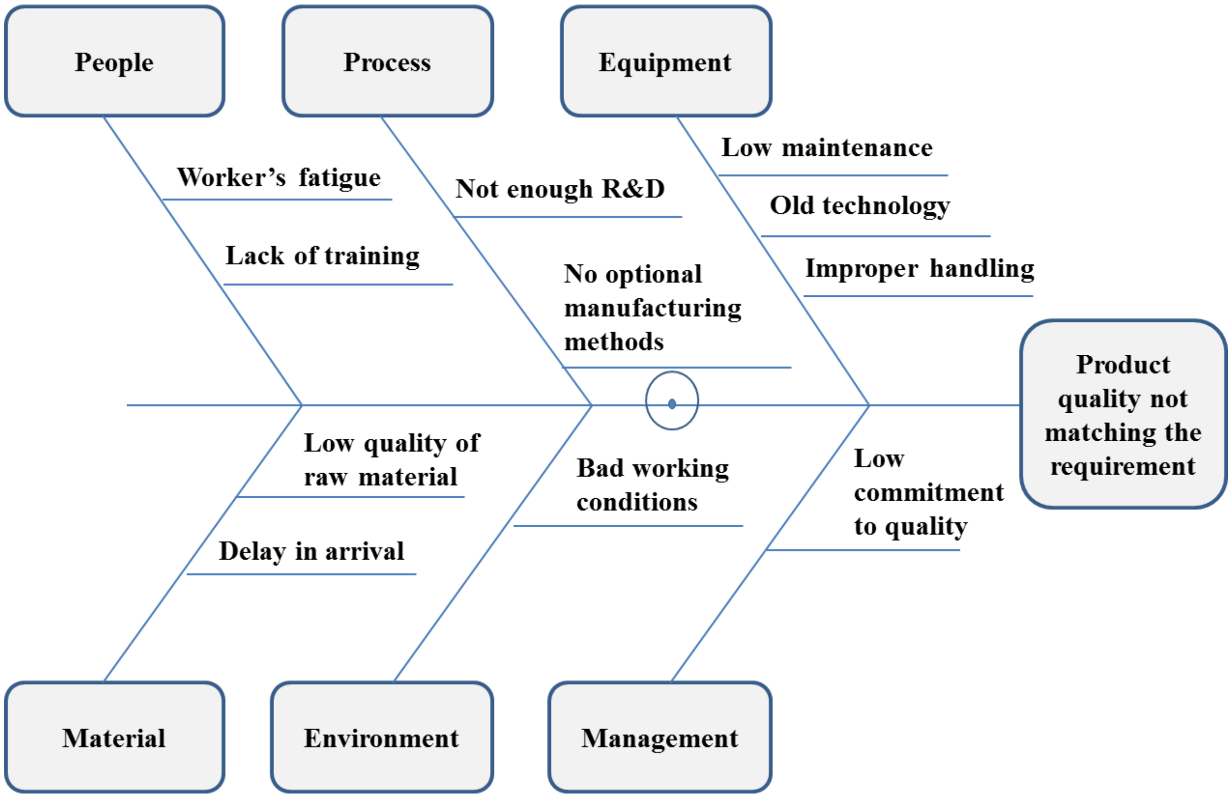

Figure 3.9. Cause ↔ effect diagram (Ishikawa)

Source: *Adapted by the author based on [131](Web-page), [22](Page 294)*

3.2.2. Deming's Wheel

Plan-do-check-act and Six Sigma are two of the most common quality improvement tools used to analyze and evaluate opportunities for improvement. Regarding the basic quality improvement initiatives such Total Quality Management (TQM) and PDCA cycle, defined by Shewhart and modified by Deming was examined in Chapter II. Regarding the quality tools such Six Sigma, Lean Six Sigma, which may improve both the quality of project management, as well as the quality of the end product, service, or result –these are not part of the body of the work.

Here we will briefly comment on the instruments superimposed on Deming's wheel.

Deming's Wheel is now considered in the West as in Japan as the universal method, the basic cycle of any improvement in quality, either solving a problem. It is a looping principle, recursive, so more satisfying than a linear, too detailed method, less natural and possibly too rational. Each of the four phases of the PDCA cycle occupies a wheel dial, and in some sectors some of the classic tools used are indicated *(Figure 3.10).*

Planning: analysis of the facts by *„Tracking sheet", „Inspection sheet", „Histogram", „Pareto chart", „Correlation chart";* Analysis of causes by *„Ishikawa diagram"*. **Do**: no classical tool is intended for this phase of solution determination, only the „*Gantt chart"* can be attributed. **Check:** evaluation of the effects of the solutions established by the *„Tracking sheet", „Inspection* *sheet", „Histogram", „Pareto chart", „Correlation chart".* **Act:** as for the realization phase there is not a specific instrument for this phase among the classical instruments.

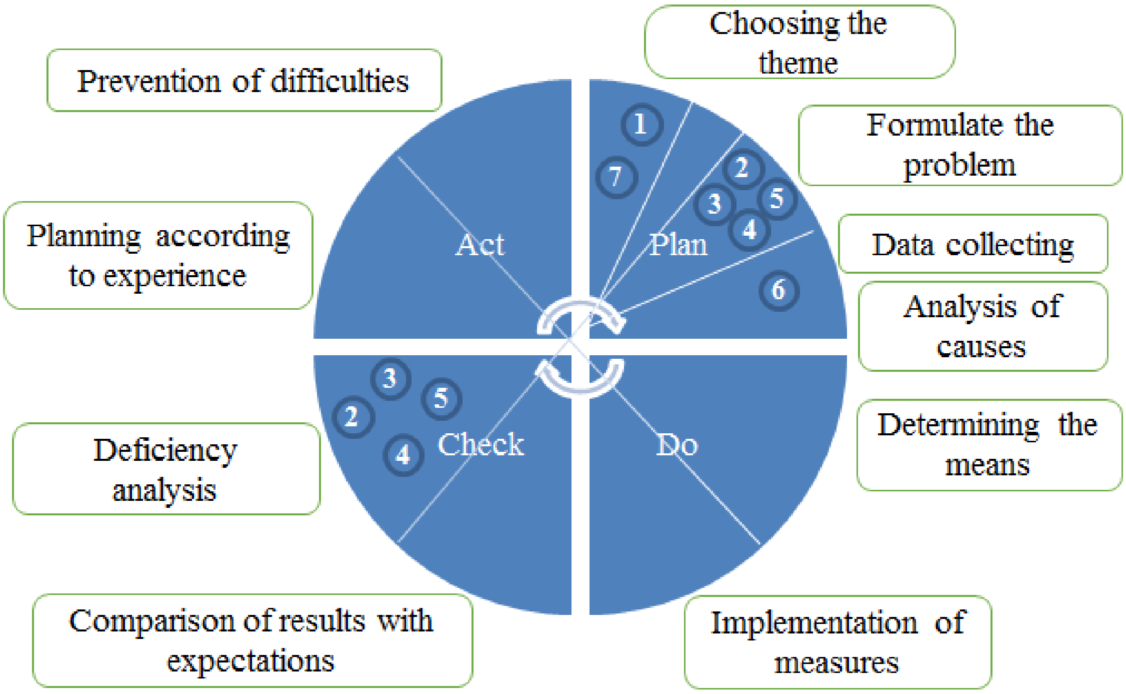

Figure 3.10. Classic tools used in the PDCA cycle to improve quality

Source: *Developed by the author based on [71] (Page 5), [72] (Section 29, page 19), [34] (Page71)*

Legend: *1 - Brainstorming; 2 - Tracking sheet; 3 - Inspection sheet; 4 - Histogram; 5 - Pareto; 6 - Ishikawa; 7 – Correlation diagram*

3.2.3. Project and quality management information system

Project management information system (PMIS, in *PMBOK [22]*) provides access to information technology software tools, such as scheduling/reporting software tools, work planning, information collection and distribution, etc. Automated gathering and reporting on key performance indicators can be part of this system.

The PMIS can also include spreadsheets, simulation software, and statistical analysis tools to assist with cost estimating. Such tools simplify the use of some cost-estimating techniques and thereby facilitate rapid consideration of cost estimate alternatives.

PMIS can also ensure that stakeholders can easily retrieve the information they need in a timely way. Project information is managed and distributed using a variety of tools, including:

- Electronic project management tools. Project management software, meeting and virtual office support software, web interfaces, specialized project portals and dashboards, and collaborative work management tools.

- Electronic communications management. Email, fax, and voice mail; audio, video and web conferencing; and websites and web publishing.
- Social media management. Websites and web publishing; and blogs and applications, which offer the opportunity to engage with stakeholders and form online communities.

PMIS examination is not an objective of the thesis. We will just remember some of them, regarding QM and developed application

Minitab is a statistics package developed at the Pennsylvania State Universityby researchers Barbara F. Ryan, Thomas A. Ryan, Jr., and Brian L. Joiner in 1972. It began as a light version of OMNITAB 80 (https://www.nist.gov/itl/sed/omnitab-80) *[132]*, a statistical analysis program by NIST, the precursor of Minitab. Statistical analysis software such as Minitab automates calculations and the creation of graphs, allowing the user to focus more on the analysis of data and the interpretation of results.

Minitab 19 products (https://www.minitab.com/en-us/) *[133]*, help businesses increase efficiency and improve quality through smart data analysis. Minitab can help customers spot trends, solve problems and discover valuable insights in their data.

Minitab's Assistant offers opportunities for, but not only: Measurement systems analysis, Capability analysis, Graphical analysis, Hypothesis tests, Regression analysis, Control charts

3.2.4. Tools for project management

Today the market has more than a hundred solutions for project management, each has strengths and weaknesses. Among the Agile project management tools, the most popular are:

- Jira
- TFS
- Version One
- Rally software
- Spreadsheet - Google Docs etc.

A short comparison between Agile management software tools is shown in *Table 3.7.*

Table 3.7

Comparison between Agile management software tools

Feature Overview	JIRA	Version One	Rally Software	TFS	Google Docs
License	Proprietary/Free community licenses for open source and academic projects	Proprietary, hosted	Proprietary/Free trial	Proprietary, Commercial	ICU license
Price	Multiple price points/Free trial	Multiple price points/Free trial	Multiple price points/Free trial	Multiple price points?	Free
Platform	Web-Based/Installed	Web-Based	Web-Based	Web-Based/Installed	Web-Based
Intended Users	Freelancers, Large Enterprises, Mid-Size Business, Non-Profits, Public Administrations, Small Business	Freelancers, Large Enterprises, Mid-Size Business, Non-Profits, Public Administrations, Small Business	Freelancers, Large Enterprises, Mid-Size Business, Small Business	Large Enterprises, Mid-Size Business, Small Business	Freelancers, Small Business
Drag-and-drop Backlog Management	Full support	Full support	Full support	Full support	No
Story points					
Task board view	Yes	Yes	Yes	Yes	Yes
Iteration burn down chart	Yes	Yes	Yes	Yes	None
Epics (hierarchy of backlog items)	Partial support	Full support	Partial support	Partial support	None
Rollups support	Partial support	Full support	Partial support	Partial support	None
Release and Iteration Planning and Tracking	Partial support	Full support	Full support	Full support	Partial support

Feature Overview	JIRA	Version One	Rally Software	TFS	Google Docs
Product Roadmapping (multiple releases)	None	Full support	Full support	None	Partial support
Multiple products/ projects	Full support	Full support	Full support	Full support	Partial support
Portfolio planning	None	Full support	Full support	Partial support	None
Test Management (Acceptance and Regression)	Partial support	Full support	Full support	Full support	Partial support
Automated Notifications of Changes to System Assets	Email	Email	Email	Email	No
Impediment tracking	None	Full support	Full support	Full support	Partial support
Defect Tracking	Partial support	Full support	Full support	Full support	Partial support
User roles	None	PO, SM, Team Member, Stakeholder, plus custom roles.	SM, PO, Team Member.	None	No
Integration, API (s), SDK	Yes (REST API)	SDK.Java, SDK.NET, SDK.Python, SDK.Javascript	SDK.Java, SDK.NET, SDK.Ruby, SDK.Nodejs	SDK.Java, SDK.NET	SDK.Java, SDK.NET
Support	Email/Phone Community Website	Email/Phone Community Website	Email/Phone Community Website	Email/Phone Community Website	Forums

Feature Overview	JIRA	Version One	Rally Software	TFS	Google Docs
Service	None	Training and Certification	None	Training and Certification	None
User docs	****	**	**	**	***
Usability	***	**	***	***	***
Pros	Big community, Multi – language support, 600+ plug ins and add-on, mobile	Free trial for up to 10 users Supports collaboration for cross-functional teams Robust planning abilities; tracking Epics, Stories and Projects	Supports collaboration for cross-functional teams Provides story and feature roll-up for enhanced program and portfolio management Includes integrated defect management	Useful features for managing Agile processes	Good tool for small teams and small processes
Cons	Poor backlog, sprint management tools Lacks burn down and resource reporting	Complex User Interface Not mobile ready Higher learning curve needed to understand all features	Complex User Interface Requires additional process for linking stories and features to higher-level portfolio items Lacks configurable out-of-the-box reporting	It is desirable to use other Microsoft tools for development	Manual work

Source: Adapted by the author based on *[134](Web-page)*

Conclusions on chapter III

Software quality models and their metrics may be used in many contexts, for instance, during the development of a new application or when selecting commercial components. Given the completeness and significance of the criteria, quality models can be subdivided into ***basic (abstract, generic models)*** in accordance with the requirements of international standard (ISO, IEEE etc.), and ***corporate (private, typical, tailored) models*** in accordance with the requirements of national, local or specific QMS standards in according to the context or/and type of the project.

All of examined models can be used in certain conditions, but the most actually and preferred basic model is the ISO 25010 model, which incorporates the best of the previous models and which is recognized at the global level. Even if some standards have been dropped, models can be applied, but certification can't be obtained, which is only active for the standards in force.

The overall conclusion is that there are very general models for assessing software quality and hence they are difficult to apply to specific cases. Also there exist tailored quality models whose range is in small domain, using as starting model the ISO 9126 and models for Free/Open source emphasize the participation of community members. The level of detail back affects the level of applicability of quality models. Existing basics models are either abstract and widely applicable, or detailed and narrowly applicable tailored models. A high level of abstraction eliminates widespread use, and narrow applicability eliminates a high degree of detail.

Finally, we note that in most of the studied models the factors and criteria have the same value which is relative because it depends of the application domain. For example, aspects of transferability can be crucial in software that is installed on different machines. However, they have so far only been applied to maintainability, usability, and security.

Despite the quality models discussed here address single issues of the hierarchical models, a comprehensive model solving all issues is still missing. Both the main challenges, the definition of quality attributes and the operationalization of a quality model, are still unsolved. For instance, the activity-based quality models provide a clear way of defining quality attributes.

Assumption. As a result of the conclusions drawn from the basic and tailored quality models analysis, we consider that both of quality factors of models may be managed in a similar manner, providing a unified framework/metamodel**.** These concepts enclosed in the standard, enriched with the decisions outlined in the precedent items, form together a rich base of knowledge for building a metamodel. But the development of quality model with a set of fit to IP metrics is far from being a simple task. In the next section we define a metamodel for quality assessment for a different type of IS.

It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change.

Charles Darwin

I can't change the direction of the wind, but I can adjust my sails to always reach my destination.

Jimmy Dean

IV. FIELD RESEARCH ON INFORMATION SYSTEMS QUALITY

In the fourth chapter, we report on a survey we conducted to assess the state of practice regarding quality models and quality assessment, the built metamodel and architecture of application to support of these.

The processes of selecting and establishing metrics and scales for describing the characteristics of the quality of software can be divided into two follow stages.

The first stage – selection and justification of a set of input data reflecting the general features and stages of the software project lifecycle and its customers, each of which affects certain quality characteristics. Obviously, we are talking about five Garvin's different points of view. SQuaRE quality models provide a framework for collecting stakeholder needs, including three point of view (for detail see *[33]*):

1. Primary user: person who interacts with the system to achieve the primary goals.

2.Secondary users who provide support, for example (a) content provider, system manager/administrator, security manager; (b) maintainer, analyzer, porter, installer.

3. Indirect user: person who receives output, but does not interact with the system.

Each of these types of user has needs for quality in use and product quality in particular contexts of use. For example, view on Garvin *[36]*, and in some other cases.

The second stage –selection, establishment and approval of specific metrics and scales for measuring the characteristics and attributes of the quality of a project for their subsequent evaluation and comparison with the requirements of specifications in the process of qualification tests or certification at certain stages of the software lifecycle.

Common understanding all point of product view, quality in use and data quality *(user view, customer satisfaction, manufacturing view, etc.)* is an issue, which can be solved by tailoring models.

The main purpose of the research in general is providing quality of information projects. To do that, there is a need to identify the quality characteristics relevant to the types of projects. For this purpose, it was necessary to collect the information from the participants (of the field research) in order to review the quality characteristics of the information projects and further to assemble a model for implementation, management and measurement for diverse types of information projects. While the goal is to build a model, which will contain quality characteristics suitable for each of information projects according to its type. From all the mentioned above, the objective of the field research was to find out which of the quality characteristics will be taken into the quality model with respect to the research results from the literature review and the field study.

The purpose of the field research examination of the research relevancy is to examine whether the common use areas in which information systems / projects information are widely agreed (to define broad agreement property - receiving a grade greater than 3.5) have influence on a number of characteristics to quality measure so that we can be defined as a successful model to evaluate the quality of the data / information project.

The suggested model is aimed to classify information projects from the quality point of view and improving the quality in a way that can be quantified and measured,

In addition, a new model was built and implemented in a software application, based on the research results dealing with meta-analysis of the most actual researches and tracking the 25 most important information quality characteristics. This software application enables to manage quality of information projects by performing quality assessments, in accordance with the quality characteristics for each type of information project.

4.1. Field research methodology

Despite the number of Standards and Quality Models, little information exists how to measure, collects the values of characteristics and how evaluate a concrete type of IS. There is a need for research providing information about significance of each quality characteristic included in quality model and the weighting of the characteristics.

The purpose of the field research was **to building a metamodel** and find out which of the quality characteristics will be taken into the quality model for some type of IS, with respect to the research results from the literature review and the field study.

For this, it was needed:

- Select the most common quality characteristics, important to software, allow for uncomplicated measurement and do not overlap (based on bibliographic study and comparative analysis for the most important types of IS). Theoretically, metamodel can include also the characteristics of Infrastructure quality, Quality of services, Process quality, Quality of organization, Maturity level etc.;

- Field assessment of the significance / weights of each characteristic for each type of IS, that measure the degree of satisfaction (based on an expert opinion poll);
- Identification of generally accepted characteristics from experts for different IS *(with a score greater than 3.5 out of 5);*
- Build an adaptable metamodel, which consists of the set of generally accepted characteristics, and the possibility of modify them according to the user's demand.

Objective and subjective assessment methods have been used. Objective assessment measures the extent to which IPs conforms to quality specifications and references. Subjective assessment measures the extent to which IPs is fitness for use by end consumers.

The study sample comprised dozens of employers in the field of information systems and other experts as a control group. The experts are a group of system analysts working in a governmental organization. All the participants answered the question relating to their information technology field experience (questions 1-3 in the questionnaire).

The survey meant to obtain practical data from experts. The surveys were conducted with the participants and a case study was prepared based on the participants' perspective of the project. The survey comprised from25 different types of 25 quality characteristic measures based on the literature review and different types of projects. The questions were developed from a review of risk and success factors in the research literature on project management and information system management. The use of surveys enables us to study a greater number of variables (Quality Characteristics) than in the case of experimental approaches. In this study, the survey is an effective method to validate Information Quality Characteristics by factor analysis. Answers to the questions reflected participants' perceptions of the project. The participants were asked to classify the importance of the factors and characteristics by choosing a value from 1 to 5 of the Likert measurement scales, where 1 stands for "Very Important" and 5 stands for "Unimportant".

For each type of the information project the respondents needed to classify the importance of each quality from a list of 25 provided. Each case study was qualitatively analyzed to identify thematic patterns and artifacts that appeared to be relevant or important in enabling or inhibiting the performance and/or outcome of the project. This survey allowed to check the consistency of the quality characteristics for each information project or to point out on a certain trend. In addition, reviews a group of 13 experts in the field to indicate by the same survey in general, that is, not for a specific information project.

4.1.1. Data collection, field research results and findings

Data collection. In the research they were collected the feedbacks to the surveys during the period between the years 2014 to 2015.

The data was recorded and summarized as responses were received. The results were organized in the Microsoft Excel spreadsheets using excel statistical functions to measure the attitudes from the data of the survey results. The data was organized into separate rows and columns for each information project type or the expert's response. The responses to each question have been assigned with numerical values for the data analysis. Following data collection, it is possible to make analysis of factors and the average calculations and find trends for verifiable the results. After that it is possible to classify the items into a valid structure, to develop the quality model for concrete type of the information projects.

The form of survey completed by the respondents is presented in *Annex 3.*

Results. This section reports main findings of the study. First, findings relating to key practice areas are outlined. Second, the major risk factors found in the study are described. Third, an unexpected finding of multiple project types, each with different implications for risk and project management, is described and discussed.

The survey's respondents' results, regarding the quality importance of each characteristic, are displayed in the below tables, in accordance with the information project type and the experts' group. The tables below display the characteristics importance average, the number of answers and the standard deviation.

Respondents were asked to specify the level of significance of the quality characteristics they consider to be the most important in the process of assessing the quality of those information systems.

The detailed results of **Mean score** and **Standard deviation** calculus for 7 groups (12 type) of IS *(ERP & CRM Group, Geographic Information Systems Group, Enterprise Portal & Knowledge Management Group, BI & Big Data, Internet Sites & Web Application Group, Document Management Systems Group, Mobile Application Group*) are presented in *Annex 4.*

Summarized results by these seven groups, media by experts and average of the mean scores is show in *Figure 3.9* and *Figure 4.1, Figure 4.2.*

Mean score and Standard deviation formulas:

AVERAGE is the arithmetic mean, and is calculated by adding a group of numbers and then dividing by the count of those numbers.

Table 4.1

Field Research Average of the Groups and Mean Scores (means)

Characteristics	ERP & CRM	GIS & Map Library	Enterprise Portal & Knowledge Management	BI & Big Data	Internet Sites & Web Application	Document Management System	Mobile Application	Experts	Average of the Means
1. Accuracy	4.7143	4.4667	3.5385	4.0000	3.4706	3.9286	3.6471	4.0769	3.9803
2. Availability	4.3182	4.3077	3.6667	3.6000	3.6250	3.3846	4.2353	3.7692	3.8633
3. Changeability	3.0455	2.7692	3.2500	3.5000	2.9375	3.0769	3.3529	3.5000	3.1790
4. Correctness	4.1818	3.6429	3.3636	3.6154	3.4375	3.1818	3.3529	3.9231	3.5874
5. Efficiency	3.6364	2.9231	3.8000	3.8333	3.6000	3.2500	4.3125	3.7692	3.6406
6. Flexibility	2.9565	2.8333	3.0000	3.0000	2.6000	2.4545	2.8235	3.6667	2.9168
7. Functionality	4.0000	3.8333	3.5000	3.4545	3.6000	3.2727	3.9375	4.3846	3.7478
8. Interface facility	3.5652	3.7273	3.2222	3.0833	2.6429	3.1000	3.9333	3.5833	3.3572
9. Integrity	4.0435	3.3636	3.0000	3.6364	3.0714	3.6667	2.9231	3.1818	3.3608
10. Interoperability	3.4545	2.3333	3.3750	3.2000	2.6154	3.2222	3.6154	2.9091	3.0906
11. Maintainability	3.7391	3.1667	3.5556	3.0000	3.5000	3.3000	4.0714	3.6154	3.4935
12. Modifiability	3.3636	3.0000	2.8889	3.1818	3.0000	2.6000	3.3846	3.0000	3.0524

Characteristics	ERP & CRM	GIS & Map Library	Enterprise Portal & Knowledge Management	BI & Big Data	Internet Sites & Web Application	Document Management System	Mobile Application	Experts	Average of the Means
13. Performance	3.6087	3.9091	4.3000	3.4545	3.6000	3.6364	3.9333	3.1538	3.6995
14. Portability	2.8261	3.3000	2.8750	2.8000	3.6154	2.6000	3.2143	2.6154	2.9808
15. Reliability	4.0870	4.2500	4.2857	4.1818	4.1667	3.7778	3.4286	4.1538	4.0414
16. Reusability	2.8182	2.5455	3.0000	3.1818	3.3077	2.8000	3.3077	3.1538	3.0143
17. Robustness	3.8182	3.2000	3.1429	2.9000	3.0833	3.6667	3.5000	3.3846	3.3370
18. Scalability	3.0909	3.1000	3.2500	2.7778	3.0000	3.1250	3.1538	3.1667	3.0830
19. Security	4.4091	4.0000	3.8182	4.1667	4.0000	3.6154	4.0000	3.5000	3.9387
20. Supportability	4.0435	3.3571	3.8333	3.0000	3.4118	3.1429	3.2941	3.2000	3.4103
21. Testability	3.6818	3.2500	3.3000	3.4167	2.7500	3.3636	3.4706	2.2727	3.1882
22. Transferability	2.3043	2.6923	3.0000	2.9167	3.0000	2.9167	3.7333	2.4545	2.8772
23. Understandability	3.9500	3.7273	4.2222	3.8182	3.9333	3.2727	3.3750	3.4545	3.7192
24. Usability	3.9545	4.3333	4.2000	4.0909	4.2667	3.6364	3.7647	3.5000	3.9683
25. Visibility	3.6818	4.0000	4.4444	3.2727	3.8571	3.0000	3.4000	2.9167	3.5716

Source: *Developed by the author in [135] (Page 48) based on survey*

AVERAGE uses the following formula:

$$\boldsymbol{AVERAGE} = \frac{\boldsymbol{X_1 + X_2 + \cdots + X_n}}{\boldsymbol{n}} \qquad (4.1.1)$$

Where: AVERAGE = the mean, X_1 = the first value, X_2 = the second value, X_3 = the third value, X_n = the last value and *n* = the number of the values

The Standard Deviation (STDEV.S) is a measure of how widely values are dispersed from the average value (the mean), and uses the following formula:

$$\text{STDEV.S} = \sqrt{\left(\left(\sum (x - \bar{x})^{2}\right) / (n-1)\right)} \qquad (4.1.2)$$

Where: x is the sample mean AVERAGE (number1, number2, …) and n is the sample size. STDEV.S assumes that its arguments are a sample of the population. If the data represents the entire population, then compute the standard deviation using STDEV.P.

STDEV.P uses the following formula:

$$\text{STDEV.P} = \sqrt{\sum (x - \bar{x})^{2} / n} \qquad (4.1.3)$$

Where: ***x*** is the sample mean AVERAGE (number1, number2, …) and ***n*** is the sample size.

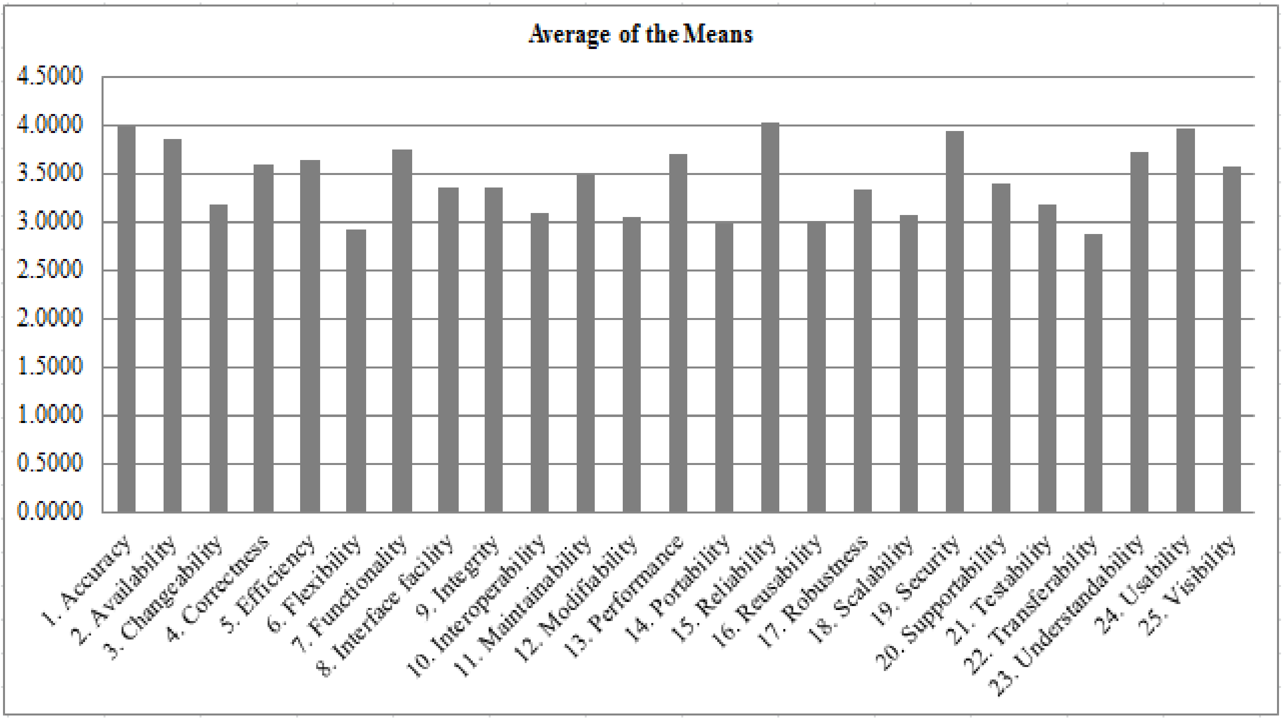

Figure 4.1. Value of the field research Average of the Mean Scores

Source: *Developed by the author based on the survey the results*

The graph *(Figure 4.2)*, represent the variation of quality characteristics by type of information system, according to the data from the *Table 4.1*.

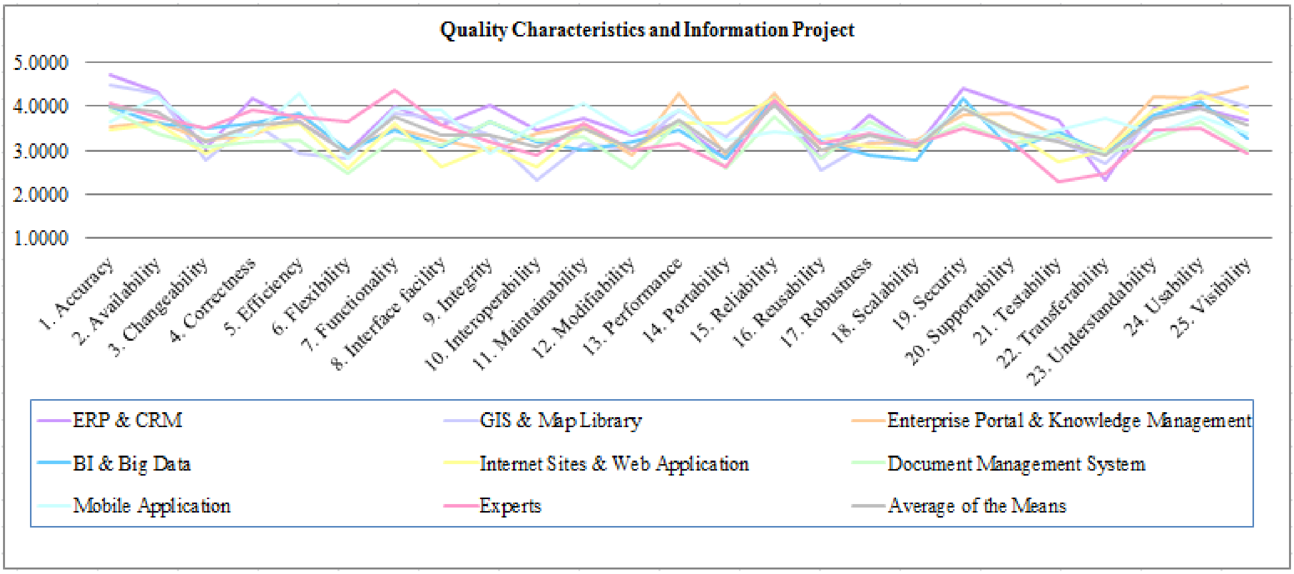

Figure 4.2. Variation of Quality Characteristics by type of IS

Source: *Developed by the author in [135] (Page 49) based on survey*

4.1.2. Data quality and Information quality dimensions

According to the ISO 25012 data model are defined using:

- 12Inherent Data Quality characteristics,
- 10 System Dependent Data Quality, of which 7 are bivalent (belong to both types of characteristics)

Another data quality classification is provided by Wand and Wang *[136]*. They limit their focus to intrinsic data qualities, of which they define four intrinsic dimensions: completeness, unambiguousness, meaningfulness and correctness. Wand and Wang take as their basis a paper, which features a review of cited data quality dimensions, i.e. the comprehensive literature review of Wang et al. *[137]*. Based on the comprehensive literature review Wand and Wang summarize the most often (Frequency) cited data quality dimensions as shown in *Table 4.2 [136]*.

Table 4.2

Cited data quality dimensions Source: Wand and Wang

Quality dimensions	Frequency	Quality dimensions	Frequency	Quality dimensions	Frequency
Accuracy	25	Format	4	Comparability	2
Reliability	22	Interpretability	4	Conciseness	2

Quality dimensions	Frequency	Quality dimensions	Frequency	Quality dimensions	Frequency
Timeliness	19	Content	3	Freedom from bias	2
Relevance	16	Efficiency	3	Informativeness	2
Completeness	15	Importance	3	Level of detail	2
Currency	9	Sufficiency	3	Quantitativeness	2
Consistency	8	Usableness	3	Scope	2
Flexibility	5	Usefulness	3	Understandability	2
Precision	5	Clarity	2		

Source: *Adapted by the author based on [136](Page92)*

The graph in *Figure 4.3***.** represent the Quality dimensions and the frequency- the most often cited data quality dimensions based on Wand and Wang *[136]* comprehensive literature review.

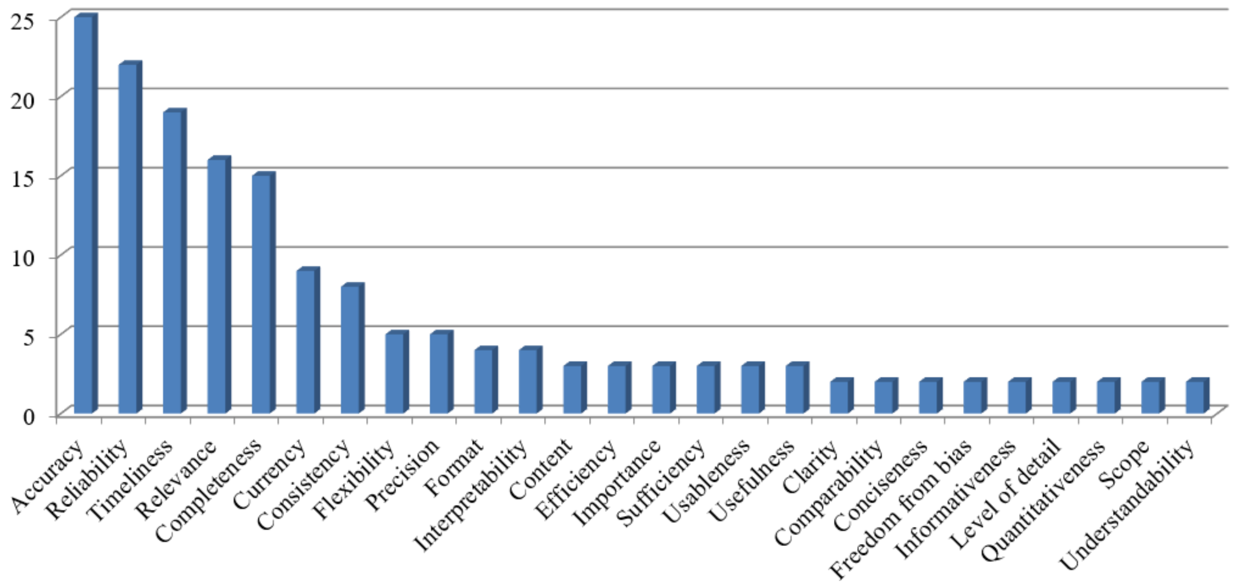

Figure 4.3. The most often cited data quality dimensions and their frequency

Source: *Developed by the author based on [136](Page 92), [138] (Presentation page 7)*

As mentioned, Wang and Strong *[139]* propose a DQ/IQ classification which divides data quality into four categories: intrinsic, contextual, representational, and accessibility. For each category, they define a set of dimensions. The definition by Wang and Strong is discussed by Haug et al *[140]*who argues that “representational data quality” can be perceived as a form of “accessibility data quality” instead of a category of its own. Thus, Haug et al. define three data quality categories: intrinsic, accessibility and usefulness. Levitin and Redman *[141]* provide another perspective by arguing that since processes to produce data have many similarities to processes that produce physical products, data producing processes could be viewed as producing data products for data consumers. With a basis in this view of data as resources, Levitin and Redman discuss how thirteen basic properties of organizational resources may be translated into properties for data *[142]*.

Information Quality is commonly thought of as a multi-dimensional (multivalent) concept with varying attributed characteristics depending on a quality view-point. Each organization or the information consumer (the customer or the user) has a different view of the dimensions of information quality. Determine information quality dimensions for the information quality can be used to add structure and instrumental to this inherent complexity. *Table 4.3* provides a summary of the most common dimensions and the frequency with which they are included in the comparison Information Quality Frameworks of Shirlee-and Knight and Janice Burn (2005) *[143]*.

Table 4.3

The Common Dimensions of IQ/DQ

Quality dimensions	Frequency	Quality dimensions	Frequency	Quality dimensions	Frequency
Accuracy	8	Understandability	5	Believability	3
Consistency	7	Accessibility	4	Navigation	3
Security	7	Availability	4	Reputation	3
Timeliness	7	Objectivity	4	Useful	3
Completeness	5	Relevancy	4	Efficiency	3
Concise	5	Usability	4	Value-Added	3
Reliability	5	Amount of data	3		

Source: *Adapted by the author based on [143] (Page 162).*

The graph in *Figure 4.4*, represents the common information quality dimensions and the frequency- number of time cited information quality dimensions based on Shirlee-ann Knight and Janice Burn *[143]* which they are included in the IQ/DQ framework table.

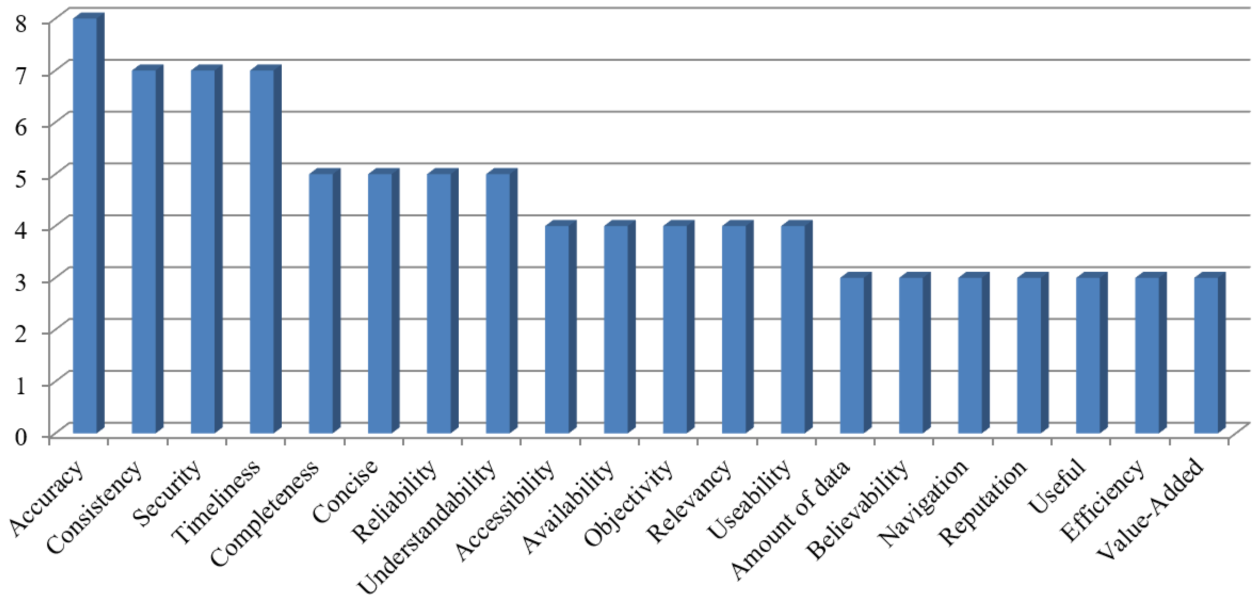

Figure 4.4. The common Information quality dimensions and their frequency

***Source:** Developed by the author based on [143](Page 162), [138](Presentation page 8)*

4.1.3. Critical analysis of methodologies of information quality assessment

More than two decades of research in the emerging field of IQ has developed useful theories, methodologies, and technologies for assessing, improving, and managing the quality of various types of information *[144]*. The concept of IQ goes beyond accuracy. It includes more than a dozen other dimensions such as timeliness, completeness, consistency, interpretability, accessibility, security, to name only a few *[139]*. *These different dimensions can be grouped into different categories*.

Several IQ frameworks have been developed to define and categorize various IQ dimensions *[139; 145-147]*. Among various IQ management methodologies, the Total Data Quality Management methodology *[148]* is one of the most used in researches and practice. It suggests that information should be treated as a product (Information Product) and managed continuously by following the cycles of improving Quality through: Define, Measure, Analyze, and Improve (DMAIC) *[149]*. Existing research has attempted to identify a full spectrum of IQ issues, *most users are only concerned with a very few IQ dimensions*.

In fact, research has shown that a user typically can only handle approximately seven concepts without being confused or to flooded with data *[150]*. Thus, it is not effective to present too many IQ dimensions when informing users or occupy their inputs about quality. Therefore, numerous machine-based IQ assessment methods have been developed. Depending on the type of the information (e.g., structured vs. unstructured, centrally produced vs. socially contributed, medical domain vs. IT domain), different sets of metrics are selected and automatically assessed using different input features. Functional dependency analysis *[151]* and statistical analysis *[152]* can be used to identify various quality problems in relational and other types of structured sources. Record linkage techniques *[153]* can be used to detect duplicates and inconsistencies. For textual data, various quality indicators can be used as a proxy for quality metrics. The indicators can be based on content (e.g., information-to-noise ratio), metadata (e.g., Web page's last update date), or other features (e.g., HTML syntactic correctness). Up to 26 such indicators have been used to assess the quality of online health information *[154]*. With the growth of social media such as Wikipedia and various discussion forums, there has been growing amount of research that focuses on assessing the quality of socially contributed contents. The algorithms are usually specific to a particular type of social media platform because they rely on certain features specific to the platform. Most machine-based methods are scalable and can produce IQ metadata useful for improving the effectiveness of Web search and information retrieval.

However, automatic algorithms can, at best, estimate the overall quality. They cannot reliably generate ratings along quality dimensions because the relationship between selected features and quality dimensions are usually unknown or unreliable. For example *[147]*, number of edits is mapped to authority and article length is mapped to completeness for Wikipedia articles. It is debatable whether such mappings make sense. Ratings along quality dimensions are necessary for explication purposes and for the effective use of information (e.g., making trade-offs between dimensions). Furthermore, *certain selected metrics may be irrelevant to users in their intended uses of the information*.

More importantly, *machine-based methods cannot capture users' perspectives about IQ*. User-based assessment relies on user inputs collected using questionnaire surveys, ratings, or freeform comments. A systematic survey instrument has been used in various organizations to assess IQ perceived by users of different roles in the information supply chain. The survey method requires significant user involvement and is often used to assess a collection of IPs as a whole, thus it is not scalable to obtain real-time IQ assessment at a fine-granularity. Minimalist approach to online voting (such as thumbs up/down and "has the article helped you") does not capture sufficient information for quality improvement purposes. Freeform feedback option is cumbersome and thus rarely used by users.

User-based methods can capture user's perspectives about IQ, but are not scalable. They also lack the necessary granularity and specificity in terms of the IPs (in the case of the survey method) and the IQ metadata (in the case of the simple voting method).

Furthermore, the lack of user incentives often results in scarcity of useful feedback and even leads to biased and malicious feedback. I. Costaș *[128],* Hongwei Zhu, Yinghua Ma, Guiyang Surealized *[155]* and more and more others researchers that these challenges require further research. „*The discussions and theory, built around the idea of an integration of knowledge management) and quality management in the framework of the same business management system, look logically rather correct, although without an empirical evidence for such a claim".*

IPs quality, software and hardware quality, quality of IS and resulting data etc. are very different from different cases, the operational environments are very dynamic and these facts require specific individual models for continuous improvement of quality. „*It is not enough to involve users at the beginning of IS project, at the stage of elaboration of new IS concept. It is impossible to define final requirements to the system quality. Thus, it is necessary to build a mechanism which ensures a permanent monitoring of user's satisfaction by information results provided by IS" [156].*

Systematic feedback from real customers allows early risk reduction**.** By delivering early and getting feedback, we reduce the risk of building the wrong product. Using Agile development methodology, we can get early and systematic feedback along the development lifecycle. By focusing on architectural risk in the early sprints, we reduce the risk that we won't have a solution that can be built in time.

The QM of information projects, such development of I&CT infrastructure, development of information systems and services is a relatively new field of research and needs new integration frames for ensuring IPs quality *(planning, management, control), including quality of IPs development processes, quality of resulting products and quality in use,* which is dependent of concrete organizational context.

4.1.4. Comparing and merging the two frameworks: data and information quality

Confusingly enough, quality dimensions are named and approached differently in different frameworks. From the two frameworks we represent in this paper, some of the quality dimensions are mentioned in both and others are cited only in one of them. The following graph in *Figure* ***4.5*** how's the comparison of quality dimensions frequencies between the two frameworks - Information and Data Quality frameworks.

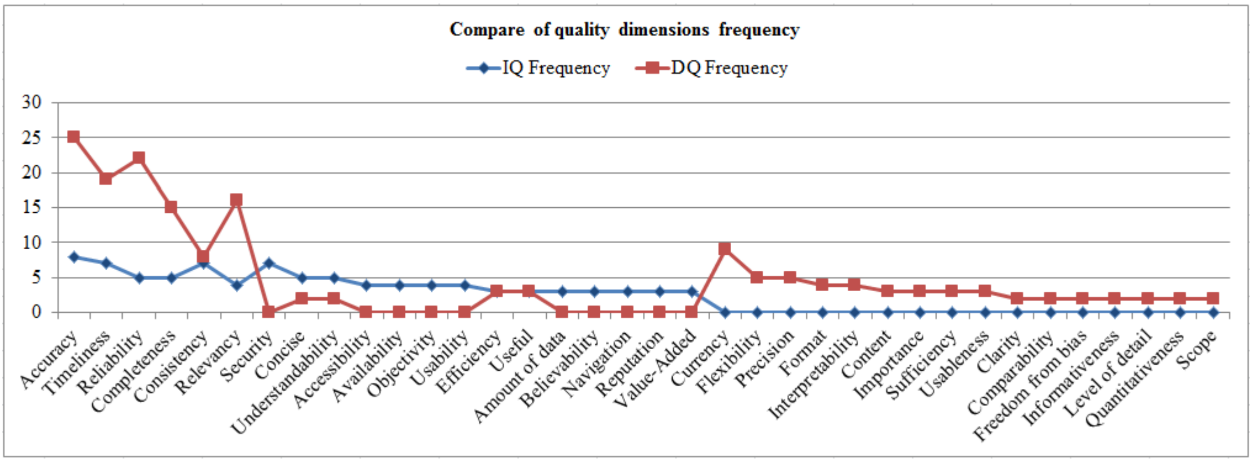

Figure 4.5. **Comparison of quality dimensions frequency from the two frameworks**

Source: *Developed by the author based on [136](Page 92) and [143](Page 162)*

In order to make the comparisons more effective and fairer, we need to construct a normalized dimension ranked for the two frameworks from 0 to 10 (The highest frequency number is given a score of 10 and all the other are measured relative to it).

Normalized Score (NS) uses the following formula:

$$NSX_1 = \left(\frac{X_1}{Z_x}\right) * 10,\ NSX_2 = \left(\frac{X_2}{Z_x}\right) * 10,\ \dots\ NSX_n = \left(\frac{X_n}{Z_x}\right) * 10 \qquad (4.1.4)$$

$$NSY_1 = \left(\frac{Y_1}{Z_y}\right) * 10,\ NSY_2 = \left(\frac{Y_2}{Z_y}\right) * 10,\ \dots\ NSY_n = \left(\frac{Y_n}{Z_y}\right) * 10 \qquad (4.1.5)$$

Where: (NS) = Normalized Score, X_1 and$Y_1$are the first value, X_2 and$Y_2$are the second value, Z = the max value, $X_n$and Y_n= are the last value and *n* = the number of the values.

This means that the quality dimension occurrence with the greatest number gets the score of 10. Therefore, we can compare and display a comparison of normalized quality dimensions frequency score from the two frameworks in one scale, as shown in *Figure 4.6.*

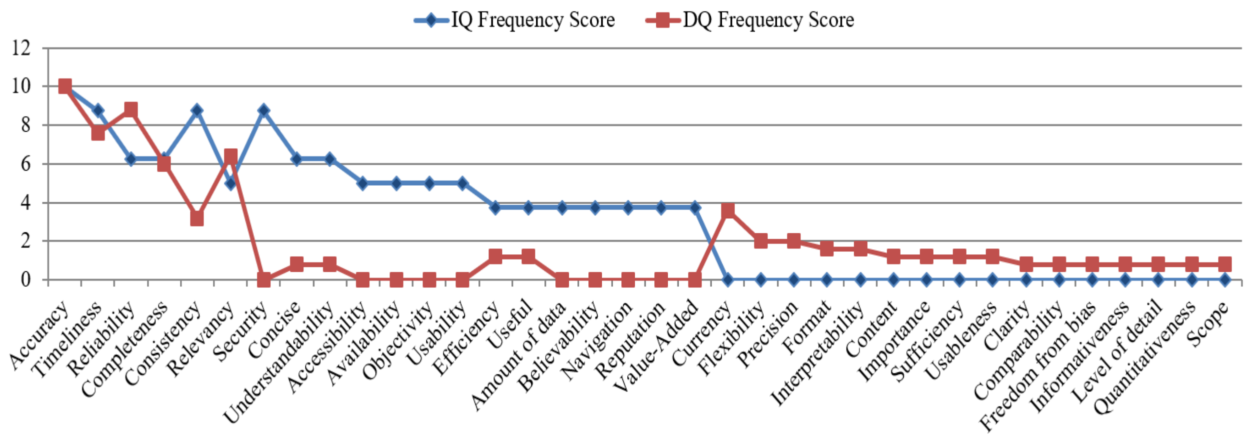

Figure 4.6. Comparison of normalized quality dimensions frequency score

Source: *Developed by the author based on [136] (Page 92) and [143] (Page 162)*

As we can see, some dimensions are common and used by the two frameworks. When we join the number of occurrences of each of the quality dimension from each framework, we can display a common graph in *Figure 4.7* of the total normalized number of citation score of each of the quality dimensions.

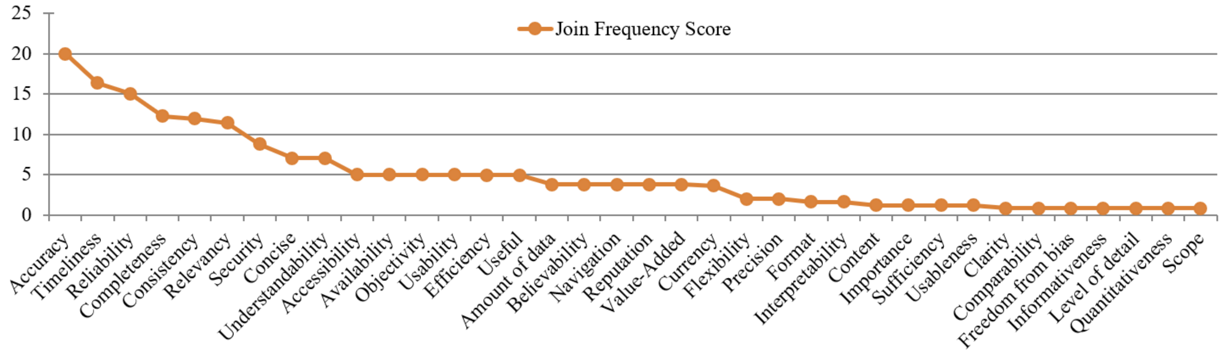

Figure 4.7. Join of normalized quality dimensions frequency score of the two frameworks

Source: *Developed by the author based on [136] (Page 92) and [143] (Page 162)*

In order to merge the two frameworks for assessing data and information quality, we propose a scale to determine the scores of the frequency weighted of each of the quality dimension measurements. For the scoring calculation, we need to sum the number of occurrences from the two frameworks (from *Table 4.2* and *Table 4.3*) and determine the highest score with the greatest value, namely, rank the dimensions. These ranks will be used as the basis for calculating the score scale between 0 and 10, with 10 - the highest score. In *Table 4.4* we can see the frequency score of the quality dimensions based on the two frameworks, so that made for each of the quality dimensions, the weighting of their grades according to the number of occurrences. This provides us a measurement scale for the quality dimensions with the frequency score.

Normalized Score (NS) uses the following formula:

$$NS_1 = \left(\frac{X_1+Y_1}{Z}\right) * 10,\ NS_2 = \left(\frac{X_2+Y_2}{Z}\right) * 10,\ \dots\ NS_n = \left(\frac{X_n+Y_n}{Z}\right) * 10 \quad (4.1.6)$$

Where: (NS) = Normalized Score, X_1 and Y_1 the sum of the first value, X_2 and Y_2 the sum of the second value, Z = the max value, X_n and Y_n = the sum of the last value and n = the number of the values. Alternatively, we can find the average of the Normalized Score of the two (or more) frameworks. Average Normalized Score (ANS) uses the following formula:

$$ANS_1 = \frac{X_1+Y_1+N_n}{n} \quad (4.1.7)$$

Where:

ANS = the average score for the characteristic, X_1 = the first normalized score of the first framework, Y_1 = the second normalized score of the second framework, X_n = the last normalized score of the last framework, n = the number of the values.

Table 4.4

Quality dimensions and the frequency score

Quality dimensions	Average Frequency score	Frequency score	Quality dimensions	Average Frequency score	Frequency score	Quality dimensions	Average Frequency score	Frequency score
Accuracy	10.0	10.0	Usability	2.50	1.21	Interpretability	0.80	1.21
Timeliness	8.18	7.88	Efficiency	2.48	1.82	Content	0.60	0.91
Reliability	7.53	8.18	Useful	2.48	1.82	Importance	0.60	0.91
Completeness	6.13	6.06	Amount of data	1.88	0.91	Sufficiency	0.60	0.91
Consistency	5.98	4.55	Believability	1.88	0.91	Useableness	0.60	0.91
Relevancy	5.70	6.06	Navigation	1.88	0.91	Clarity	0.40	0.61
Security	4.38	2.12	Reputation	1.88	0.91	Comparability	0.40	0.61
Concise	3.53	2.12	Value-Added	1.88	0.91	Freedom from bias	0.40	0.61
Understandability	3.53	2.12	Currency	1.80	2.73	Informativeness	0.40	0.61
Accessibility	2.50	1.21	Flexibility	1.00	1.52	Level of detail	0.40	0.61
Availability	2.50	1.21	Precision	1.00	1.52	Quantitativeness	0.40	0.61
Objectivity	2.50	1.21	Format	0.80	1.21	Scope	0.40	0.61

Source: *Developed by the author*

We can represent a graph in *Figure 4.8***.** that shows the quality dimensions with the average normalized frequency score *[157]*.

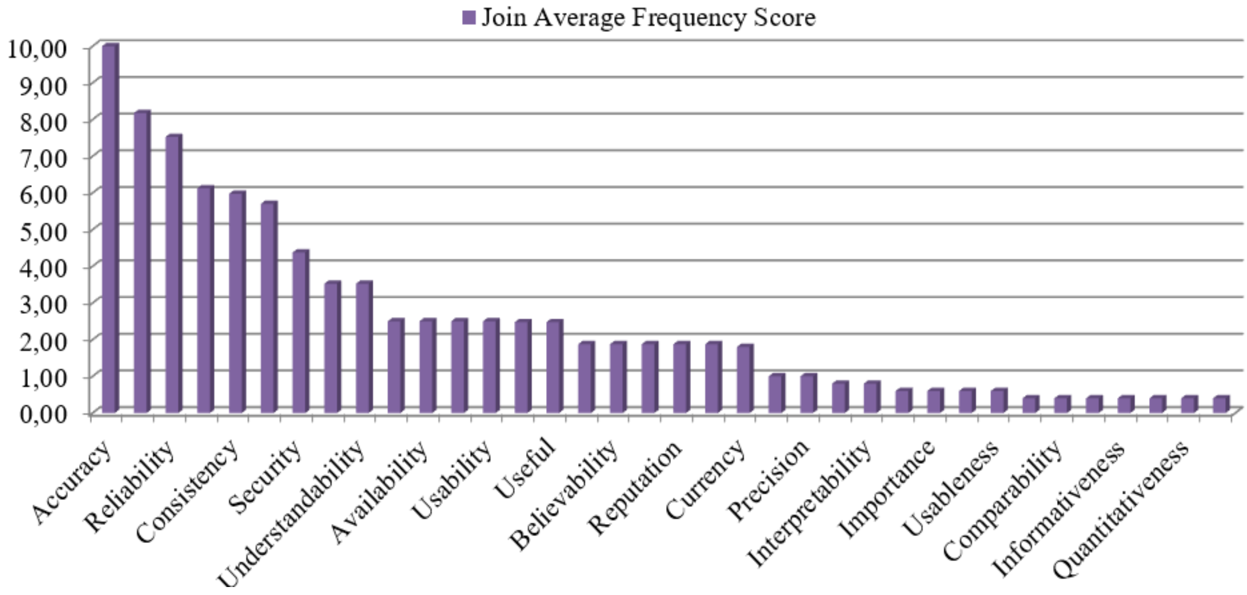

Figure 4.8. Quality dimensions average normalized frequency score

Source: *Developed by the author in [157](Page 93)*

In *Figure 4.9*, the graph represents the 15 quality dimensions from the two frameworks, which got the highest score.

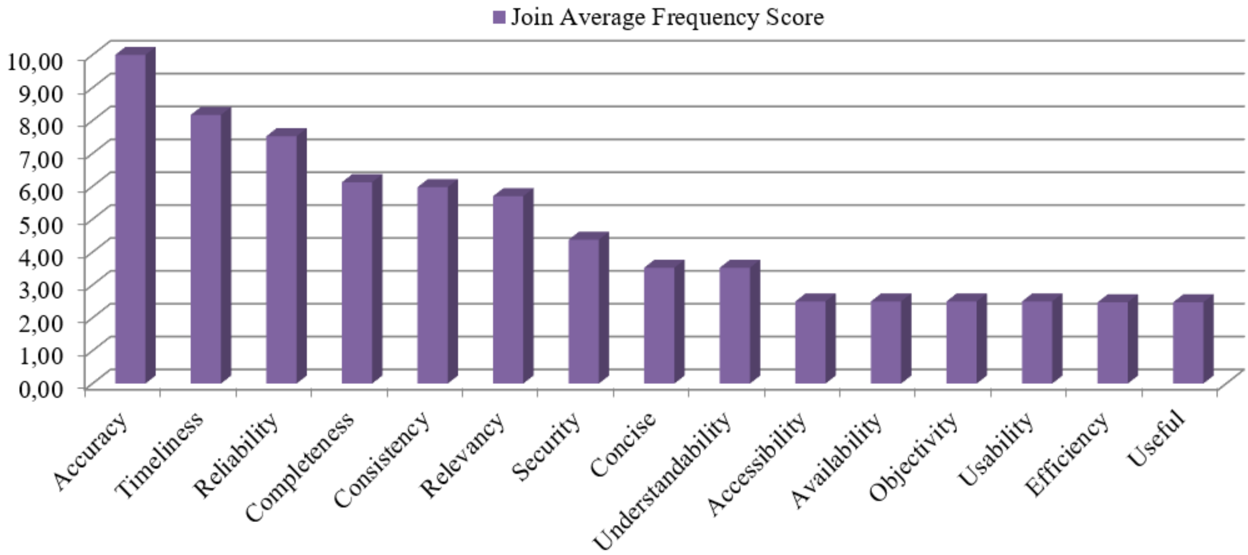

Figure 4.9. The 15-quality dimension normalized highest frequency score

Source: *Developed by the author in [135](Page 50)*

In *Figure 4.10*, the graph represents only the 10 quality dimensions which are common and shared in the two frameworks *[157]*.

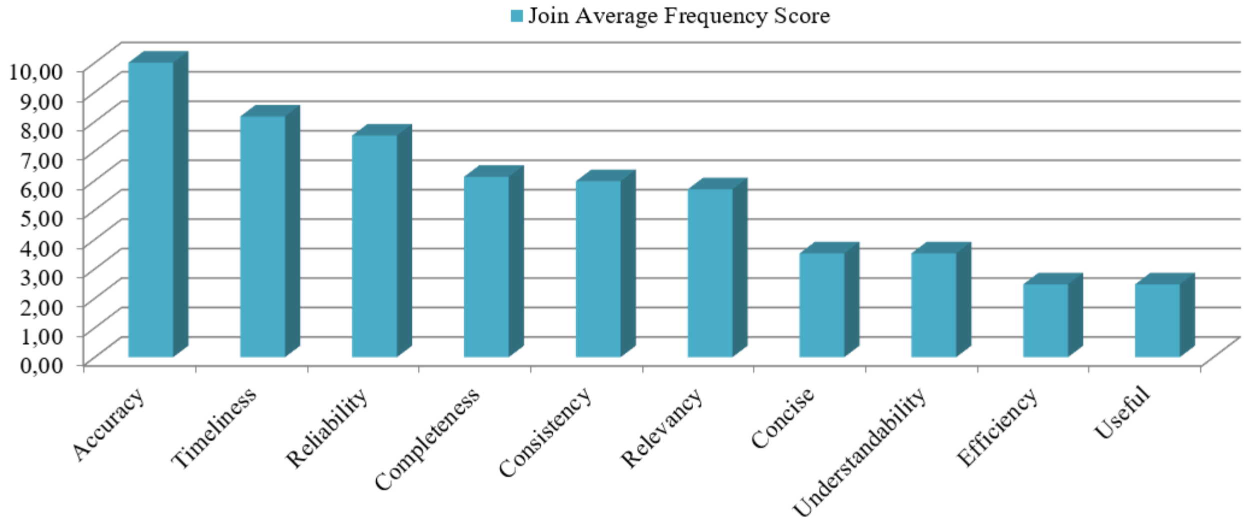

Figure 4.10. The 10 shared quality dimension normalized frequency score

Source: *Developed by the author in [157](Page 93)*

4.2. A model for improving information projects quality

Quality models are useful if they are understood by users, both in their constitution and intent, because:

- The intent of a measure gives hints on how to understand and interpret results. This somehow prevents the counter-use of metrics.
- Software quality concerns are at the very heart of what the developers, and the foundation, are doing. In some contexts (or projects) this may be a very delicate subject to deal with.
- Critics are very easy, especially in this domain: software engineering is not mature enough to have certitudes. As a consequence, a common agreement has to be found with users to really get the most out of such a work.

The fields for improvement:

- Defining practically usable quality models (quality attributes to be improved);
- Adapting quality models to specific organizations and application contexts;
- Integrating quality models into life-cycle development processes, for example Agile;
- Simple, transparent and unique definition form quality characteristics;
- Quantifying quality/measuring of primitive – input data directly from PMO, which support IPs development technological process.

4.2.1. Two level approach: metamodel and tailored quality model

In Chapter III, several quality models were presented, among which Call and Boehm early quality models to the ISO 9126, ISO SQuaRE (25010, 25012) models for system software and data quality, ISO 15504/SPICE or CMMI for quality processes. Since quality varies according to the domain *(safety, testability, usability etc. do not always have the same importance in each software product),* some quality models have been published for specific domains, like aeronautics, navigation etc.

In present section we define an explicit metamodel formally for quality models, which is more general and allows more degrees of freedom. This metamodel includes some basic models *(McCall [101], Boehm [102], ISO 9126 [108], ISO 25010 [33], ISO 25012 [158] models)* and some enriched models *(Dromey [106], FURPS [104])* with the possibility to generate other particular quality models based on the adaptation of one of them. The specific model is developed according to the metamodel and user requirements. Adaptation consist in eliminate some characteristics and/or add other characteristics, including individual characteristics, defined by the developing organization and stakeholders for concrete use-case. The resulting specific particular model is built under the subset of the most common quality characteristics and organization needs. In resulting model all relations between characteristics – subcharacteristics – metrics and the formula calculus are inherited from the metamodel, and have two variants of realization: simple media or multicriterial calculus.

This two-tier approach offers the possibility to compare similar products based on a particular manufacturer's model as well as different products and different manufacturers based on the basic features included in globally accepted and recommended ISO standards. The particular model allows systematic control and product quality monitoring over the entire lifecycle.

In favor of approach **metamodel↔ specific model** in the previous Chapters II-III, several arguments have been made, among which it is necessary to be mentioned:

- Software systems share some characteristics, but may really differ according to other, such application field, size and complexity of project, number and type of user, etc.
- Developing organizations also differ, ranging from size (small – to medium and/or large) and ending with specialization in certain fields, type of IS, culture, traditions, competitive environment etc. They may not be consistent in conventions, patterns used, process followed etc.
- For IPs/IS some characteristics, as maintainability, *error resistance simplicity in use*, etc. has a strong importance since the resulting product is intended to be used for

longtime, adapted or modified by anyone. So, some characteristics can be added in the model and others – can be deleted.

According to Software Quality Model Requirements for Software Quality Engineering, a quality model shall meet the following requirements *[159]*:

- It should be usable from top to bottom: users shall be able to understand how quality is decomposed down to metrics used.
- It should be usable from bottom to top: quality shall be assessed from the retrieved metrics up to the quality characteristics.
- It should include the different perspectives on quality, for example five points of view, defined by D. Garvin *[36]*.

Among the principle requirements for building a particular quality model we add:

- The quality measurement should be open and transparent, because on last resort quality comes from people themselves.
- The extraction of input data and analysis process should be as much as is possible automated (ideally fully).

4.2.2. Metamodel and particular/tailored models

The complete quality model is broken into three main parts:

1. The core quality model (metamodel), is compose of a series of quality characteristics attributes of included basic models and user added characteristics. All the attributes are described in these respective models and standards. Terminology and definition of quality characteristics are in according with ISO 25010:2011 *[33]*. All of the definitions of characteristics, subcharacteristics, metrics, and measurement function are including in the repository file of application *(fragment of Quality Knowledge Database is shown in Annex 1 and Annex 2)*. All of the formulas were taken from the respective standards: *ISO/IEC 25022:2016, Systems and software quality requirements and evaluation (SQuaRE), Measurement of quality in use; ISO/IEC 25023:2016, SQuaRE, Measurement of system and software product quality; ISO/IEC 25024:2015, SQuaRE, Measurement of data quality*. The templates of the quality metrics and the calculation functions, including the generalization of the results of the measurements in the values of the characteristics and their integration in a single value, are made in Excel, they are competently self-defined, explicit and full described according to the respective standards.

2. The quality *characteristics* and subcharacteristics are mapped to some type of IS, e.g. *web application* and applied standards from company, like ISO 9126, ISO 25010, etc.

3. Measurement concepts are in turn mapped to one or more metrics, e.g. „code size", which is measured through source lines of code, and „control-flow complexity", measured through cyclomatic complexity. List of metrics can be extended/ added to the user's need and desire. The metrics are defined in ISO 25022 *[60]*, ISO 25023 *[38]*, ISO/IEC 25024:2015 and other.

The whole model structure (i.e. links between quality characteristics, subcharacteristics and metrics) are defined in the „Models file repository". By default, this file contains basic models defined in ISO 25010:2011 (quality in use and quality of product) and the built tailored models for seven groups of IS.

The implicit model of project quality/project success includes six competing characteristics: *Duration, Cost, Scope, Quality, Risk and Customer satisfaction.* Each of these can be evaluated as 1 or 0: *Yes – the requirement constraint is satisfied, or No – isn't present, is not satisfied).* The simplified structure of the model of project success for six factors is presented in *Figure 4.11.*

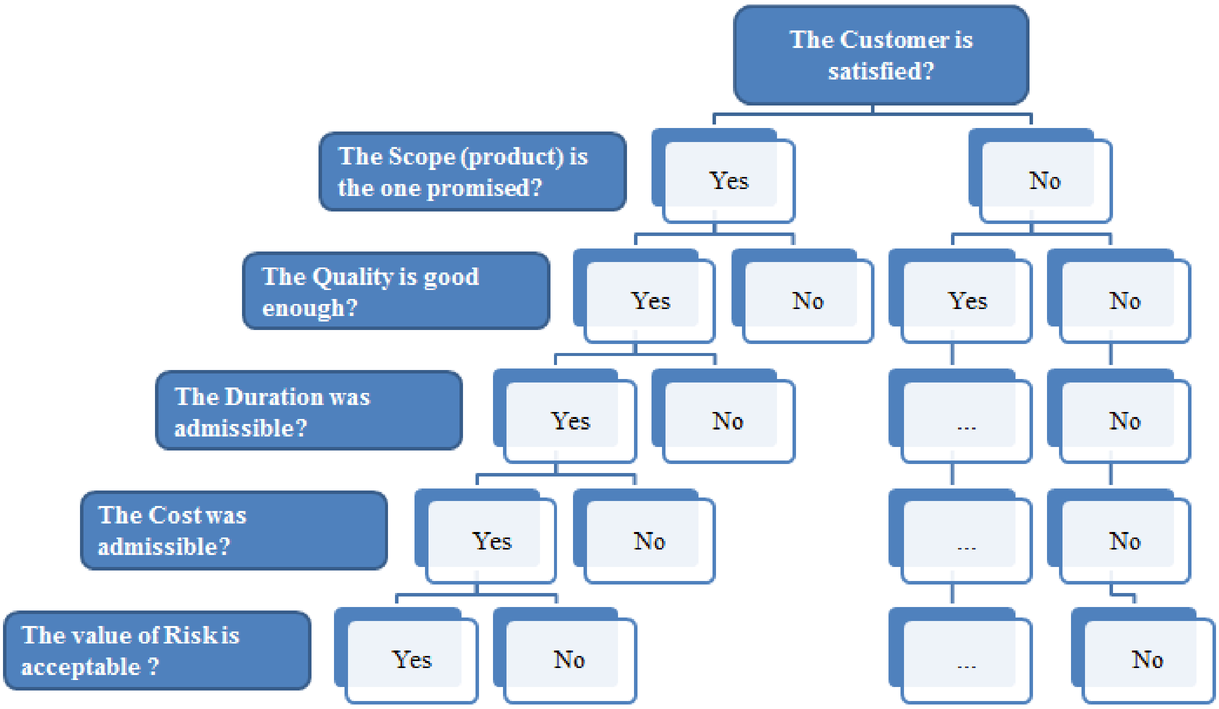

Figure 4.11. The simplified model of project success for six factors

Source: *Developed by the author*

The case when *all of the factors have the value* ***1*** $(111111)_2 = (64)_{10}$ is considered a total success, when the product, results, requirements and constraints are those expected, and the expectations of the stakeholders are met. The case when all of the factors have the value ***0*** $(000000)_2 = (0)_{10}$ is considered a total failure, when nor of the product, results, requirements and limitations are not these expected, and the stakeholder's expectations aren't fits.

Any intermediate value between 0 and 64 can be interpreted according to the mutual agreement between the beneficiary and the executor. The appreciation of the project is proportional to the obtained value, and it is desirable to maximize it. Apart from the interpretation of the zero/one values a weighted value can be evaluated. At the same time, it is worth mentioning that in reality the interpreted alternatives are much less, as rule, not more than 5-8.

The implicit models of the product and of data quality are those defined in standards ISO 25010:2011 and ISO 25012:2008. But all of these models can be adapted according to the user's requirements based on metamodel.

While the quality attributes and measurements concepts are not supposed to change (accepted for the evolutions of the measurement process), quality metrics may change according to some specific characteristics, like the programming language used and the availability of data.

The following picture *(Figure 4.12)* exemplifies a fragment of metamodel, from quality attributes to measurement concepts and metrics. Quality attributes are on the left (blue background), measurement concepts are in the middle (green), and metrics are on the right (orange).

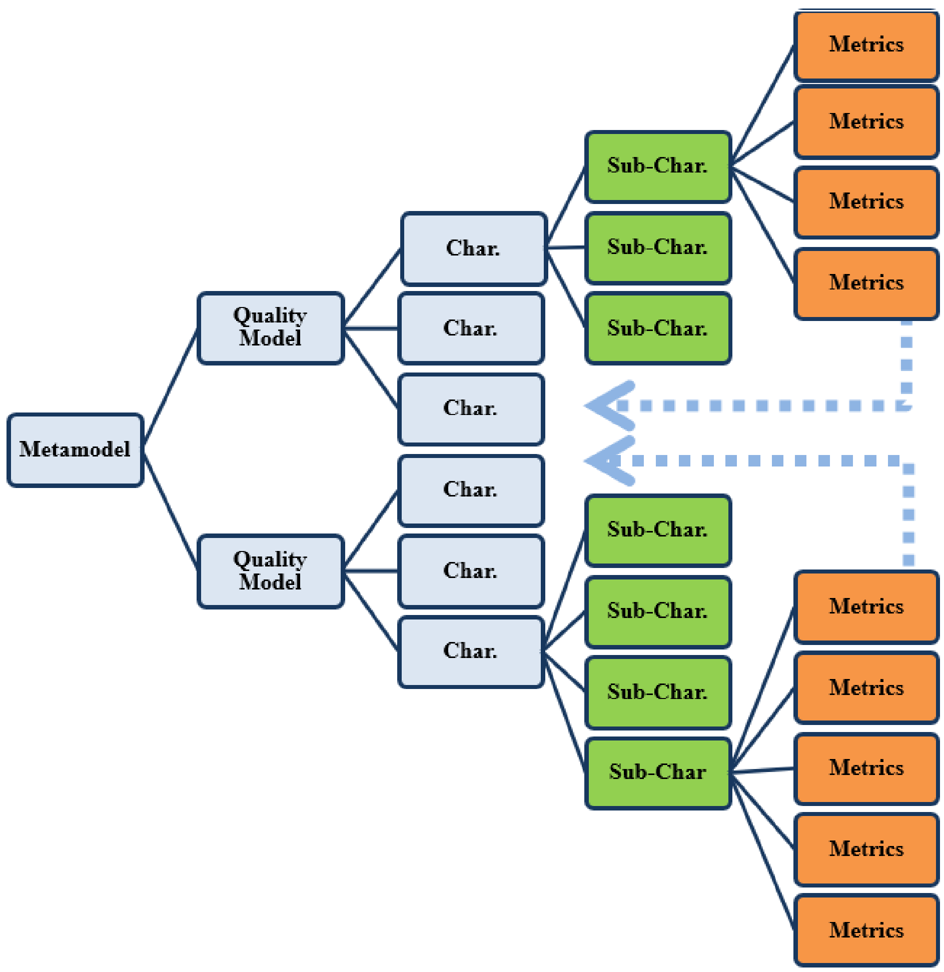

Figure 4.12. A fragment of the metamodel of quality

Source: *Developed by the author*

Metrics for quality in use and are defined in ISO 25022, and metrics for product quality are defined in ISO 250223. The relationships between models, characteristics, subcharacteristics and metrics were defined in database of applications (*Figure 4.13)*.

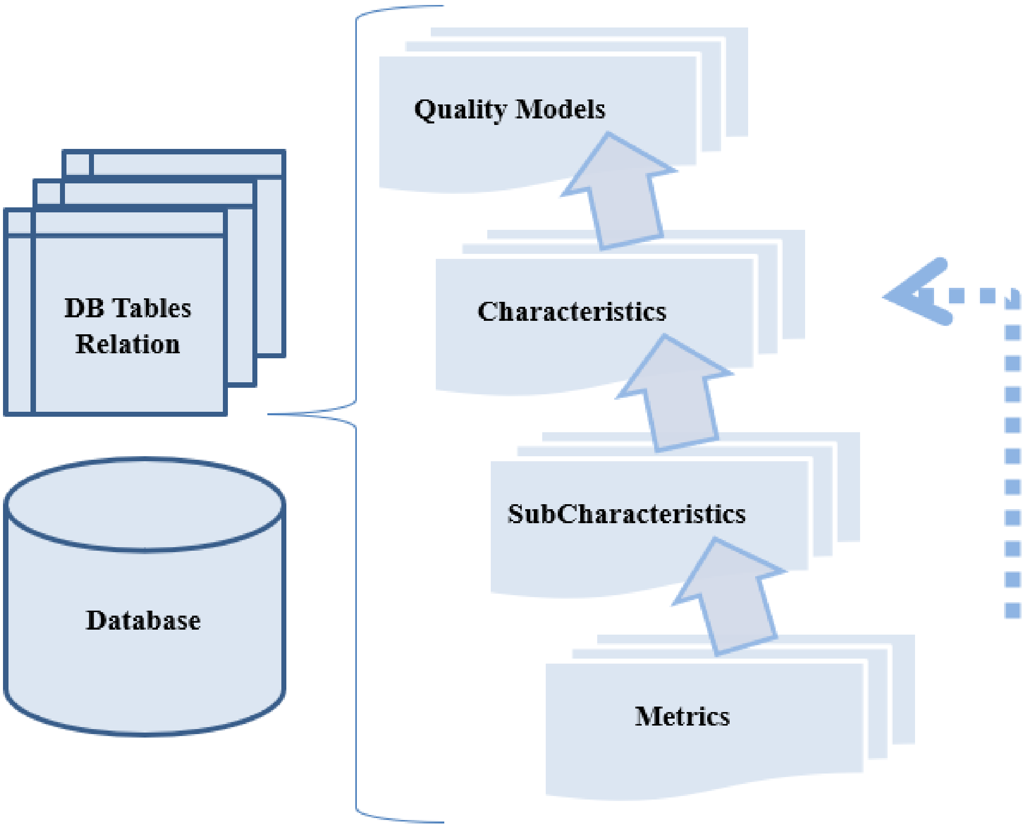

Figure 4.13. Relations between models, quality factors and metrics

Source: *Developed by the author*

4.2.3. Improving information quality

Organizations who realize that information is a part of the organizational quality process will get superiority over the competitors *[160]*. Therefore, organizations should determine who is responsible for quality improvement and quality assessment of the information. Moreover, it is important to determine the quality evaluation system, using machine-based methods and user-based assessment, to monitor and measure quality improvement over a time period and compare it to previous periods. To ensure information quality, organizations must comply in accordance with clearly defined quality dimensions, like quality control in the manufacture of other products, which are provided and are valued in accordance with the specific quality characteristics.

Information Quality literature has provided a great amount of proposals for assessing the quality of information, but there is still a need to develop frameworks for assessing and improving the quality of information from the information consumer and the organizational point of view in the perspective of the information project classification. Moreover, for each dimension there must be set a clear definition what it represents, in order to be able to compare it for any type of Information Project (i.e. information system) throughout its lifecycle. In this paper, we prefer to base on the Data quality dimensions scale proposed initially by Wand and Wang *[136]* and the Information quality dimensions proposed by Shirlee-ann Knight and Janice Burn *[143]*. Both are strong and well validated. There are some basic differences in the theories of both models since data and information are not the same (as we presented in the introduction). However, most of the dimensions used, and especially the most frequent ones, are very similar. In this work, we tried to combine the two models and built a shared set of dimensions. Hopefully, this will give a starting point for the further research.

In order to prepare an assessment scale and give the appropriate weight for each of the quality indicator for certain project information, we revealed that quantifying information quality involves two main stages: first, identifying which dimensions are important and relevant to the information project and second, determining how these dimensions affect the customers' needs when they consume the information. This will enable to perform more accurate assessment of the quality, identifying discrepancies, and determining the necessary actions for improvement.

4.2.4. Proposed scenario for improving information project quality

In order to improve the quality of information projects, it is required to present an information project quality improvement model that can be iterative measured and improved, during a period of time. Quality improvement is an iterative process that requires planning, execution, checks and feedback from the information consumers (IS users) in the organization. By this process, during a certain time frame, the quality of the project can be improved and reach higher business performances *[161], [162]* (presentation), *[163] (poster) [164]* (presentation).

Therefore, the quality activities are performed on a time scale, all along with the project's lifecycle, in order to achieve an effective improvement of the Project quality. During each activity, the quality measurements' results will be tested and further recommendations for improvements will be given, by feedback and controlling. This type of process will enable to compare, along with the time periods on the time scale, the quality improvement from iteration to iteration and produce a quality improvement scale for that period.

Since that information project owns unique characteristics and specific quality measurements, they are the core parameters of the proposed model. The main relevant parameters that take part in the model are:

1. Project classification, such information project type (CRM, ERP, BI, etc.), end-user quantity, information volume, budget, project content, etc.
2. Information quality dimension required according to the decisions making type and the information needs within the organization.

During the progress of the project, the quality activities will be carried out according to the quality measurement scale and the input parameters that were set in advance. The end user of the information system, will feedback the received quality and those will be are turn parameters used for quality improvement during the next interactions (namely, the next quality activities). As well as the user's feedback on the information system quality, this will be used as additional quality parameters for quality improvement in the model. Scenario for continuous improving information project quality along the lifecycle is shown in *Figure 4.14.*

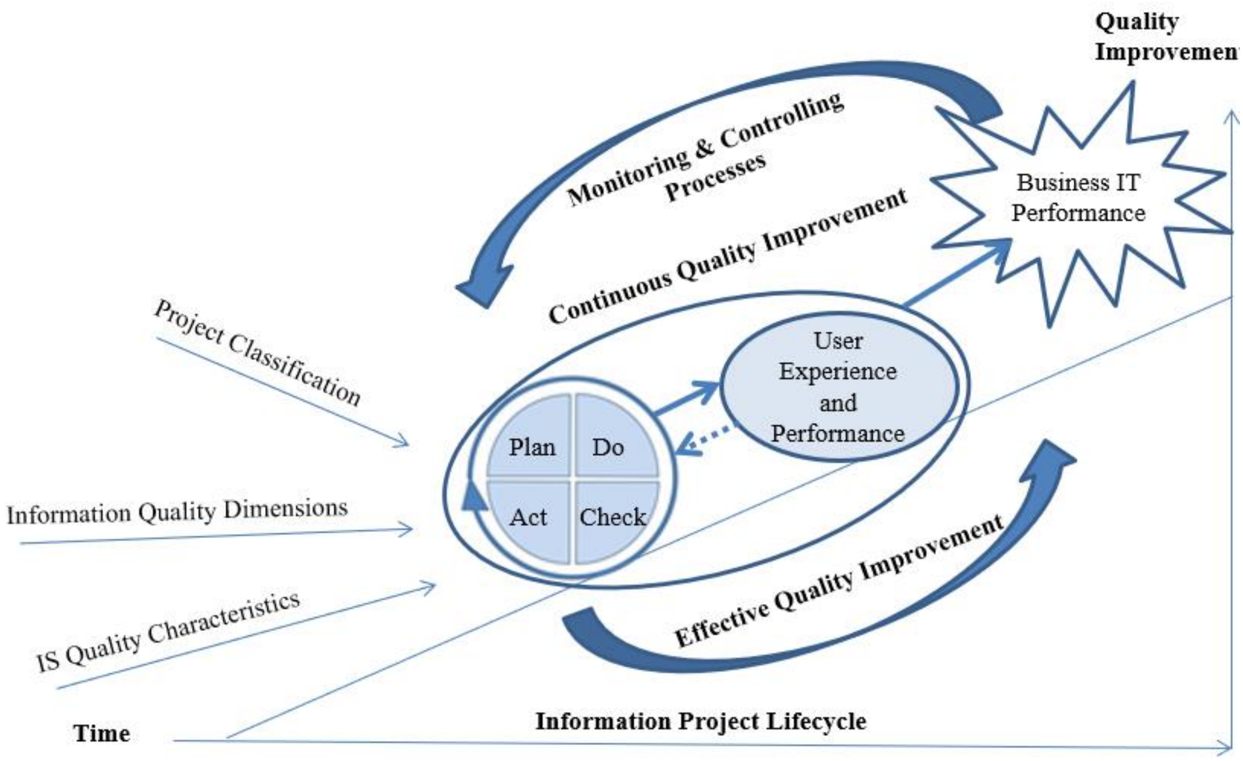

Figure 4.14. Scenario for continuous improving information project quality

Source: *Developed by the author in [135](Page 52)*

The basis assumption of the scenario is that by constant improvement, iteratively, it will be possible to increase the quality measurements and quality itself, required by the project stakeholders, which will result in increasing the performances of the system.

The described metamodel is a general model and does not represent the characteristics of a specific project. Therefore, in order to activate this model, it should be adjusted to the specific private case of each individual project.

Implementing an **Enterprise QMS** *(standards + procedures + templates)* in according ISO 9001, CMMI, AGILE is a major effort, which needs a *Policy, Engagements, Involve personnel, stakeholders*, etc. Project managers, sponsors, stakeholders and team members need to receive some level of training on how to play their roles. Organizations also face the problem of „cleaning out the pipeline" of in-process projects that have not been managed according to the new assessment of quality. Often times the best approach is to simply let these projects wash themselves through the pipeline, which means the presence of many projects with the need to increase their quality.

And other major implementation issue is the sophistication of the project management processes. The temptation is *to go too far and insist on more project management processes than necessary and many functions of product*. Principle KISS *(Keep it Simple, Stupid)* and YAGNI = *„You Aren't Going to Need It", which means "have the courage to say no, to prevent a functional overload"*. By this principle we are encouraged to implement only the elements, requested by the client, nothing more!

It is better to adopt a simple system that requires relatively little additional time from sponsors, project managers and team members.

Since a project is a private case of a certain information system foundation, time frame dependent, budget, contents and quality dependent in order to achieve a certain business improvement, it is required to determine, for each project, its own relevant characteristics, quality measurements and quality demands from the information system, according to its specific type.

In this research, quality and software characteristics were presented, that are relevant to a model, according to the project type. For the realization of the model, it is required to make certain adjustments according to the organization, the project and the information quality type needed. This is done *in order to fit the model into the private case of the project*.

4.3. Numerical methods for quality assessment

4.3.1. Linear calculation of quality factors values

As a rule, quality evaluation according to the quality models McCall, Boehm, FURPS, Dromey, ISO 9126, ISO 25010 etc. uses a set of values/metrics and linear expressions of calculation for each factor, such as:

$$\boldsymbol{Fq = \sum_{i=1}^{n} Ai * Mi} \qquad (4.3.1)$$

Where:

$\boldsymbol{Fq}$ = is the quality factor,

$\boldsymbol{Ai}$, i=1,...n are the coefficients of regression,

$\boldsymbol{Mi}$, i=1,... n are the metrics corresponding to the quality factor,

Then the general quality is determined as the weighted sum of the factor values.

$$\boldsymbol{Q} = \sum_{q=1}^{m} \boldsymbol{Fq} * \boldsymbol{Pq} \qquad (4.3.2)$$

Where:

$\boldsymbol{Fq}$, *q=1,... m*are the calculated values of the factors,

$\boldsymbol{Pq}$, *q=1, ... m* are the corresponding weights of the quality factors.

All metrics of the attribute are summed up and form a quality indicator. When all attributes are evaluated for each of the quality indicators, a total assessment of a separate indicator is made, and then an integral quality assessment taking into account the weighting factors of all software indicators.

This plot is only appropriate when the factors do not depend on each other, which does not correspond to reality. In the first chapter we discuss about interdependence between quality characteristics and subcharacteristics, no matter what perspective they are viewed or grouped, there are multiple relationships of interdependence, subordination, hierarchy, composition or aggregation and the impacts of quality subcharacteristics on characteristics are not equivalent and it is hard to determine. And we were talked that, *models must be made more meaningful for different cases by using coefficients which relate characteristics and subcharacteristics*.

So, in complex systems such IS with the use of several indicators/criteria there may be several criteria objectives/functions, often interdependent. Thus, during the creation, research, application and development of complex software systems, the assessment of the quality of the relevant processes becomes possible only with the use of several indicators (several target, criterion functions). **This leads to the emergence of multi-criteria choice tasks.** *Below is the specificity of the multicriteria choice and some numerical methods that can be used successfully to evaluate the quality of the processes/products.*

4.3.2. Multicriteria numerical methods to evaluate the quality

In order to evaluate the quality, it is necessary to identify which characteristics are most closely related to this software product. Therefore, the characteristics that must be taken into account should be selected based on the type of software product (e.g. embedded, real-time, etc.). In addition, the subcharacteristics (in the case of internal and external software products) and the measures also need to be identified. Under these conditions, the problem of a multi-criteria choice is reduced to finding a vector$\vec{x}$, such that:

$$f_1(\vec{x}) \to \underset{\vec{x}\in\Delta_{s\beta}}{extr}\ ; f_2(\vec{x}) \to \underset{\vec{x}\in\Delta_{s\beta}}{extr}\ ;...; f_m(\vec{x}) \to \underset{\vec{x}\in\Delta_{s\beta}}{extr} \qquad (4.3.3)$$

The condition for the existence of a solution (4.3.3) can be written as a condition for the coincidence of the solution of m-particular problems of searching for an extremum for each ***i****-th* quality indicator on the set $\Delta_{S\beta}$:

$$x_1^* = x_2^* = x_3^* = ... = x_m^* \qquad (4.3.4)$$

There:

$$\vec{x}_i^* = \arg \underset{\vec{x}\in\Delta_{s\beta}}{extr}\ f_i(\vec{x}), i = 1,...,m \qquad (4.3.5)$$

The fulfillment of condition from formula (4.3.4) is possible only in the case of the consistency of the particular indicators of the quality of the software development. However, as the information analysis in previous chapters, these characteristics can be contradictory and optimization of the software development parameters for each of them leads to alternative (non-coincident) solutions.

Thus, the formalization of problem such in formulas (4.3.3-4.3.4) is not correct within the framework of the axiomatics of the classical theory of extremal problems and for its solution it is necessary to define the problem by invoking additional qualitative and quantitative information about the properties of criterion functions, alternatives, principles of optimality, etc.

As a rule, the main source of additional information when searching for the best alternatives are experts, who know the given subject area well and the decision maker who has a specific goal in order to achieve which the problem under consideration is being solved. Obviously, sometimes additional information in such problems of multi-criteria choice can be obtained from other sources, for example, based on the analysis of the results of system modeling.

But there are some solutions for solving the condition from formula (4.3.3).

The Method *„Elimination Et Choix Traduisant La REalité”* (ELECTRE) *[165], [166]* was proposed by Bertrand Roy (France, 1967) and provides the possibility of arranging variants of quality criteria in descending order of preference criteria. Today is known five versions of ELECTRE, of I to IV and IS versions, used in the most diverse fields. But only a few criteria involved in 7 steps taken to solve the problem using ELECTRE method. The idea of using ELECTRE method for quality assessment consists in *elimination of the various alternatives/quality criteria that do not meet the concordance.*

The Combinex® method *[167], [166]*, aims to sum up the contributions of the various characteristics or criteria of appreciation of the variants to the merit or overall performance. These inputs are appreciated by the utility of the criteria seen from the point of view of the interests related to the realization and use of the analyzed object *(product, service, program)* by means of utility expression. According to the specialized literature, in a matrix table the options (variants) are switched on the lines, and the columns are affected by the characteristics.

The method is called „Combinex" not only because it sums up all the merits of different characteristics, but also because it is used in the design process, combining qualities and costs in a balanced way. It starts from the idea that the beneficiaries of products and services would want maximum qualities for each criterion what designers might do at excessively high prices, which usually do not agree with customers. On the contrary, low prices usually correspond to poor performance. Combinex therefore means „combination of expenses with quality".

A risk in engineering software requirements is to increase the level of a quality characteristic to the detriment of another at least or as important *(for example diminuend a cost/expenses often is in detriment of performance or quality of product).* Any software projects have been abandoned because they had a poor set of quality requirements, even though they had well-specified interface and functional requirements. Developing software with a level of quality that meets user expectations requires a balance between quality characteristics. However, as mentioned above, linear patterns do not always adequately reflect the relationships between the quality characteristics.

To achieve quality is possible only in the case of consistency of quality indicators. However, as shown by the analysis in sections 1 and 3, these indicators are often contradictory and optimization of parameters for each of them leads to inadequate solutions.

The ELECTRE and Combinex methods present serious inconveniences on the credibility of the conclusions reached. Indeed, the coefficients of importance of the criteria are intuitively fixed as the notes for each criterion in each variant. Consequently, all these methods may be subjected to subjective influences *(two distinct groups of experts can reach different values of the coefficients and grades that are given).*

ROMPEDET Method (ROmanian Model of PErformance DETermination), designed by I. Stăncioiu, *[166]* could partly solve this problem.

ROMPEDET allows to greatly eliminating the subjectivity of quality level appreciation. Thus, the Hi performance of a product variant $\boldsymbol{V_i}$ $(i = 1, 2,..., m)$ is obtained by adopting a variant $\boldsymbol{V_k}$ as the basis and by reference to its characteristics the values of the variants $\boldsymbol{V_i}$, taking into account the importance of each characteristic, according to the formula

$$H_i = a \prod_{j \in S_1} \left(\frac{X_{ij}}{Xkj} \right)^{\gamma_j} * \prod_{j \in S_2} \left(\frac{Xkj}{Xij} \right)^{\gamma_j}, \qquad (4.3.6)$$

Where:

a – is a scale factor (is proposed 1000 for a more conclusive differentiation of the variants);

S_1 – the subset of characteristics, that are desirable to have the highest values for performance to be as high as possible *(e.g. usability)*;

S_2 – the complementary subset of characteristics, that are desirable to be as small as the performance to be as high as possible *(e.g. cost)*;

X_{ij}– the values of the technical characteristics of the variant ***V_i*** *($i \in m$);*

X_{kj} – idem for the variant ***V_k*** *($k \in m$);*

If in (4.11) ***i = K***, will result ***$H_k = a$***, so the performance of the given reference product equal to the scale factor ***a***.

The weight occupied by the ***j*** characteristic in defining the level of ***H_i*** performance is normalized as follows:

$$0 \le \gamma_j \le 1; \quad \sum \gamma_j = 1; \quad \gamma_j = \left| \frac{\partial C}{\partial Xj} \right| \frac{dXj}{dC}; \qquad (4.3.7)$$

Where:

C = f (X) represents the function of the expenses for ensuring the characteristics, including the exploitation, and ***X***– the set of characteristics ***X_j*** *(j = 1, 2,... n).*

If there is a lack of information about expenditures or their reporting on the qualitative characteristics of the IS, the weights ***γ_j*** can be established with the formula:

$$\gamma_j = \gamma_{j1} = \frac{\sum_{j_1} aj_1 j_2}{\sum_{j_2} \sum_{j_1} aj_1 j_2}; \; 0 \le \gamma_j \le 1; \; \sum \gamma_j = 1; \; j_1 j_2 = 1,2,\ldots,n; \qquad (4.3.8)$$

a_{j1j2} represent the elements of the square matrix ***A***nxn = ||aj1j2||, having the values:

$$a_{ij} = \begin{cases} 1,\ C_{j1} I C_{j2}, \text{ if characteristic } C_{j1} \text{ is of equal importance with } C_{j2}, \\ 2,\ C_{j1} P C_{j2}, \text{ if characteristic } C_{j1} \text{ is more preferred than } C_{j2}, \\ 4,\ C_{j1} PP C_{j2}, \text{ if characteristic } C_{j1} \text{ is much more important than } C_{j2} \\ 0, \text{ in rest of cases} \end{cases} \qquad (4.3.9)$$

Where:

I – is a logical operator of indifference (characteristic C_{j1} is of equal importance with C_{j2}),

P – is a logical operator of preference (C_{j1} is more preferred than C_{j2}),

PP – is a logical operator of preference (C_{j1} is much more important than C_{j2}).

The ROMPEDET method is essentially similar to the selection and hierarchy methods of the above-described variants in that they call these methods as two elements that define the overall quality *(overall performance)* of a product: the notes or utilities of each feature in each variation of the formula and the coefficient of importance of each characteristic *(weights* γ_j*)*.

However, there are fundamental differences in determining the usefulness of the features. Thus, the „notes" will result from the reporting of characteristics to those of the reference variation. The linearity that occurs in the definition of „notes" by this process is acceptable, because for most products and technical characteristics, the differences in the values of the characteristics have low rates, in those portions being allowed the approximation of the curves with the straight lines.

4.4. The summary description of applications tool.

4.4.1. Requirements for the application

General requirements. To estimate the values of quality indicators, four methods can be used: measurement, registration, calculation and expert (as well as combinations of these methods).

The measuring method is based on the use of measuring and special software for obtaining information about software characteristics, for example, determining the volume, the number of lines of code, operators, the number of branches in the program, the number of entry (exit) points, reactivity, etc.

The registration method is used when calculating the time, number of failures or failures, the beginning and end of the software operation in the course of its execution.

The calculation method is based on statistical data collected during testing, operation and maintenance of software. The estimated methods are estimated indicators of reliability, accuracy, stability, reactivity, etc.

The expert method is carried out by a group of experts (users-specialists) who are competent in solving this problem or the type of software. Their assessment is based on experience and intuition, and not on the direct results of calculations or experiments. This method is carried out by viewing programs, codes, accompanying documents and contributes to the qualitative assessment of the created product. For this purpose, controlled signs are established that are correlated with one or several quality indicators and are included in the expert questionnaire cards.

The method is used in assessing indicators such as analyzability, documentation, software structuring, etc.

The primary goal of application IPMS (Information Project Management System) according to the thesis results is to manage quality of project along lifecycle, assess and visualize measurement values and trends. Often, they also give a high-level overview of the quality of the analyzed system. Thus, they incorporate aggregation of measurement values. Since the first version use linear functions to compute the quality according to the hierarchical quality models as foundation for an aggregating, the application tool is considering the weights of the characteristics for each quality model.

The software application IPMS must be simple, transparent and with intuitive interface in English language, but all the knowledge and database objects (terminology, definitions, explanations, measurement functions) must be possible to be displayed/edited including in Romanian and Russian languages.

Taking into account the particular character of quality models for different organizations and types of projects, the application must be universal, suitable for any tailored quality models.

Taking into account different evaluating methods and measurement functions for each quality characteristic, the application must be independent of assessment method:

1. Objective evaluation. In this case, the value of each quality factor is calculated based on specific measurement function and objective initial data. Inputs/measurements can be extracted directly from PMO or can be manually introduced in Excel template. Excel template realize all of the calculus in according of the measurement functions and return the value of quality factors. All templates according *4.2.2. Metamodel and particular/tailored models* are actually usage examples and are recorded on the CD attached to the thesis. All three templates *(ISO/IEC 25010:2011 Quality in use_Template_v02.xlsx, ISO 25010 Quality_of_Product_ Template_v02.xlsx, ISO/IEC 25012:2008 Quality_of_Data Template v01.xlsx)* are actually usage examples and are recorded on the CD attached to the thesis, Some fragments of Quality Knowledge Database was sown in Annex 1 and Annex 2.

2. Subjective evaluation. The values of quality factors are obtained as feedback from users and/or from expert subjective assessment. In this case value of each quality facto can be manually introduced directly in application

3. Combined evaluation method. Some of the values of quality factors are obtained by objective evaluation, some – by subjective evaluation.

All of the measurement functions are described in the Knowledge database of the metamodel and are programmed in the Excel template in according with their ISO definitions.

Functional requirements. The application must allow the introduction and updating of the database about the *organizations, projects, assessments, quality metamodel, tailored quality models, quality characteristics, quality subcharacteristics and quality metrics*.

The quality metamodel contains all of the definitions of *quality characteristics, quality subcharacteristics and quality measurements* (calculation formulas) and *evaluation methods.*

The build of private model is done by simply selecting of the quality factors.

Each new assessment is saved in the database, after which the results can be displayed in the desired form. It is required to display the dynamics of several consecutive evaluations of project (sprints).

Technical requirements. This program will run on Windows Operating systems, starting from windows 7 up till windows 10.

Computers may not run this program unless they have dot net 4.6 installed.

A RAM of 4 GB or above will be enough to run the program.

A Processor of 2.0 GHz will run the program without lags.

Microsoft SQL Server 2012 or newer is required to use this program across many computers in Local Area Network (LAN). This option allows all computers on the LAN to use a central database. Other details can be found in the application description (*Annex 9*).

4.4.2. Description of the application

IPMS is a desktop application which enable organizations to manage quality of information projects, by performing quality assessments, in accordance with the tailored models for each type of information project, obtained from metamodel *(knowledge about quality models, quality characteristics, subcharacteristics and metrics)* and using a linear calculation of quality factors values *[162] (presentation).* The application is independent of the tailored model applied; it is suitable for any type of organization and/or any type of project.

The main menu displays some options for initial introducing/editing of the data, such *Organizations, Assessments, Tables settings of the application*, which define for each information project type a list of quality characteristics with a quality factor according to the thesis results *databases tables', Quality concepts of the metamodel and tailored models*, a *List of existing information projects,* to choose and manage and to display the *Information project quality assessment reports, graphs, statistics* etc. *(Annex9).*

Using the data input screens, it is possible to create a new organization, new project, new assessment and/or update any database objects, including metadata about *quality models, quality characteristics, quality subcharacteristics, quality metrics etc.* Metamodel can be extended/

adapted from organization needs*: user can add some characteristics, subcharacteristics and metrics, can modify some measurement functions in the knowledge database and/or in the Excel templates for collecting of primary data and calculation of value of quality factors*.

In addition, it is possible to create or maintain quality assessments or quality activities.

Each information project relates to quality characteristics in accordance with the project type, in order to perform the relevant quality assessments. Each project assigned to a specific organization in order to manage the projects of the organization.

The application has a built-in help with detailed description of operation. *Install kit and applications scenario, help text and others print-screens of the IPMS are shown in Annex 9 and, along with the entry templates, are written on the CD.* The application is installed on Microsoft Azure cloud server and can be verified from the Web, without installing it on the desktop. To access IPMS application, send an e-mail to the author's address to receive information regarding the updated Login and Password.

Conclusions on chapter IV

The misconception that ISO standards or quality models are enough to evaluate the systems/software or informational projects quality is fully supported by the field research which testify, that *each area in which was conducted the test that has different quality characteristics.* As a result, *the general, abstract quality assessment models have to be adapted for each type of project separately*: and we need to adjust the quality characteristics according the project classification, user quality requirements, company quality policy, etc.

So, neither ISO standards are performance standards. They do not address quality itself; they address only the management processes necessary to achieve quality. Standards say what needs to be done, but they do not say how to do it for concrete case of company, project etc. Because quality is strongly dependent of context – quality management system must be „born", „grow" and „mature" inside the company through tailored models, build in according with organization culture, strategy, needs, type of project etc.

The developed generic metamodel of quality and the corresponding application, incorporates all the known quality factors (characteristics, subcharacteristics and metrics) of the basic ISO quality models, which today reflect the best practices and the best international experience in the domain of software systems quality. In fact, the duration of all project activities, cost, project quality, project risks etc. are probabilistic, not deterministic. These cannot be perfectly anticipated, cannot be determined accurately, as the project is nearing completion. So, we must accept the variability of the parameters of a project quality, cost, risks etc. This is inherent in any process. Over time, as the project nears completion, the variability can be reduced, but it can never be eliminated. And the main purpose of a project manager is trying to reduce of risks to which a project is subjected or to minimize their effects to obtain required quality in accordance with the preset constraints. This it is possible using developed application along development lifecycle.

Application can be easily customized-adapted to the needs of a particular company, its own quality policy, quality culture etc. The metamodel are flexible and extensible. The definition of quality factors can be adapted to the concrete organizations and project context. The high adaptability and extensibility of the application in terms of defining/refining the metamodel and/or tailored models allows its use for other types of projects and organizations. The elaborated model of project success (project quality) respect requirements for quality models, as it is simple, transparent, and it is easy to extend, to calculate, to understand, to interpret.

The proposed approach and realized application open up the possibility to define the quality of IPs at the conceptual level, creating the basis for the subsequent formal assessment of the degree of compliance of the developed IPs with the quality requirements. The current version of application uses linear calculation of quality factor values.

GENERAL CONCLUSIONS AND RECOMMENDATIONS

The important scientific solved problem in the research is *a new approach for continuous assessment and improvement of the quality of information projects through the lifecycle*. This approach opens up the possibility *to define the quality of IPs at the conceptual level*, creating the basis for the subsequent formal assessment of the degree of compliance of the developed IPs with the quality requirements.

The new approach is composed by following obtained results: (1) generic metamodel of quality, which includes the best knowledge about quality factors, extracted from known basic models and quality standards; (2) tailored quality models built from metamodel, based on field research, which permit providing quality of some type of IPs; and (3) on an original support application; with the extraction of the some initial data directly from the Agile PMO.

The results were published in 10 scientific papers, among which 8 by a single author, with a total volume over 4 sheets of author, including 2 in magazines recognized abroad, 2 in journals category B, were reported in 4 international and 2 home conferences. On page 189 are listed all own publications, which are referred in the thesis in about 20 references.

The main research result is core/generic quality metamodel, adaptable, flexible and extensible, which contain the quality characteristics of included basic models *(McCall, ISO 9126, ISO 25010 etc.)* and user defined characteristics, what absorbs the best of the moment from the ISO 9001 standards and the ISO 25000 family, which meet current trends in quality management of software. Obtained metamodel is based on an extended literature review, qualitative and quantitative analysis methods, detailed survey questionnaire for different kind of people and different type of IS.

Some of these ideas can be seen in: [44], pages 59-60, 125-130; [100], pages 125-130; [135], pages 45,47-50, 52; [157], pages 89-93.

A second obtained result– was established the tailored quality models for some of the most important types of IS and values of corresponding weights of these quality factors, which allows a more accurate quality assessment for these type of IS. The research area was mainly oriented for information systems, such ERP, CRM, BI, etc. Weights were determined based on experts' answers. The most important result of this experts' survey says that there is a variance in scaling the characteristics of quality, depending of the type of project, the fitting quality scale according to information needs of the customers both inside the organization and out of it. A well-established QMS quickly develops business and offers advantages such as: simplifying and optimizing processes, increasing customer satisfaction, motivating employees, lowering costs and increasing productivity, creating or adjusting specific quality tools, quality standards and others.

The different IS use different quality characteristics with different weights. One can see about their choice and justification in: [135], pages 45,47-50, 52; [157], pages 90, 89-93; also was presented on MITRE 2015 and MITRE 2016 international conferences [138] (presentation), [161].

The third obtained result is universal application for continuous quality assessment. The quality activities should be carried out in an iterative way of measurement and improvement. This fact requires a quality model that, on the one hand, gets the suitable parameters that fit an information project and, on the other hand, allows measuring, in a unified scale, the quality improvement results, along with the project lifecycle.

Proposed scenario for improving information project quality and functionalities of the IPMS application were presented at 2 local (Israel) conferences [163] (poster), [164] and at 3 international conferences (MITRE 2015, MITRE 2016, MITRE 2029 [138 (presentation), [161], [162] (presentation)).

In a proposed application for quality assessment was used a tailored model, which is flexible, adaptable and extensible. End user can select the basic quality model depending of type of applications and then adapt this according to user requirements, define and add some new characteristics and measurement functions. Universality of application was achieved by the separation of the initial input data from the application and from measurement, registration, calculation and expert methods. All of these are programmed in the Excel templates, and permit to determine the values of the quality factors, defined in according of the users' needs. The Excel files also serves as a data-collector, extracted from various PMO instruments, used along lifecycle of development of IPs.

Considering the volume of routine operations, software evaluation and management can be effective only under automation conditions. The proposed application can extract many of the input data directly from collaborative PMO instruments in the Excel files/templates, which serves as a data-collector. So, the application is promising because it saves costs, avoids many routine operations by directly importing of input data, excludes data mismatch, etc. Adaptability and

The scientific novelty and originality are reflected in a new approach for continuous assessment and improvement of IPs quality along lifecycle based on combination between modern Agile development methodology and tailored quality models, obtained from generic quality knowledge metamodel, which is extensible, flexible and adaptable and which is supported by software application with primary data extraction directly from the PMO tools.

The presented approach is new, even if we were using the classical models of software engineering and standardized best practices. The modification we respect is integration of existing quality models and best practices with user-oriented quality conception, in according with users' needs, users' requirements, using the new more suitable quality assessments models at conceptual level, mix of traditional evaluation methods and modern development methodology, such Agile.
This approach permits to build the tailored quality models, more suitable to the concrete organization and project contexts; fits the quality characteristics, metrics with users' needs; realize a combination of quality models with modern development methodology Agile and tools for assessment and improvement of quality. That's mean the quality of IPs can be carried out in an iterative way of measurement and improvement along the lifecycle.

The theoretical significance is supported by analyze, synthesis, specifying and defining the theoretical principles, generic metamodel and tailored models for personification of the quality, continuous assessment process of the quality of IPs through the project lifecycle, based on connection between several well-known basic models, tailored models, Deming quality wheel, Plan-Do-Check-Act cycle, Agile development methodology and PMO tools for quality assessment.

The applicative value of the research. The new assessment approach, generic metamodel, tailored models, measurement functions – all of these have been realized in a software universal application tool IPMS for support of mangers. The research outcomes can be directly used for 12 type of IS, such ERP & CRM, GIS & Map Library, Enterprise Portal & Knowledge Management, Business Intelligence & Big Data, Internet Site & Web Application, Document Management System and Mobile Application. The new assessment approach and the realized software tool are implemented in „WGS", Israel *(Annex 5)* and in the study process of the Moldova State University *(Annex 6)*. But the results could be used by researchers and students in software engineering disciplines, could be easily implemented in any organizations that use IPs, in according to the specific criteria of that project. The applications can save the inputs and outputs of quality assessment results to the database, in order to compare between the quality activities over time.

Recommendations and suggestions for future research. The present research can be expandable. Building a framework for measuring, assessing and improving quality requires both methodological support and technology support with the right tools. Also, developing/implementing this tool as a software application designed to support quality management, *more theoretical and empirical research, including expert opinion surveys are required.*

In particular, research could be continued in several directions, but not only:

(1) Quality modeling along the lifecycle, investigation of characteristics and related metrics, measurement functions to determine correlation, significance, the degree of overlap, dependencies and degree of automation is the most important way to increase the quality, inclusively using a formal, mathematical modeling, such set theory, graph theory, etc.

(2) Refinement of many quality factors, metrics that, in aggregate, adequately reflect the quality of software along the lifecycle.

(3) Because quality indicators and expert opinions often can be contradictory, identification of these parameters and their importance for each of IPs require adequate solutions, which can be identified among multicriterial quality assessment methods.

(4) The most qualitative input data for quality evaluation are the objective data, collected directly from the outputs of the technological processors. Corroboration of inputs/outputs of technological development processes and of quality assurance can offer a good basis for improvement of quality.

BIBLIOGRAPHY

All web references were verified on December 10, 2019

1. ICT Development Index 2017. ITU, 2017 (publ. 06.11.2019). https://www.itu.int/net4/ITU-D/idi/2017/index.html/
2. Gross Domestic Product in the Republic of Moldova in 2018 and in quarter IV 2018. (publ. 5.03.2019) https://statistica.gov.md/newsview.php?l=en&id=6300&idc=168
3. ИТ-рынок Израиля (for 2018-2017, publ. 06.11.2019). http://www.tadviser.ru/index.php/Статья:ИТ-рынок_Израиля/
4. ISACA Model Curriculum for IS Audit and Control. ISACA, 2nd Edition, 2009. 57 p.
5. Sundararajan S., Bhasi M., Pramod K. An Empirical Study of industry practices in Software Development Risk Management. In: International Journal of Scientific and Research Publications. 2013; 3(6). pp.1-11.
6. The State of Project Management Annual Survey 2016. Management WP, 2016. https://www.wellingtone.co.uk/wp-content/uploads/2016/01/The-State-of-Project-Management-Survey-2016.pdf. 23 p.
7. Keil M., Cule P., Lyytinen K. et al. A Framework for Identifying Software Project Risks. In: Communications of the ACM. 1998; 41(11). pp.76-83.
8. Jiang J., Klein G. Software project risks and development focus. In: Project Management Journal. 2001; 32(1). pp.4-9.
9. Tiwana A., Keil M. The one-minute risk assessment tool. In: Communications of the ACM. 2004; 47(11). pp. 73-77.
10. Hoermann S., Schermann M., Krcmar H. When to manage risks in IS projects: An exploratory analysis of longitudinal risk reports. In: Proceeding of 10th International Conference on Wirtschaftsinformatik, 2011. pp. 871-880.
11. Jiang James J. Software project risks and development focus - double (20-23). In: Project Management Journal, 2001; 31(1). pp. 4-9.
12. Dominguez J. The Curious Case of the CHAOS Report 2009. (publ. 01 July 2009). https://www.projectsmart.co.uk/the-curious-case-of-the-chaos-report-2009.php.
13. Hewagamage C/, Hewagamage K. Redesigned Framework and Approach for IT Project Management. In: International Journal of Software Engineering and Its Applications. 2011; 5(3). pp. 89-106.
14. International SG. CHAOS Manifesto 2012. https://cs.calvin.edu/courses/cs/262/kvlinden/resources/CHAOSManifesto2012.pdf, 64 p.
15. International SG. CHAOS Manifesto 2013. http://fliphtml5.com/scht/mwde/basic, 33 p.

16. International SG. CHAOS REPORT 2015. https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf, 13 p.
17. Carbone T., Tippett D. Project Risk Management Using the Project Risk FMEA. In: Engineering Management Journal, 2004; 16(4). pp. 28-35.
18. Mansoorzadeh S., Yusof S., Mansoorzadeh S., et al. A Comprehensive and Practical Framework for Reliable Scheduling in Project Management. In: Advanced Materials Research. 2014; 903. pp.378-383.
19. Brandas C., Didraga O., Bibu N. Study on Risk Approaches in Software Development Projects. In: Informatica Economică. 2012; 16(3). pp.148-157.
20. Digital Moldova 2020 (2013). https://mei.gov.md/en/content/digital-moldova-2020/. 37 p.
21. Grand strategy for Israel: Reflections and Directions. Samuel Neaman Institute for National Policy Research, 2017. 419 p.
22. A Guide to the Project Management Body of Knowledge (PMBOK). PMI, Sixth Edition U.S.A., 2017. 976 p.
23. Rob Cole and Edward Scotcher. Brilliant Agile project management. A practical guide to using Agile, Scrum and Kanban. U.S.A., 2015. 198 p.
24. Cobb C. The Project Manager’s Guide to Mastering Agile. Principles and Practices for an Adaptive Approach. U.S.A., 2015. 426 p.
25. ISO 9001:2015. Quality management systems. Requirements.
26. ISO 10006:2017. Quality management. Guidelines for quality management in projects.
27. ISO/IEC 20000-1:2018. IT Service management system. Requirements.
28. ISO/IEC 27001:2017. Information security management systems. Requirements.
29. ISO/IEC/IEEE 12207:2017. Systems and software engineering. Software life cycle processes.
30. Guide to the Quality Body of Knowledge (QBOK®). U.S.A., ASQ, v. 2.0, 2015. 32 p.
31. A Guide to the Project Management Body of Knowledge (PMBOK). PMI, Fourth Edition U.S.A., 2008.
32. The Project Management Cycle. PM4DEV, 2007. 15 p.
33. ISO/IEC 25010:2011. Systems and software quality requirements and evaluation (SQuaRE). System and software quality models.
34. Rose K. Project Quality Management: Why, What and How. PMP, 2005. 193 p.
35. Dwyer R. IT Project Management. SME IT Strategy and Management Resource, 2010. 10 p.
36. Garvin D. What Does Product Quality Really Mean? In: Sloan Management Review. 1984; 26(1). pp. 25-43.
37. ISO 8402:1994. Quality management and quality assurance. Vocabulary.

38. ISO/IEC 25023:2016. Systems and software Quality Requirements and Evaluation (SQuaRE). Measurement of system and software product quality.
39. ISO/IEC 25001:2014. Systems and software quality requirements and evaluation (SQuaRE). Planning and management.
40. IEEE 730-2014. IEEE Standard for Software Quality Assurance Processes.
41. Hill G. The Complete Project Management Office Handbook. ESI International Project Management Series; 2008. 690 p.
42. Verhoef C., Eveleens J. The Rise and Fall of the Chaos Report Figures. In: IEEE Software. 2010; 27(1). pp. 30-36.
43. Silberberg D., Mitzel G. Information Systems Engineering. In: Johns Hopkins APL Technical Digest. 2005; 26(4). pp. 343-349.
44. Bergmann R. Information systems and their business value. In: Materials of International Conference „European Economic Integration”. Chisinau: USEM, 2016, p.58-60.
45. Raja Faheem. What is MIS? Introduction & Definition. Types of Information Systems (publ. 6 November, 2019). https://rajafaheem.wordpress.com/2019/11/06/module-1/
46. Quality management principles. ISO 9000, ISO 9001 and related ISO quality management standards are based on these seven QMPs 2015. https://www.iso.org/files/live/sites/isoorg/files/archive/pdf/en/pub100080.pdf
47. Hartney J., Comment L. The Best KPI's for your Business. Project Engineer 2014. https://www.projectengineer.net/the-best-kpis-for-your-business/
48. PMI's Success Rates Rise Transforming the high cost of low performance. Pulse of the Profession. 9th Global Project Management Survey 2017. https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf, 32 p.
49. PwC, Insights and Trends: Current Portfolio, Programme, and Project Management Practices, The third global survey on the current state of project management, 2012. https://www.pwc.com.tr/en/publications/arastirmalar/pages/pwc-global-project-management-report-small.pdf, 40 p.
50. Daw Alwerfalli, Aslihan Karatas. Application of Quality Management Systems in Construction Industry. In: Proceedings of the 2016 International Conference on Industrial Engineering and Operations Management. Detroit, Michigan, USA: IEOM Society International, 2016. pp. 257 – 264.
51. Microsoft Operations Framework (MOF) Executive Overview v 3.0. Microsoft, 2004. 27 p.
52. Brinkworth J. Project Management and Quality Plan. DG Enterprise – Gavino Murgia Project Team. 2001 (3). 26 p.

53. Kazaz A., Birgonul M., Ulubeyli S. Cost-based analysis of quality in developing countries: a case study of building projects. In: Building and Environment. 2015; 40(10). pp. 1356-1365. https://doi.org/10.1016/j.buildenv.2004.11.010
54. Crosby P. Quality is Free: the art of making quality certain. New York: McGraw-Hill, 1979. 309 p.
55. Deming W. Out of the Crisis. MIT-CAES,1986. 523 p.
56. Deming W. The New Economics for Industry, Government & Education. MIT -CAES, 1993. 266 p.
57. Juran J. Juran on Quality by Design: The New Steps for Planning Quality into Goods and Services. The Free Press; 1992. 538 p.
58. Rory O'Connor. Massachusetts Institute of Technology. Introduction to ISO/IEC software engineering standards (Presentation), 75 p. https://www.coursehero.com/file/33970764/Intro-to-ISO-IEC-SE-standards-02RO-1ppt/
59. ISO/IEC 15504-5:2012 IT Process assessment. Part 5: An exemplar software life cycle process assessment model.
60. ISO/IEC 25022:2016. Systems and software quality requirements and evaluation (SQuaRE). Measurement of quality in use.
61. ISO/IEC 25066:2016. Systems and software Quality Requirements and Evaluation (SQuaRE). Common Industry Format (CIF) for Usability, Evaluation Report.
62. ISO/IEC TR 29110-1:2016. Lifecycle profiles for Very Small Entities (VSEs). Part 1: Overview.
63. Dumitrascu A., Nedelcu A. Project Costs and Risks Estimation Regarding Quality Management System Implementation. In Tech. 2012 (Chapter 3). pp. 41-68.
64. Leffingwell, D., Widrig, D. (First Edition 1999, Second Edition 2003). Managing Software Requirements: A Use Case Approach: Addison Wesley.
65. Arthur Hicken. What Is Shift-Left Testing? DZone, 2018. https://dzone.com/articles/what-is-shift-left-testing
66. Callum Gavin. Massively reduce the cost of bugs with error tracking. RAYGUN, 2014. https://raygun.com/blog/massively-reduce-the-cost-of-bugs-with-raygun-error-tracking/
67. Rayskiy, A. Why Should Testing Start Early in Software Project Development? XB Software, 2017. https://xbsoftware.com/blog/why-should-testing-start-early-software-project-development/
68. What is PRINCE2? https://www.prince2.com/uk/what-is-prince2
69. The IPMA Individual Competence Baseline (IPMA ICB), 2017. https://www.ipma-usa.org/standards/the-ipma-individual-competence-baseline/

70. PRINCE2® and ICB3® – Similarities and differences, 2015. https://www.ipma.world/prince2-and-icb3-similarities-and-differences/
71. Ronald Moen. Associates in Process Improvement-Detroit (USA). Foundation and History of the PDSA Cycle, 2009. 9p. https://deming.org/uploads/paper/PDSA_History_Ron_Moen.pdf
72. Juran J., Godfrey A. Juran's Quality Handbook. McGraw-Hill; 1998. 1730 p.
73. Project Quality Management. PM4VED, 2008 (22 p.), 2016 (23 p.) https://www.pm4dev.com/ resources/docman/pm4dev-ebooks/5-project-quality-management/file.htm/
74. ISO/IEC/IEEE 90003:2018. Software Engineering: Guidelines for the application of ISO 9001:2015 to computer software.
75. Lungu I. Integrating Quality Management System into Software Development Processes. ASSIST; 2017. https://assist-software.net/blog/integrating-quality-management-system-software-development-processes
76. Paulk M., Curtis B., Chrissis M. et al. Capability Maturity Model for Software. Software Engineering Institute Carnegie Mellon University, 1993. 87 p. https://apps.dtic.mil/dtic/tr/fulltext/u2/a263403.pdf
77. Capability Maturity Model Integration (CMMI). Software Engineering Institute. SEI, 2001. 708 p. https://www.uio.no/studier/emner/matnat/ifi/nedlagte-emner/INF5180/h05/ undervisningsmateriale/CMMIv1.1ippd-staged.pdf
78. Forth S. Pricing across maturity models. 2019 https://nq-chang.squarespace.com/blog/2019/7/4/pricing-across-maturity-models
79. Group T. Improving Capability and Performance with CMMI V2.0: What has changed?, 2019, http://processgroup.com/improving-capability-and-performance-with-cmmi-v2-0-what-has-changed/
80. Turner J. The Handbook of Project-based Management: Leading Strategic Change in Organizations. McGraw-Hill Companies; 2009. 473p.
81. Andersson L. Quality Assurance in the IT Project Management Process. Uppsala Universitet. 2007. 53 p. https://www.utn.uu.se/sts/student/wp-content/uploads/gamla%20exjobb/0708_anderssonlinnea.pdf
82. Tiwana A., Bush A. A Comparison of Transaction Cost, Agency, and Knowledge-Based Predictors of IT Outsourcing Decisions: a U.S.–Japan Cross-Cultural Field Study. In: Journal of Management Information Systems, 2007; 24(1). pp. 259-300.

83. Bakker K., Boonstra A., Wortmann H. Does risk management contribute to IT project success? A meta-analysis of empirical evidence. In: International Journal of Project Management, 2010; 28(5). pp. 493-503.
84. Sarigiannidis L., Chatzoglou P. Software Development Project Risk Management: A New Conceptual Framework. In: Journal of Software Engineering & Applications, 2011; 4(5), pp. 293-305.
85. Zhang L. ITIL Introduction (Presentation). ADP Small Business Services, 2006, 67p. http://slideshowes.com/doc/796482/itil-introduction-ppt/
86. ISO. Quality management systems — Fundamentals and vocabulary (ISO 9000:2000).
87. Rudd C. An Introduction Overview of ITIL. itSMF, 2004, 42 p. http://www.itsmf.co.il/Articles/ITIL_Overview_Pocket_Guide.pdf
88. IT Governance Institute ITGI. Aligning CobiT® 4.1, ITIL® V3 and ISO/IEC 27002 for Business Benefit. OGC, 2008.
89. Project Quality Management. CDC, 2006, 5p. https://www2.cdc.gov/cdcup/library/practices_guides/CDC_UP_Quality_Management_Practices_Guide.pdf
90. COBIT Transforming Enterprise IT (Presentation). ISACA, 2009, 17 p. https://www.slideshare.net/yudz_maestro/cobit-overview-9429864
91. Burger R. 20 Surprising Project Management Statistics (2016). https://blog.capterra.com/surprising-project-management-statistics/
92. Burger R. 15 Incredible Agile Project Management Statistics to Know (2018). https://blog.capterra.com/agile-project-management-statistics-for-2018.
93. Elbertsen L., Kremser T. How Agile are companies in Luxembourg? PwC, 2017, 28 p. https://www.pwc.lu/en/technology/docs/pwc-pmi-study.pdf
94. The 13th Annual State of Agile Report. VersionOne C (2019). 16 p. https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508
95. Manifesto for Agile Software Development (2001). https://www.agilealliance.org/agile101/the-agile-manifesto/
96. Ingram I. How Agile keeps projects on track, on time, on budget: vertical motion (2019). https://www.verticalmotion.ca/2019/05/08/how-agile-keeps-projects-on-track-on-time-on-budget/
97. Pflaeging N. Org Physics - Explained. How structures work, how they interact, what this means for organizational effectiveness and change (2011). 27p. https://www.slideshare.net/npflaeging/betacodex11-the-3-structures-of-an-organization
98. Yoder J., Wirfs-Brock R. Aguiar A. QA to AQ. Patterns about transitioning from Quality Assurance to Agile Quality (2014), 18 p. http://www.wirfs-brock.com/PDFs/QA2AQ.pdf

99. SAFe for Lean Enterprises 5.0 (2019). https://www.scaledagileframework.com/safe-for-lean-enterprises/,

100. Bergmann R., Bragaru T. Standards and Software Quality Models. In: Scientific and didactic journal ECONOMICA, nr.2 (108), Chişinău: ASEM, June 2019. pp. 118-132. https://ase.md/files/publicatii/economica/2019/ec_2019_2.pdf/

101. McCall J., Richards P., Walters G. Factors in Software Quality. RADC TR-77-369, Rome Air Development Center, Rome. National Technical Information Service (NTIS). 1977; Volumes I, II, and III. http://www.dtic.mil/dtic/tr/fulltext/u2/a049014.pdf, https://apps.dtic.mil/dtic/tr/fulltext/u2/a049015.pdf, https://apps.dtic.mil/dtic/tr/fulltext/u2/a049055.pdf

102. Boehm B. W. et all. Characteristics of Software Quality. TRW Series of software Technology. North-Holland Pub. Co.; Amsterdam 1978. 169 p.

103. Ghezzi C., Jazayeri M., Mandrioli D. Fundamental of software Engineering. Prentice Hall; 1991. 573 p.

104. Grady RB. Practical software metrics for project management and process improvement. Prentice Hall; 1992. 270 p.

105. IEEE. IEEE 1219-1993 Standard for Software Maintenance. 45 p. https://www.idi.ntnu.no/grupper/su/publ/ese/ieeestd-1219-1993-swmaint.pdf

106. Dromey R. A model for software product quality. In: IEEE Transactions on Software Engineering. 1995; 21(2). pp. 146-162.

107. Hyatt L., Rosenberg L. A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality. Proceeding of Product Assurance Symposium and Software Product Assurance Workshop Noordwyk. 1996/ pp. 209-212.

108. ISO/IEC 9126-1:2001. Software engineering. Product quality. Part 1: Quality model.

109. Bansiya J., Davis C. A Hierarchical Model for Object-Oriented Quality Assessment. In: IEEE Transactions on Software Engineering, 2002; 28(1). pp. 4-17.

110. Miguel J., Mauricio D., Rodríguez G. A Review of Software Quality Models for the Evaluation of Software Products. In: International Journal of Software Engineering & Applications (IJSEA), 2014; 5(6). pp. 31-53.

111. Lazić L., Kolašinac A., Avdić D. The Software Quality Economics Model for Software Project Optimization. In: WSEAS transactions on computers. 2009; 8(1), https://www.pm4dev.com/resources/docman/pm4dev-ebooks/5-project-quality-management/file.htm/

112. Bergmann R. Models and standards of information quality assurance. In: Наука №4-3, февраль 2016 Костанай, научно-производственным журнал, ISSN 1684-9310 pp. 9-12. https://www.twirpx.com/file/1934885/

113.Bazzana G., Andersen O., Jokela T. ISO 9126 and ISO 9000: friends or foes? In: Software Engineering Standards Symposium, 1993. pp. 79-88.
114.Côté M., Suryn W., Martin R., et al. Evolving a Corporate Software Quality Assessment Exercise: A Migration Path to ISO/IEC 9126. In: Software Quality Professional 2004; 6(3). pp .4-17.
115.Bulavko A. SQA Research. Kingston University, 2017, 27p. https://arturasbulavko.com/documents/SQA_Research.pdf
116.Manuel F. Bertoa, Antonio Vallecillo. Quality Attributes for COTS Components. In: I+D Computación. 2002; 1(2). pp.128-144. https://pdfs.semanticscholar.org/0cbb/94003e4d42ae1e89384c4e2a46740ac38603.pdf
117.Elli G. GEQUAMO-A Generic, Multilayered, Customizable Software Quality model. In: Software Quality Journal, 2003; 11(4). pp. 313-323.
118.Alvaro Alexandre Eduardo Santana de Almeida, Marcos Alexandre Lins de Vasconcelos, Romero Silvio de Lemos Meira. Towards a Software Component Quality Model. In: Proceeding of the 5th International Conference on Quality Software (QSIC), 2005. 11p. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.7840&rep=rep1&type=pdf
119.Rawashdeh Adnan, Matalkah Bassem. A New Software Quality Model for Evaluating COTS Components. In: Journal of Computer Science 2006; 2(4). pp. 373-381. https://pdfs.semanticscholar.org/cce1/1b6114a409629e19862a5bc49c8557e578d8.pdf
120.Glott Ruediger et all. Quality Models for Free/Libre Open Source Software Towards the „Silver Bullet"? Proceeding of EUROMICRO Conference on Software Engineering and Advanced Applications IEEE Computer Society, 2010. pp. 439-446.
121.Adewole A., Misra S., Omoregbe N. A Review of Models for Evaluating Quality in Open Source Software. Proceeding of International Conference on Electronic Engineering and Computer Science, IERI Procedia, 2013; 4. pp. 88-92.
122.Haaland Kirsten et al. Free/Libre Open Source Quality Models-a comparison between two approaches. 4th FLOS International Workshop on Free/Libre/Open Source Software, 2010. pp. 1-17.
123.Egyed A. et al. Using the WinWin Spiral Model: A Case Study. IEEE Computer. 1998; 31. pp. 33-44.
124. Al-Kilidar H., Cox K., Kitchenham B.The use and usefulness of the ISO/IEC 9126 quality standard. In: Proceeding of the International Symposium on Empirical Software Engineering (ISESE '05); IEEE Computer Society, 2005. pp. 126-132.
125.Kitchenham B., Pfleeger S.L. Software Quality: The Elusive Target. In: IEEE Software. 1996; 13(1). pp.12-21.

126.Lochmann K. Defining and Assessing Software Quality by Quality Models. Technischen Universität München, 2013, 173p. http://mediatum.ub.tum.de/doc/1169637/955300.pdf
127.Al-Qutaish Rafa E. Quality Models in Software Engineering Literature: An Analytical and Comparative Study. Journal of American Science. 2010; 6(3). pp. 166- 175.
128. Costaș I. Applying knowledge and quality management to it project management. Conferința internațională „Competitivitatea și Inovarea în Economia Cunoașterii, 2017”; Chişinău: ASEM; 2018. pp. 13-16.
129.Tomar V. Scatter Diagram (2019). https://visheshtomar.com/scatter-chart/
130.Creating a Pareto Chart in Excel (Static & Dynamic). https://trumpexcel.com/dynamic-pareto-chart-in-excel/
131.Fishbone Diagram. (ASQ, 2019). https://asq.org/quality-resources/fishbone
132.OMNITAB 80 (NIST, 2018). https://www.nist.gov/itl/sed/omnitab-80
133.MINITAB. Graphical analysis and before/after control charts can you bike to work on time? (2019) https://www.minitab.com/en-us/products/minitab/quick-start/getting-to-work/
134.Tolbakov E. Top Five Agile Management Tools: weekly-geekly (2015). https://weekly-geekly.github.io/articles/290340/index.html
135.Bergmann R. Information Projects Quality Model. In: EPH - International Journal of Science and Engineering, November 2016; 2(11). pp. 38-56.
136.Wand Y., Wang R. Anchoring Data Quality Dimensions in Ontological Foundations. In: Communications of the ACM. 1996; 39(11). pp. 86-95. http://0374288.netsolhost.com/pdf/MIT-wandwang.pdf
137.Wang R., Storey V., Firth C. A Framework for Analysis of Data Quality Research. In: IEEE transactions on knowledge and data engineering. 1995; 7(4). pp. 623-640.
138.Bergmann R. Providing Quality of Information Projects. Proceeding of International Conference Mathematics & Information Technologies (MITRE-2015). Chişinău: CEP USM, 2015, p. 90.
139.Wang R., Strong D. Beyond accuracy: what data quality means to data consumers. In: Journal of Management Information Systems. 1996; 12(4). pp. 5-33.
140.Haug A., Arlbjørn J., Pedersen A. A classification model of ERP system data quality. In: Industrial Management & Data Systems. 2009; 109(8). pp.1053-1068.
141.Levitin A., Redman T. Data as a Resource: Properties, Implications, and Prescriptions. In: Sloan Management Review. 1998; 40(1). pp. 89-101.
142.Haug A., Zachariassen.F, Liempd D. The costs of poor data quality. In: Journal of Industrial Engineering and Management. 2011; 4(2). pp. 168-193.

143.Knight S., Burn J. Developing a Framework for Assessing Information Quality on the World Wide Web. In: Informing Science Journal. 2005; 8. pp.159-172. https://pdfs.semanticscholar.org/a7fb/08f4cf1aa8ea8b7af35b05650be7aee787d9.pdf
144.Madnick S., Wang R., Lee Y., et al. Overview and Framework for Data and Information Quality Research. In: ACM Journal of Data and Information Quality, 2009, 1(1). 22 p. https://irandoc.ac.ir/sites/fa/files/attach/event/overviewandframeworkfordatamadnick.pdf
145.Bovee M., Srivastava R., Mak B. A Conceptual Framework and Belief-Function Approach to Assessing Overall Information Quality. In: International Journal of Intelligent Systems. 2003; 18(1). pp. 51-74.
146.Price R., Shanks G. A Semiotic Information Quality Framework Decision Support in an Uncertain and Complex World. In: The IFIP TC8/WG83 International Conference 2004. pp. 658-672.
147.Stvilia B., Gasser L., Twidale M., et al. A Framework for Information Quality Assessment. In: Journal of the American Society for Information Science and Technology. 2007; 58(12). pp.1720–1733.
148.Madnick S., Wang R. Introduction to the TDQM Research Program. In: Sloan School of Management MIT. 1992. http://web.mit.edu/tdqm/papers/92/92-01.html
149.Wang R., Lee Y., Pipino L., et al. Manage Your Information as a Product. In: Sloan Management Review. 1998; 39(4). pp. 95-105.
150.Miller G. The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review. 1956; 63(2). pp. 81-97.
151.Fan W. Dependencies revisited for improving data quality. Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 2008. pp.159-170.
152.Dasu T., Johnson T. Exploratory Data Mining and Data Cleaning. A John Wiley & Sons, Inc. Publication. 2003. 212 p. https://onlinelibrary.wiley.com/doi/book/10.1002/0471448354
153.Herzog T., Scheuren F., Winkler W. Data Quality and Record Linkage Techniques. Springer. 2007, 234 p. https://dl.acm.org/citation.cfm?id=1534235
154.Eysenbach G., Powell J., Kuss O., et al. Empirical studies assessing the quality of health information for consumers one the World Wide Web. A Systematic Review. JAMA. 2002; 287(20). pp. 2691-2700.
155. Hongwei Zhu, Yinghua Ma, Guiyang Su. Collaboratively Assessing Information Quality on the Web. (2012). https://pdfs.semanticscholar.org/2a0b/74185663e13cfccc9a836417b7db1a80f829.pdf?_ga=2.37646249.820014572.1579426933-1742148397.1579426933

156. Costaş I. Managing for knowledge through the quality management of information system. The Third Conference of Mathematical Society of the Republic of Moldova; 2014; Chişinău, Acad. of Sciences of Moldova; 2014. pp 485-488.
157. Bergmann R. Quantifying Information Quality. In Studia Universitatis Moldaviae nr 7(87), 2015. pp. 86-97.
158. ISO/IEC 25012:2008. Systems and software quality requirements and evaluation (SQuaRE). Data quality model.
159. Marc-Alexis Côté M., Witold Suryn, Elli Georgiadou E. Software Quality Model Requirements for Software Quality Engineering. In: Software Quality Management & INSPIRE Conference (BCS), 2006. 16 p. http://eprints.mdx.ac.uk/2077/, https://pdfs.semanticscholar.org/de1d/bd9ba3ec0399ef373c68903806bbf2b7ecce.pdf
160. Caballero I., Caro A., Calero C., et al. IQM3: Information Quality Management Maturity Model. In: Journal of Universal Computer Science. 2008, Vol.14(22). pp. 3658-3685.
161. Bergmann R. Information Projects Quality Model. Proceeding of International Conference Mathematics & Information Technologies (MITRE). Chişinău: CEP USM, 2016. pp.74-75.
162. Bergmann R. Information Projects Quality Assessment. Proceeding of International Conference Mathematics& Information Technologies (MITRE-2019). Chişinău: CEP USM, 2019, p.68.
163. Bergmann R. Information Projects Quality Model. The 5th social psychology conference for PhD students, SODOCO, Haifa University, Israel, December 12, 2016.
164. Bergmann R., Sirota, J. Information Projects Quality Model and the Global Volume of Data. The 5th Kinneret Conference on Software Engineering Education, Kinneret Academic College, Israel, February 21, 2017, pp. 18-19.
165. Napitupulu J., Hasibuan D. Study Approach ELimination Et Choix Traduisant la REalite (ELECTRE) for Dynamic Multi-Criteria Decision. In: International Journal of Scientific Research in Science and Technology. 2017; 3(3). pp. 460-465.
166. Stăncioiu I, Purcărea A, Niculescu C. Management, development, research. Bucharest: Ed MONDERO, 1993. 244 p.
167. Theodore C. Fowler, Combinex® - A Method for Sound Decisions. 2005, 12 p. http://sive.org/files/site1/user_files_b69e0b/admin-A-10-1-6-9b92fe9.pdf

Annex 1. Quality models according to the ISO 25000 family of standards

(Fragments of Quality Knowledge Data Base)

	A	B	C	D
1	Name of characteristics	Name of subcharacteristics	Nume caracteristici	Nume sucaracterestici
2	**1. Effectiveness**	1. Effectiveness	**1. Eficacitatea**	**1. Eficacitatea**
3	**2. Efficiency**	2. Efficiency	**2. Eficiență**	**2. Eficiență**
4	**3. Satisfaction**	3.1. Usefulness	**3. Satisfacția**	**3.1. Utilitate**
5		3.2. Trust		**3.2. Încredere**
6		3.3. Pleasure		**3.3. Plăcere**
7		3.4. Comfort		**3.4. Confort**
8	**4. Freedom from risk**	4.1.Economic risk mitigation	**4. Libertatea de risc**	**4.1. Reducerea riscurilor economice**
9		4.2. Health and safety risk mitigation		**4.2. Reducerea riscurilor pentru sănătate și siguranță**
10		4.3.Environmental risk mitigation		**4.3. Atenuarea riscului mediului**
11	**5. Context coverage**	5.1. Context completeness	**5. Context acoperire**	**5.1. Completitatea contextului**
12		5.2.Flexibility		**5.2. Flexibilitate**

Terminology | **Q.in use Model (ISO 25010)** | Q.of product model (ISO 25010) | Q.Product Measure ...

	A	B	C	D
1	Name of characteristics	Name of subcharacteristics	Numele caracteristicilor	Numele subcharacteristics
2	**1. Functional suitability**	1.1. Functional completeness	1. Adecvare funcțională	1.1. Completitudine funcțională
3		1.2. Functional correctness		1.2. Corectitudinea funcțională
4		1.3. Functional appropriateness		1.3. Potențialitate funcțională
5	**2. Performance efficiency**	2.1. Time behavior	2. Eficiența performanței	2.1. Comportamentul în timp
6		2.2. Resource utilization		2.2. Utilizarea resurselor
7		2.3. Capacity		2.3. Capacitate
8	**3. Compatibility**	3.1. Co-existence	3. Compatibilitate	3.1. Coexistență
9		3.2. Interoperability		3.2. Interoperabilitate
10	**4. Usability**	4.1. Appropriateness recognizability	4. Utilizabilitate	4.1. Recunoașterea adecvării
11		4.2. Learnability		4.2. Abilitatii de a invata
12		4.3. Operability		4.3. Operativitate
13		4.4. User error protection		4.4. Protecția împotriva erorilor utilizatorului
14		4.5. User interface aesthetics		4.5. Estetica interfeței cu utilizatorul
15		4.6. Accessibility		4.6. Accesibilitate
16	**5. Reliability**	5.1. Maturity	5. Fiabilitate	5.1. Maturitate
17		5.2. Availability		5.2. Disponibilitate

... | Quality in use Model | **Quality of product model** | QProduct Measures | QUse Metrics | Quality o ...

C	D	E	F	G	H	I
Clause	NameSubchar	Id_cod	Id_Measure	Name_of_Mesure	Explanation	Measurement_function
8.2.	1.1. Functional completeness	111	FCp-1-G	Functional coverage	What proportion of the specified coverage	X = 1-A/B; A = number of functions n
	1.2. Functional correctness	121	FCr-1-G	Functional correctness	What proportion of functions provides the c	X = 1-A/B, A = Number of functions t
	1.3. Functional appropriateness	131	FAp-1-G	Functional appropriateness of usage objective	What proportion of the functions required b	X = 1-A/B, A = Number of functions r among those that are required for ac objective; B = Number of functions r
		132	FAp-2-G	Functional appropriateness of system	What proportion of the functions required b	X=∑(i=1 to n)⟦Ai/n⟧; Ai = Appropriate
8.3.	2.1. Time behavior	211	PTb-I-G	Mean response time	How long is the mean time taken by the sy	X=∑(i=1 to n)⟦Ai/n⟧; Ai = Time taker
		212	PTb-2-G	Response time adequacy	How well does the system response time m	X= A/B, A = Mean response time me
		213	PTb-3-G	Mean turn around time	What is the mean time taken for completion	X=∑(i=1 to n)(Bi-Ai), Ai = Time of sta
		214	PTb-4-G	Turnaround time adequacy	How well does the turnaround time meet th	X=A/B, A = Mean turnaround time m

odel (ISO 25010) | **Q.Product Measures (ISO 25023)** | QUse Metrics | Quality of da ...

Annex 2. Fragment of Data quality model

Clause	Characteristic	Inherent Data Quality	System Dependent Data Q.	Definition	Ecample of Measure	Example of Measurement function	Data Validation
5.3.1.1.	Accuracy	X		The degree to which data has attributes that correctly represent the true value of the intended attributes of a concept or event in a specific context of use.	Record's field syntactic accuracy	X=A/B; A=number of records with the s	A≤B
					Record's field semantic accuracy	X=A/B; A=number of records with the s	A≤B
5.3.1.2.	Completeness	X		The degree to which subject data associated with an entity has values for all expected attributes and related entity instances in a specific content of use.	Completeness of data within a file	A/B; A=number of data, required for the	A≤B
5.3.1.3.	Consistency	X		The degree to which data has attributes that are free from contradiction and are coherent with other data in a specific context of use. It can be either or both among data regarding one entity and across similar data for comparable entities.	Consistency of a data file	A/B; A=number of data consistent in the file; B=nur	
5.3.1.4.	Credibility	X		The degree to which data has attributes that are regarded as true and be	Credibility of data used by a bank for evaluating credit risk	A/B; A= Number of data certified by internal audit a	
5.3.1.5.	Currentness	X		The degree to which data has attributes that are of the right age in a specific context of use.	Currentness of a field dat	X=A/B; A=number of data inspections where the de	
5.3.2.1.	Accessibility	X	X	The degree to which data can be accessed in a specific context of use, particularly by people who need supporting technology or special configuration because of some disability	Ex. of inherent data qualit	X=A/B; = number of data stored only as "sound" (e	
					Ex. of system dependent	X=A/B; A=Number of data that the differently able	
5.3.2.2.	Compliance	X	X	The degree to which data has attributes that adhere to standards, conventions or regulations in force and similar rules relating to data quality in a specific context of use.			

QUse Metrics (ISO 25022) | DataQ.Metrics (ISO 25012) | DataQ Metrics | Formula types | Import

Annex 3. Field Research Questionnaire

<u>Quality of Information Projects Questionnaire</u>

Questions:

1. What is your area of expertise in the organization you are working in?
    a. Finance/Controlling
    b. Logistics/Distribution
    c. Production
    d. Procurement
    e. Maintenance/Service
    f. Quality Assurance
    g. Engineering and Infrastructure
    h. Sales/Customer Service
    i. Information Technologies
    j. Human Resources
    k. Other:_______________

2. What kind of information project you participate in?
    a. Transaction Processing System (ERP/CRM)
    b. GIS & Map Library
    c. Enterprise Portal & Knowledge Management
    d. BI & Big Data
    e. Internet Site & Web Application
    f. Document Management System
    g. Mobile Application
    h. Other:___________

3. For how long are you working within the information project?
    a. Less than one year
    b. Between 1 to 3 years
    c. Between 3 to 6 years
    d. Between 6 to 10 years
    e. Over 10 years

4. Are you interested to participate in an interview about quality of information project?
    Yes □ No □

If Yes, please leave your contact information or give your business card with this questionnaire:

Name: ____________________ Phone: ________________

Email: ____________________ Company: ______________

5. Please rank the Importance of the Quality characteristics regarding the information project types begins with the most importance characteristic (1-Very Important) to the most unimportance characteristic (5-Unimportant). Please do not use the same importance level more than 5 times, for each information project.

Importance levels: 1. Very Important, 2. Important, 3. Moderately Important, 4. Of Little Importance, 5. Unimportant

Information projects / **Characteristics**	ERP & CRM	GIS & Map Library	Enterprise Portal & Knowledge Management	BI & Big Data	Internet Site & Web Application	Document Management System	Mobile Application
1. Accuracy	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
2. Availability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
3. Changeability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
4. Correctness	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
5. Efficiency	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
6. Flexibility	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
7. Functionality	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
8. Interface facility	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
9. Integrity	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
10. Interoperability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
11. Maintainability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
12. Modifiability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
13. Performance	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
14. Portability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
15. Reliability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
16. Reusability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
17. Robustness	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
18. Scalability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
19. Security	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
20. Supportability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
21. Testability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
22. Transferability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
23. Understandability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
24. Usability	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
25. Visibility	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5

The definitions of the main quality characteristics are presented in Italic font below:

1. ***Accuracy*** The capability of the software product to provide the right or agreed results or effects with the needed degree of precision.
2. ***Availability*** The degree to which a work is operational and available for use as a product or to users.
3. ***Changeability*** The characterization of the amount of effort to change a system.
4. ***Correctness*** The ease with which minor defects can be corrected between major releases while the application or component is in use by its users.
5. ***Efficiency*** The capability of the software product to provide appropriate performance, relative to the amount of resources used understated conditions.
6. ***Flexibility*** The effort required modifying an operational program.
7. ***Functionality*** The capability of the software product to provide functions meet stated and implied needs when the software is under specified conditions.
8. ***Interface facility.*** The degree to which two software products can be connected successfully.
9. ***Integrity*** The extent to which access to software or data by unauthorized persons can be controlled.
10. ***Interoperability*** The capability of the software product to interact with one or more specified systems.
11. ***Maintainability*** The capability of the software product to be modified.
12. ***Modifiability*** Corrections, improvements or adaptations of the software to changes in environment and in requirements and functional specifications.
13. ***Performance*** The degree to which timing characteristics are adequate.
14. ***Portability*** The capability of the software product to be transferred from one environment to another.
15. ***Reliability*** The capability of the software product to maintain a specified level of performance when used under specified conditions.
16. ***Reusability*** The ease with which an existing application or component can be reused.
17. ***Robustness*** The degree to which an executable work product continues to function properly under abnormal conditions or circumstances.
18. ***Scalability*** The ease with which an application or component can be modified to expand its existing capabilities.
19. ***Security*** A system is secure if it protects its data and services from unauthorized access and modification.
20. ***Supportability*** The ability to extend the program, adaptability and serviceability, in addition to testability, computability, configurability, the ease with which a system can be installed and the ease with which problems can be localized.
21. ***Testability*** The capability of the software product to enable modified software to be validated.
22. ***Transferability*** The cost of transferring a product from its hardware or operational environment to another.
23. ***Understandability*** The capability of the software product to enable the user to understand whether the software is suitable and how it can be used for particular tasks and conditions of use.
24. ***Usability*** The capability of the software product to be understood learned, used and attractive to the user, when used under specified condition.
25. ***Visibility*** A process-related quality meaning that all steps and the current process status are documented clearly.

Annex 4. Survey results

Test of the importance of quality characteristics for *ERP & CRM group, GIS, Enterprise Portal & Knowledge Management Group, BI & Big Data.*

Table A4.1

Field Research Average of ERP & CRM Group

ERP & CRM			
Characteristics	Mean score	No. of answers	Standard deviation
1. Accuracy	4.7143	21	0.5606
2. Availability	4.3182	22	0.8387
3. Changeability	3.0455	22	1.2141
4. Correctness	4.1818	22	0.7327
5. Efficiency	3.6364	22	1.2553
6. Flexibility	2.9565	23	1.3307
7. Functionality	4.0000	23	1.2060
8. Interface facility	3.5652	23	1.3425
9. Integrity	4.0435	23	1.3973
10. Interoperability	3.4545	22	1.2622
11. Maintainability	3.7391	23	1.1762
12. Modifiability	3.3636	22	1.1358
13. Performance	3.6087	23	1.2336
14. Portability	2.8261	23	1.5271
15.Reliability	4.0870	23	1.0407
16. Reusability	2.8182	22	1.4683
17. Robustness	3.8182	22	1.6224
18.Scalability	3.0909	22	1.1916
19. Security	4.4091	22	0.9591
20. Supportability	4.0435	23	1.0215
21.Testability	3.6818	22	1.2105
22. Transferability	2.3043	23	1.2590
23.Understandability	3.9500	20	1.3563
24.Usability	3.9545	22	1.2527
25. Visibility	3.6818	22	1.1705

Painted green - characteristics which pass through grade 3.5

Table A4.2

Field Research Average of the Geographic Information Systems Group (GIS)

GIS & Map Library			
Characteristics	Mean score	No. of answers	Standard deviation
1. Accuracy	4.4667	15	1.0601
2. Availability	4.3077	13	0.7511
3. Changeability	2.7692	13	0.8321
4. Correctness	3.6429	14	1.2157
5. Efficiency	2.9231	13	1.0377
6. Flexibility	2.8333	12	1.2673
7. Functionality	3.8333	12	1.2673
8. Interface facility	3.7273	11	1.3484
9. Integrity	3.3636	11	1.2060
10. Interoperability	2.3333	9	1.2247
11. Maintainability	3.1667	12	1.5859
12. Modifiability	3.0000	11	1.0954
13. Performance	3.9091	11	1.1362
14. Portability	3.3000	10	1.5670
15.Reliability	4.2500	12	0.8660
16. Reusability	2.5455	11	1.4397
17. Robustness	3.2000	10	1.5492
18.Scalability	3.1000	10	1.1972
19. Security	4.0000	13	1.1547
20. Supportability	3.3571	14	1.2774
21.Testability	3.2500	12	0.9653
22. Transferability	2.6923	13	1.1094
23.Understandability	3.7273	11	1.7939
24.Usability	4.3333	12	1.2309
25. Visibility	4.0000	12	0.9535

Painted green - characteristics which pass through grade 3.5

Table A4.3

Field Research Average of the Enterprise Portal & Knowledge Management Group

Enterprise Portal & Knowledge Management			
Characteristics	Mean score	No. of answers	Standard deviation
1. Accuracy	3.5385	13	0.9674
2. Availability	3.6667	12	1.0731
3. Changeability	3.2500	12	1.2154
4. Correctness	3.3636	11	0.9244
5. Efficiency	3.8000	10	1.0328
6. Flexibility	3.0000	12	1.5954
7. Functionality	3.5000	10	1.4337
8. Interface facility	3.2222	9	1.2019
9. Integrity	3.0000	10	1.4907
10. Interoperability	3.3750	8	1.3025
11. Maintainability	3.5556	9	1.3333
12. Modifiability	2.8889	9	1.0541
13. Performance	4.3000	10	1.2517
14. Portability	2.8750	8	1.2464
15.Reliability	4.2857	7	1.2536
16. Reusability	3.0000	9	1.1180
17. Robustness	3.1429	7	1.7728
18.Scalability	3.2500	8	1.1650
19. Security	3.8182	11	1.2505
20. Supportability	3.8333	12	0.8348
21.Testability	3.3000	10	1.4181
22. Transferability	3.0000	11	1.1832
23.Understandability	4.2222	9	0.9718
24.Usability	4.2000	10	1.2293
25. Visibility	4.4444	9	0.8819

- Painted green - characteristics which pass through grade 3.5
- Painted yellow- characteristics which reach the grade 3.5

Table A4.4

Field Research Average of the BI & Big Data

BI & Big Data			
Characteristics	Mean score	No. of answers	Standard deviation
1. Accuracy	4.0000	18	1.5339
2. Availability	3.6000	15	1.3522
3. Changeability	3.5000	14	1.0190
4. Correctness	3.6154	13	1.1209
5. Efficiency	3.8333	12	1.3371
6. Flexibility	3.0000	13	1.4142
7. Functionality	3.4545	11	1.5725
8. Interface facility	3.0833	12	1.5050
9. Integrity	3.6364	11	1.5015
10. Interoperability	3.2000	10	1.5492
11. Maintainability	3.0000	10	1.4142
12. Modifiability	3.1818	11	1.0787
13. Performance	3.4545	11	1.3685
14. Portability	2.8000	10	1.6193
15.Reliability	4.1818	11	1.4013
16. Reusability	3.1818	11	1.3280
17. Robustness	2.9000	10	1.7920
18.Scalability	2.7778	9	1.3944
19. Security	4.1667	12	1.1146
20. Supportability	3.0000	12	1.2060
21.Testability	3.4167	12	1.0836
22. Transferability	2.9167	12	1.1645
23.Understandability	3.8182	11	1.1677
24.Usability	4.0909	11	0.9439
25. Visibility	3.2727	11	1.3484

Painted green - characteristics which pass through grade 3.5

Painted yellow- characteristics which reach the grade 3.5

Table A4.5

Field Research Average of the Internet Sites & Web Application Group

Internet Sites & Web Application			
Characteristics	Mean score	No. of answers	Standard deviation
1. Accuracy	3.4706	17	1.3747
2. Availability	3.6250	16	1.4549
3. Changeability	2.9375	16	1.4361
4. Correctness	3.4375	16	1.0308
5. Efficiency	3.6000	15	0.8281
6. Flexibility	2.6000	15	1.0556
7. Functionality	3.6000	15	1.4041
8. Interface facility	2.6429	14	1.3363
9. Integrity	3.0714	14	1.3281
10. Interoperability	2.6154	13	1.0439
11. Maintainability	3.5000	14	1.1602
12. Modifiability	3.0000	15	1.0690
13. Performance	3.6000	15	1.5024
14. Portability	3.6154	13	1.3868
15.Reliability	4.1667	12	0.8348
16. Reusability	3.3077	13	1.6013
17. Robustness	3.0833	12	1.5050
18.Scalability	3.0000	13	1.1547
19. Security	4.0000	16	1.2111
20. Supportability	3.4118	17	1.3257
21.Testability	2.7500	16	1.1255
22. Transferability	3.0000	15	1.1952
23.Understandability	3.9333	15	1.0328
24.Usability	4.2667	15	0.9612
25. Visibility	3.8571	14	1.4601

Painted green - characteristics which pass through grade 3.5

Painted yellow- characteristics which reach the grade 3.5

Table A4.6

Field Research Average of the Document Management Systems Group

Document Management System			
Characteristics	Mean score	No. of answers	Standard deviation
1. Accuracy	3.9286	14	1.4392
2. Availability	3.3846	13	1.1929
3. Changeability	3.0769	13	1.1152
4. Correctness	3.1818	11	1.0787
5. Efficiency	3.2500	12	1.4222
6. Flexibility	2.4545	11	1.1282
7. Functionality	3.2727	11	1.4206
8. Interface facility	3.1000	10	1.1005
9. Integrity	3.6667	9	1.1180
10. Interoperability	3.2222	9	1.3944
11. Maintainability	3.3000	10	1.2517
12. Modifiability	2.6000	10	1.2649
13. Performance	3.6364	11	1.1201
14. Portability	2.6000	10	1.5055
15.Reliability	3.7778	9	1.2019
16. Reusability	2.8000	10	1.4757
17. Robustness	3.6667	9	1.4142
18.Scalability	3.1250	8	1.6421
19. Security	3.6154	13	1.5021
20. Supportability	3.1429	14	1.3506
21.Testability	3.3636	11	1.2863
22. Transferability	2.9167	12	1.3790
23.Understandability	3.2727	11	1.6787
24.Usability	3.6364	11	1.6293
25. Visibility	3.0000	10	1.4907

Painted green - characteristics which pass through grade 3.5

Table A4.7

Field Research Average of the Mobile Application Group

Mobile Application			
Characteristics	Mean score	No. of answers	Standard deviation
1. Accuracy	3.6471	17	1.3201
2. Availability	4.2353	17	1.2515
3. Changeability	3.3529	17	1.2217
4. Correctness	3.3529	17	1.1695
5. Efficiency	4.3125	16	0.7932
6. Flexibility	2.8235	17	1.2862
7. Functionality	3.9375	16	1.4818
8. Interface facility	3.9333	15	1.0998
9. Integrity	2.9231	13	1.4412
10. Interoperability	3.6154	13	1.3253
11. Maintainability	4.0714	14	1.0716
12. Modifiability	3.3846	13	1.3868
13. Performance	3.9333	15	0.9612
14. Portability	3.2143	14	1.4239
15.Reliability	3.4286	14	1.3986
16. Reusability	3.3077	13	1.1094
17. Robustness	3.5000	14	1.5566
18.Scalability	3.1538	13	1.4051
19. Security	4.0000	17	1.1726
20. Supportability	3.2941	17	1.3117
21.Testability	3.4706	17	1.4194
22. Transferability	3.7333	15	1.2228
23.Understandability	3.3750	16	1.5864
24.Usability	3.7647	17	1.3477
25. Visibility	3.4000	15	1.5492

Painted green - characteristics which pass through grade 3.5

Painted yellow- characteristics which reach the grade 3.5

Table A4.8

Field research average of the experts

Experts			
Characteristics	Mean score	No. of answers	Standard deviation
1. Accuracy	4.0769	13	1.6053
2. Availability	3.7692	13	1.4806
3. Changeability	3.5000	12	1.1677
4. Correctness	3.9231	13	1.3205
5. Efficiency	3.7692	13	0.9268
6. Flexibility	3.6667	12	0.7785
7. Functionality	4.3846	13	1.2609
8. Interface facility	3.5833	12	0.7930
9. Integrity	3.1818	11	1.1677
10. Interoperability	2.9091	11	1.0445
11. Maintainability	3.6154	13	1.2609
12. Modifiability	3.0000	11	1.2649
13. Performance	3.1538	13	1.2810
14. Portability	2.6154	13	1.5566
15.Reliability	4.1538	13	1.1435
16. Reusability	3.1538	13	0.9871
17. Robustness	3.3846	13	1.0439
18.Scalability	3.1667	12	1.0299
19. Security	3.5000	12	1.3143
20. Supportability	3.2000	10	1.2293
21.Testability	2.2727	11	1.0090
22. Transferability	2.4545	11	1.4397
23.Understandability	3.4545	11	1.3685
24.Usability	3.5000	12	1.6787
25. Visibility	2.9167	12	1.3114

Painted green - characteristics which pass through grade 3.5

Painted yellow- characteristics which reach the grade 3.5

Table A4.9

Field research average of the mean scores

Average of the mean scores		
Characteristics	Mean score	Standard deviation
1. Accuracy	3.9803	0.4398
2. Availability	3.8633	0.3675
3. Changeability	3.1790	0.2662
4. Correctness	3.5874	0.3299
5. Efficiency	3.6406	0.4140
6. Flexibility	2.9168	0.3600
7. Functionality	3.7478	0.3600
8. Interface facility	3.3572	0.4197
9. Integrity	3.3608	0.3915
10. Interoperability	3.0906	0.4392
11. Maintainability	3.4935	0.3382
12. Modifiability	3.0524	0.2574
13. Performance	3.6995	0.3462
14. Portability	2.9808	0.3597
15.Reliability	4.0414	0.2924
16. Reusability	3.0143	0.2736
17. Robustness	3.3370	0.3119
18.Scalability	3.0830	0.1425
19. Security	3.9387	0.2918
20. Supportability	3.4103	0.3546
21.Testability	3.1882	0.4556
22. Transferability	2.8772	0.4321
23.Understandability	3.7192	0.3273
24.Usability	3.9683	0.3076
25. Visibility	3.5716	0.5228

Painted green - characteristics which pass through grade 3.5

Annex 5. Implementation Act <u>at WGS</u>

Wide Geographic Systems (WGS LTD), Israel, is a company with extensive knowledge and experience in GIS mapping and GPS navigation systems.

During the Implementation three quality assessments were performed for the chosen project. The improvement in quality score can be seen and compared between the assessments.

1 מערכות גיאוגרפיות מתקדמות בע"מ WGS בס"ד

October 30, 2019

To whom it may be of concern:

This is to certify that Ran Bergmann implemented a pilot of his software application "IPMS" in our company WGS LTD, during October 2019.

It is confirmed that Ran Bergmann presented his model for improving information quality and demonstrated his software application of "IPMS" (Information Project Management System) in our company.

We used his "IPMS" software application, in one of our projects. We found it very useful to manage and monitor our project quality using his software application, in compare to other projects. We also had an excellent user experience.

If further information regarding this statement is necessary, please contact me.

Sincerely,

Mark Israel, Owner & CEO
+972-5[illegible]

דב ליון
מערכות גיאוגרפיות
מתקדמות בע"מ
ח.פ. 514612381

WIDE GEOGRAPHIC SYSTEMS

ת.ד. 170 , תלמי יחיאל, 79810 טל: 08-8601001 פקס: 08-8601003

Annex 6. Implementation Act in MSU

Annex 6

MINISTERUL EDUCAȚIEI
al REPUBLICII MOLDOVA

UNIVERSITATEA DE STAT DIN MOLDOVA

MD-2010, Chişinău
str. A. Mateevici, 60
tel: 57-74-01, fax (373-22) 24-42-48

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ
РЕСПУБЛИКИ МОЛДОВА

МОЛДАВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МД-2010, Кишинёв
ул, А. Матеевич, 60
тел: 57-74-01, факс (373-22) 24-42-48

ADEVERINȚĂ

Prin prezenta se confirmă:

Ran BERGMANN, cetățean al Statului Israel, doctorand al Departamentului Informatică a Universității de Stat din Moldova, în cadrul tezei de doctorat **„Asigurarea Calității Proiectelor Informaționale"** a elaborat un **Sistem de Management al Proiectelor Informaționale** (IPMS) destinat evaluării şi controlului calității sistemelor/produselor software,

Care **a fost implementat în procesul de instruire la Universitatea de Stat din Moldova**, *Departamentul Informatică, Specialitatea Management Informațional,* disciplina *Managementul Calității Software,* anul de studii 2019-2020, titular de curs conf.univ., dr. T. Bragaru

Sistemul elaborat IPMS este folosit în procesul de instruire în calitate de:

1. Metamodel de calitate software adaptabil şi extensibil, ce întruneşte bază generalizată de cunoştințe privind calitatea software, caracteristici, subcaracteristici şi metrici de calitate, inclusiv modele de calitate de bază şi particularizate, metode de măsurare şi formulele de calcul a indicilor de calitate conform ISO 25022, ISO 25023;
2. Instrument de simulare a proceselor de măsurare, evaluare, raportare şi interpretare grafică a calității sistemelor/produselor software în cadrul activităților practice şi de laborator.

Adeverința este eliberată pentru a confirma relevanța teoretică şi aplicativă a rezultatelor obținute de doctorandul Ran BERGMANN în cadrul tezei de doctor susnumite.

Prorectorul **Universității de Stat din Moldova**
Pentru activitatea didactică
Otilia DANDARA,
Prof.univ. dr.

Exp. Valeriu Ungureanu, conf. univ. dr.
Tel. +373 67 56 00 49

V. Ungureanu

Annex 7. AUTHOR'S Curriculum Vitae

Name: Ran Bergmann

Date and place of birth: August 8th, 1973, Germany

Citizenship: Israeli and Austrian

e-mail: bergman.rani@gmail.com

Studies - higher education:

- Moldova State University, PhD part-time Student, since November 2012, at the Faculty of Mathematics and Informatics, Department of Informatics, specialty - Computer Programming.
- Manchester University in Israel, 2002 - 2003, MBA in Human Resource Management and Organizational Consulting.
- Tel-Aviv Academic College of Engineering, Israel, 1997 - 2000, Diploma of B.Tech in Software Engineering & Industrial Management.

Studies - practical engineering:

- School of Technology of Tel-Aviv University, Israel, 1995-1997, Diploma of Technical Engineering in Industrial Management.

Professional courses:

- John-Bryce, Chief information officer (CIO), 2012, Israel.
- Ness Training Center, SAP Enterprise Portal System Administration, 2004, Israel.
- SAP UK Education Center, IDoc Interface Technology, 2002, Israel.
- SAP UK Education Center, IDoc Development, 2002, Israel.
- Ness Training Center, Business Information Warehouse, 2001, Israel.
- Ness Training Center, SAP PM Plant Maintenance, 2001, Israel.
- Ness Training Center, SAP ABAP/4 Programming, 2000, Israel.

Areas of expertise and scientific interest:

- Over 15 years' experience in information systems analysis, definition, specification and realization of business processes, which includes extensive experience in implementing ERP and PLM information systems, engineering and logistics modules.
- Over 10 years' lecturing experience in information systems analysis courses and information system project management.

Professional activities:

- McKit Systems: PLM Project manager, 2015 - today.
- Ness, Malam: an independent consultant: Project manager, information systems analyst and lecturer, 2011 - 2015.
- Top Ramdor Systems&Computers Co., SAP ERP Manager, 2010-2011.
- Danshir Systems: CTO, 2007-2009.
- B2Tech: Co.CEO, 2003-2007.
- Netafim: Development Team Manager, 2003.
- Techtrends: Project manager, information systems analysis and developer, 1998-2002.
- Deloitte &Touche Consulting Services: Writing procedures for ISO 9000 quality, 1998.

Participation in national scientific projects:

- Courses' trainer in information systems analysis, implementation and development in the following educational institutions: John Bryce, Ness College, The College of Management, One 1 College, Israel.
- The 5th social psychology conference for PhD students, poster representation, 2016, Haifa University, Israel
- The 5th Kinneret Conference on Software Engineering Education, 2017, Israel

Participation in international conference and forums:

- International scientific conference MITRE 2015, Moldova.
- International scientific conference MITRE 2016, Moldova.
- European economic integration 2016, Moldova.
- International scientific conference MITRE 2019, Moldova.

Certificate of lecturing

- Lecturing in the ministry of justice for the department of informatics in the subject: Providing Quality of Information Project, 2015, Israel.
- An excellent Lecturer for Ness College, June 2014, Israel.

Ran Bergmann

Author’ list of publications

1.1. In recognized magazines abroad

1. BERGMANN, R. Information Projects Quality Model. In: EPH - International Journal of Science and Engineering, Volume 2, Issue 11, November-2016, p.38-56. https://ephjournal.com/index.php/se/article/view/224, ISSN 2454-2016
2. BERGMANN, R. Models and standards of information quality assurance. In: Наука №4-3, февраль 2016 Костанай, научно-производственный журнал, ISSN 1684-9310 pages 9-12.: https://www.twirpx.com/file/1934885//

1.2. In magazines from the National Register of accredited profile magazines, Category B

3. BERGMANN, R. Quantifying Information Quality. In: Studia Universitatis Moldavia, 2015, nr.7 (87), Seria „Ştiințe exacte și economice”, pp. 86-97. http://studiamsu.eu/nr-7-87-2015/, ISSN 1857-2073
4. BERGMANN, R. BRAGARU, T. Standards and Software Quality Models. In: Scientific and didactic journal ECONOMICA, nr.2 (108), Chişinău: ASEM, June 2019, pp. 118-132. https://ase.md/files/publicatii/economica/2019/ec_2019_2.pdf/, ISSN 1810-9136

2. *Reports in the conference*

2.1. International conferences

5. BERGMANN, R. Providing Quality of Information Projects. Proceeding of International Conference Mathematics & Information Technologies (MITRE-2015). Chişinău: CEP USM, 2015, p.90. ISBN 978-9975-71-678-9
6. BERGMANN, R. Information systems and their business value. In: Materials of International Conference „European Economic Integration”. Chişinău: USEM, 2016, pp.58-60. ISBN 978-9975-3147-2-5
7. BERGMANN, R. Information Projects Quality Model. Proceeding of International Conference Mathematics & Information Technologies (MITRE-2016). Chişinău: CEP USM, 2016, pp.74-75. ISBN 978-9975-71-794-6
8. BERGMANN, R. Information projects quality assessment. Proceeding of International Conference Mathematics & Information Technologies (MITRE-2019). Chişinău: CEP USM, 2019, p.67-68. ISBN 978-9975-149-17-4

2.2. Home conferences

9. BERGMANN, R. Information Projects Quality Model. The 5th social psychology conference for PhD students, SODOCO, Haifa University, Israel, December 12, 2016, poster, 17 p.
10. BERGMANN, R., SIROTA, J. Information Projects Quality Model and the Global Volume of Data. The 5th Kinneret Conference on Software Engineering Education, Kinneret Academic College, Israel, February 21, 2017, pp. 18-19.

Annex 8. Declaration of responsibility

Compliance Regulations

For doctoral theses elaborated in the English language

The undersigned, declare on my own responsibility that the materials presented in the present doctoral thesis are the result of my own research and scientific achievements. I am aware of the fact that, otherwise, I will bear the consequences in accordance with the present legislation.

Bergmann, Ran

Signature

Ran Bergmann

Date

15.01.2020

Annex 9. Description of the software application

Software application name: IPMS - Information Project Management System.

Development environment: Visual Studio 2013 and Microsoft Visual Studio 2017.

DB versions: Microsoft SQL Server 2014 database and can be accessed with SQL Server 2014 and later.

Reports: The reports for this program were built using Report viewer for Visual Studio.

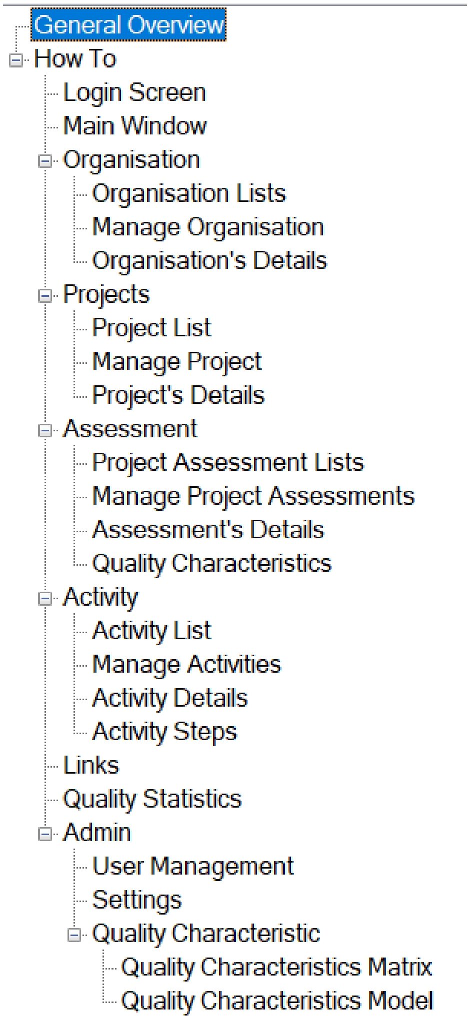

Figure A9.1. Help Window and main menu of IPMS

When the user creates a new project, he enters the project classification data.

The application identifies the appropriate quality model according to the type of project that the user selects. Then user creates new assessment and the application load the quality characteristics factors of the model. The user inputs the assessment characteristics values or import from external Excel file. The application calculates the assessment score. The application analyzes the project data with the assessments score and display the project quality graph.

BD structure, main menu, and some applications print screens are displayed below:

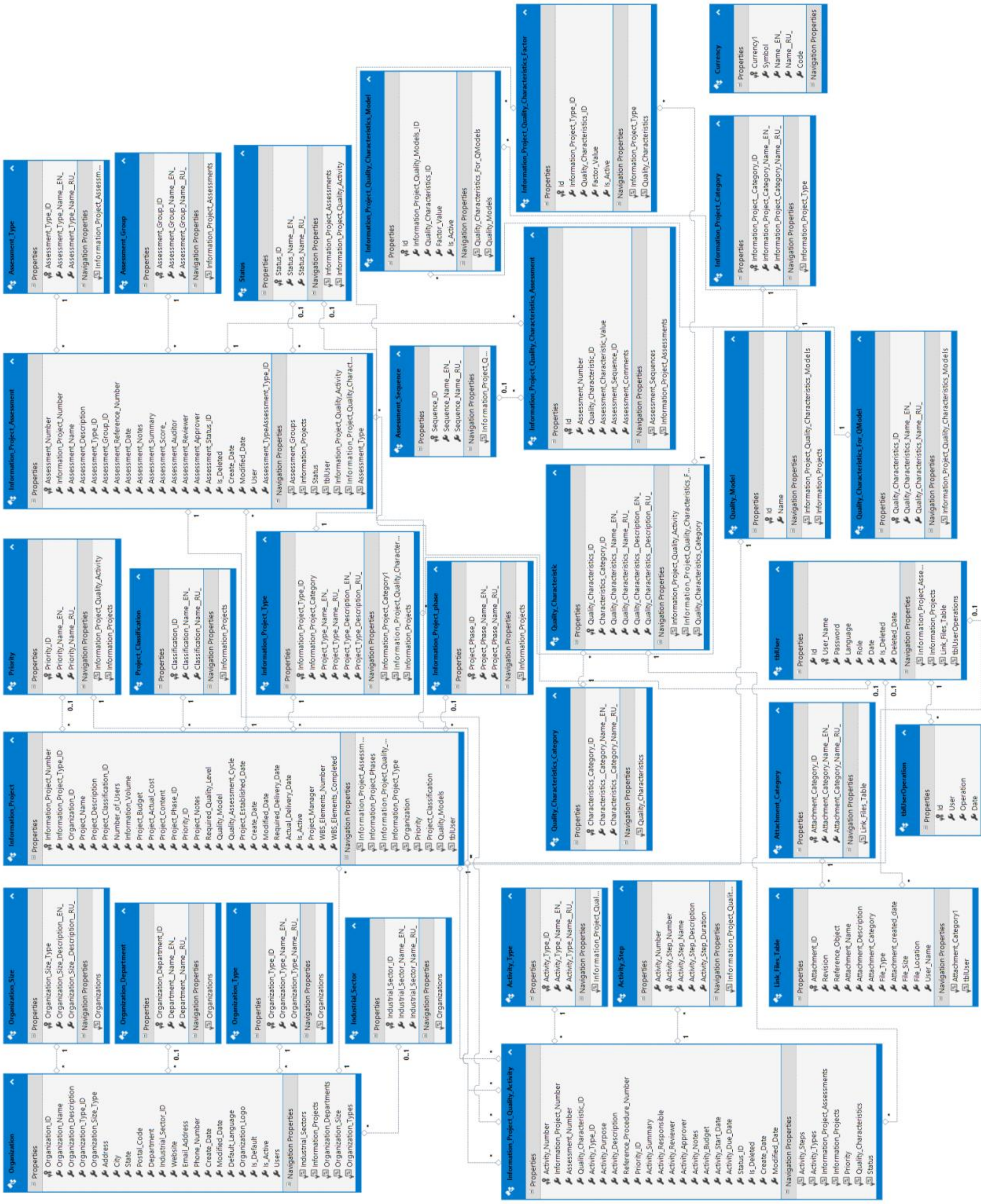

Figure A9.2. IPMS Entity Relationship Diagram

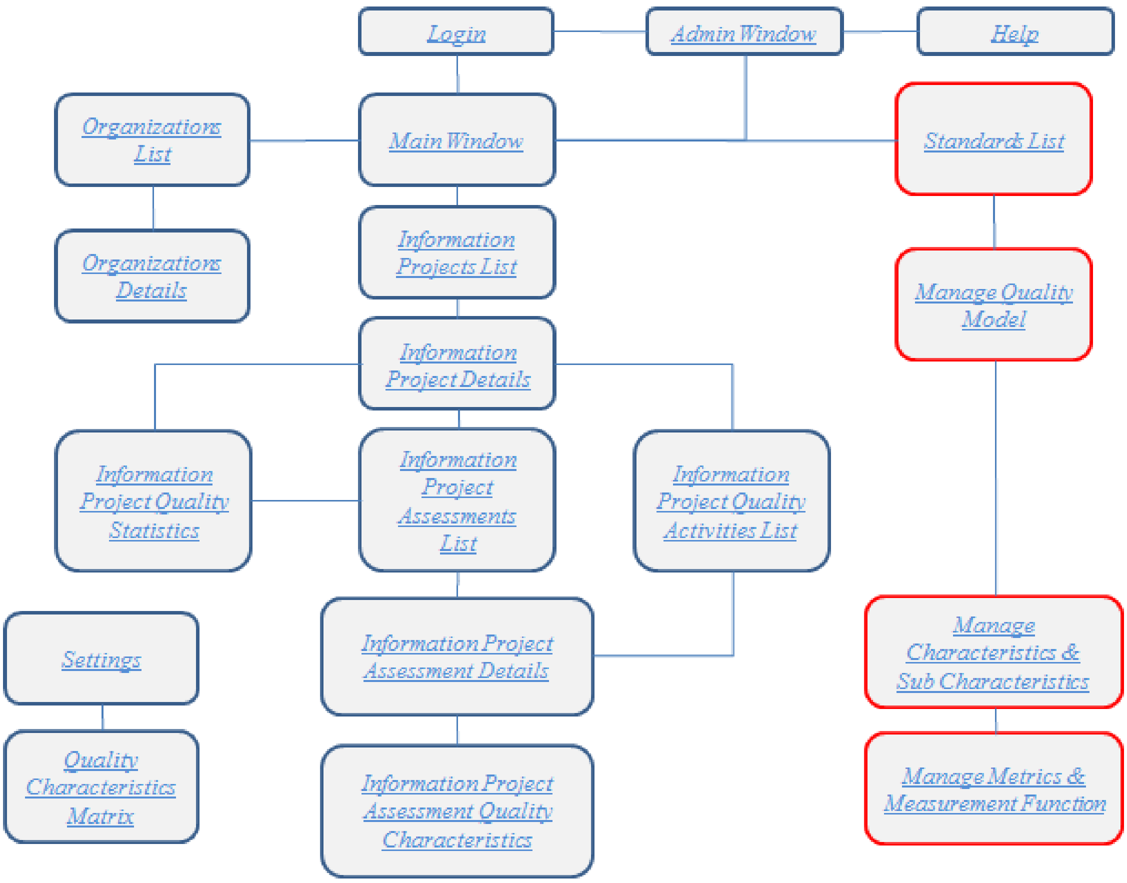

Figure A9.3. Main Screens Wireframe

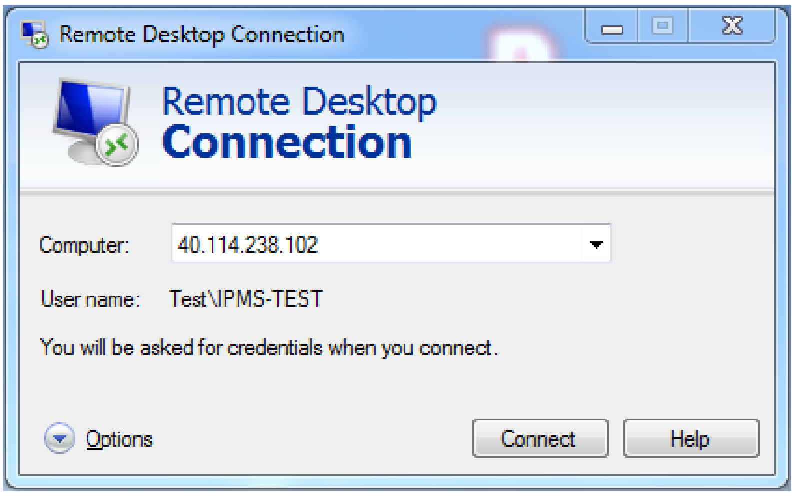

Figure A9.4. Remote Desktop Connection to launch the IPMS application on the cloud

To access IPMS application, send an e-mail to the author's address to receive information regarding the updated Login and Password.

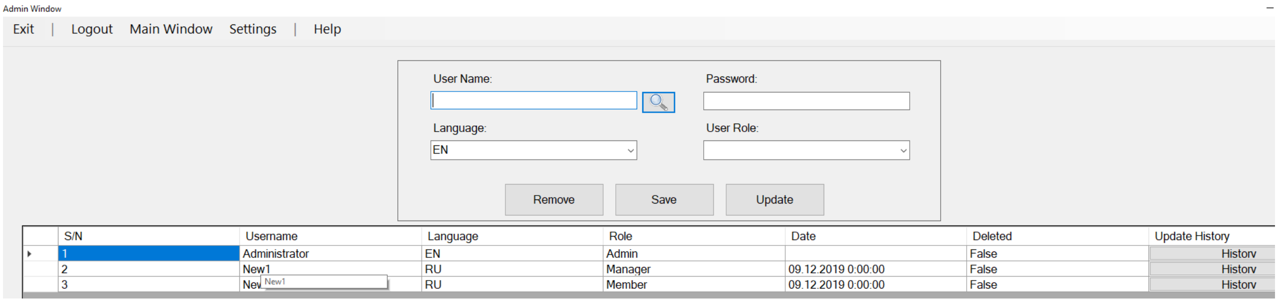

Figure A9.5 Admin window

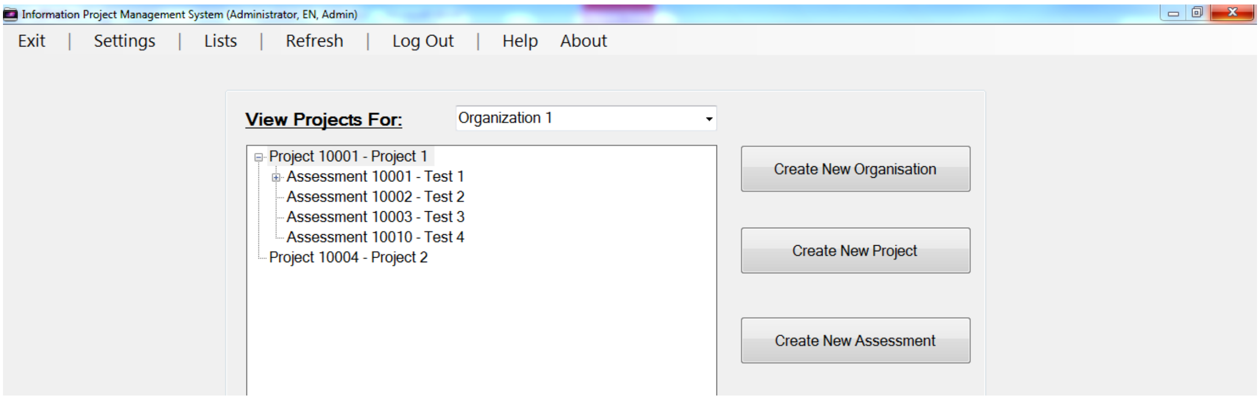

Figure A9.6 Main menu window

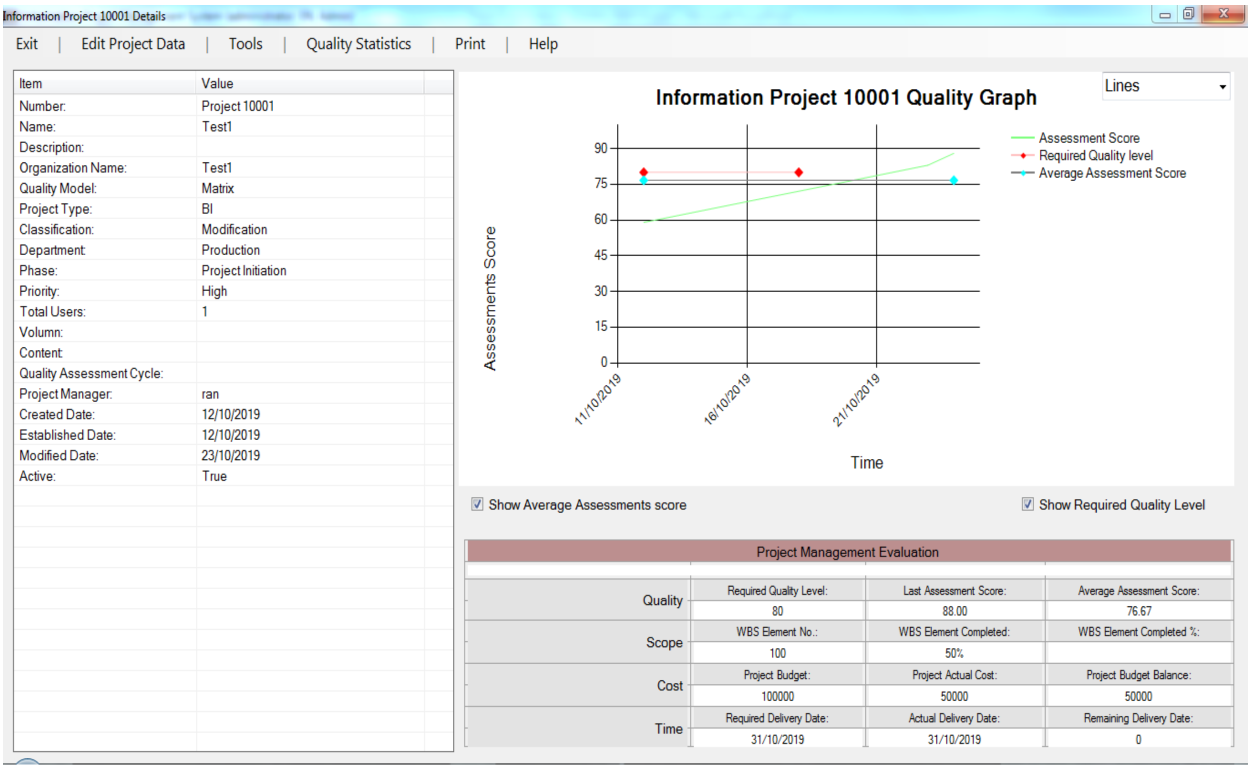

Figure A9.7. Information project details

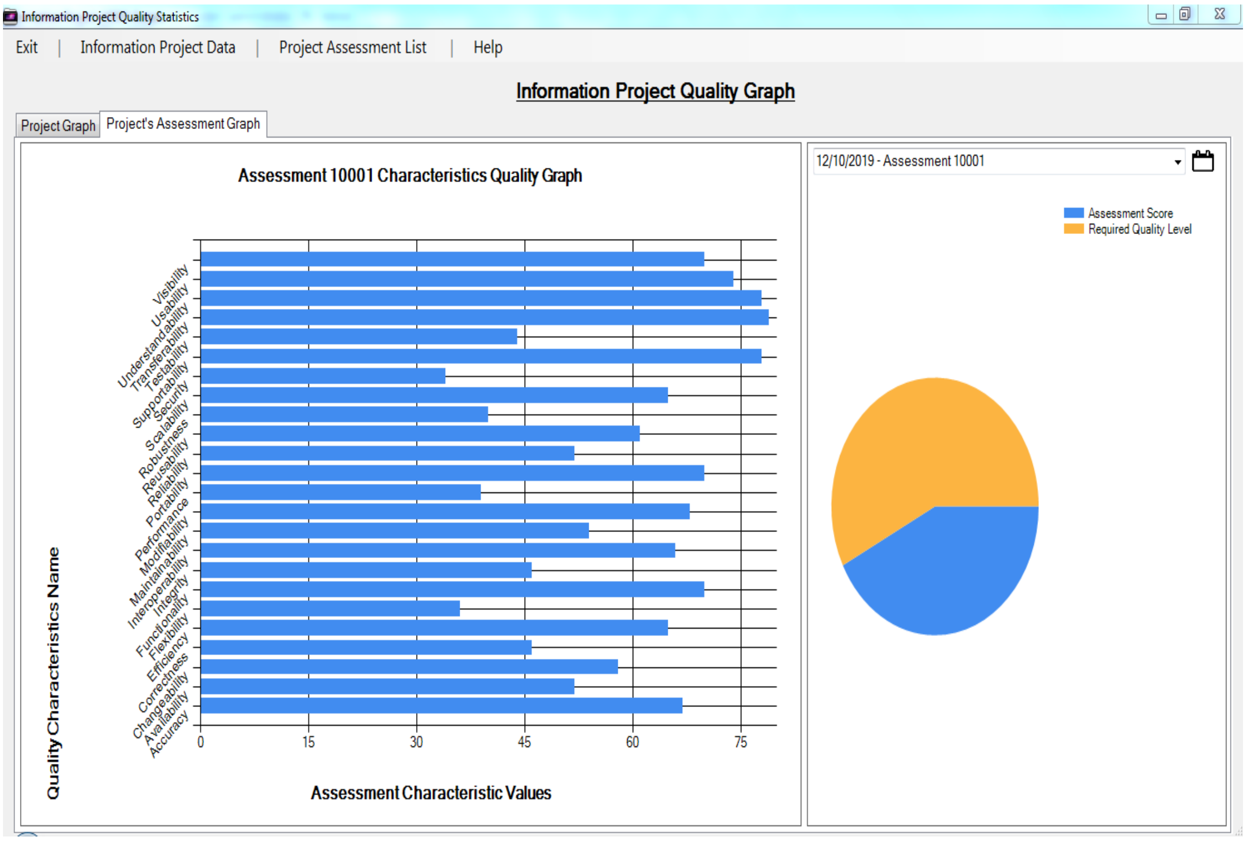

Figure A9.8. Information project quality statistics

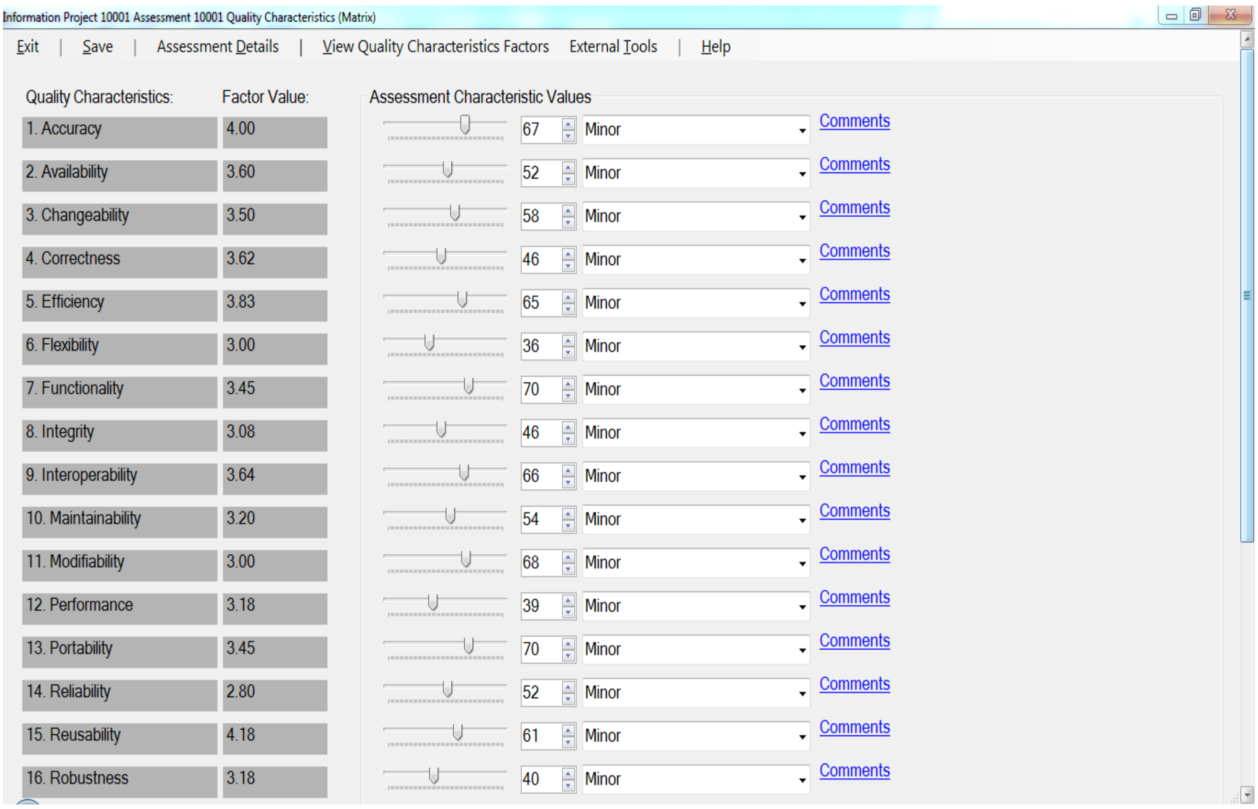

Figure A9.9. Value of assessment quality characteristics

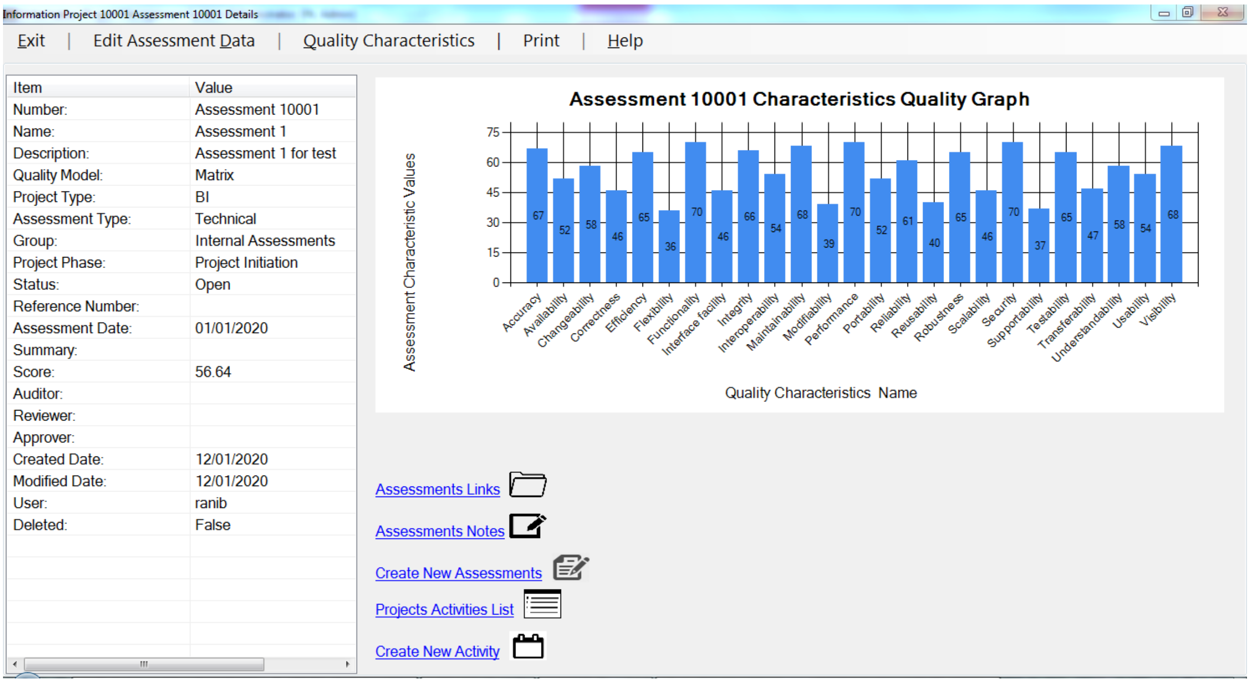

Figure A9.10. Information project assessment details

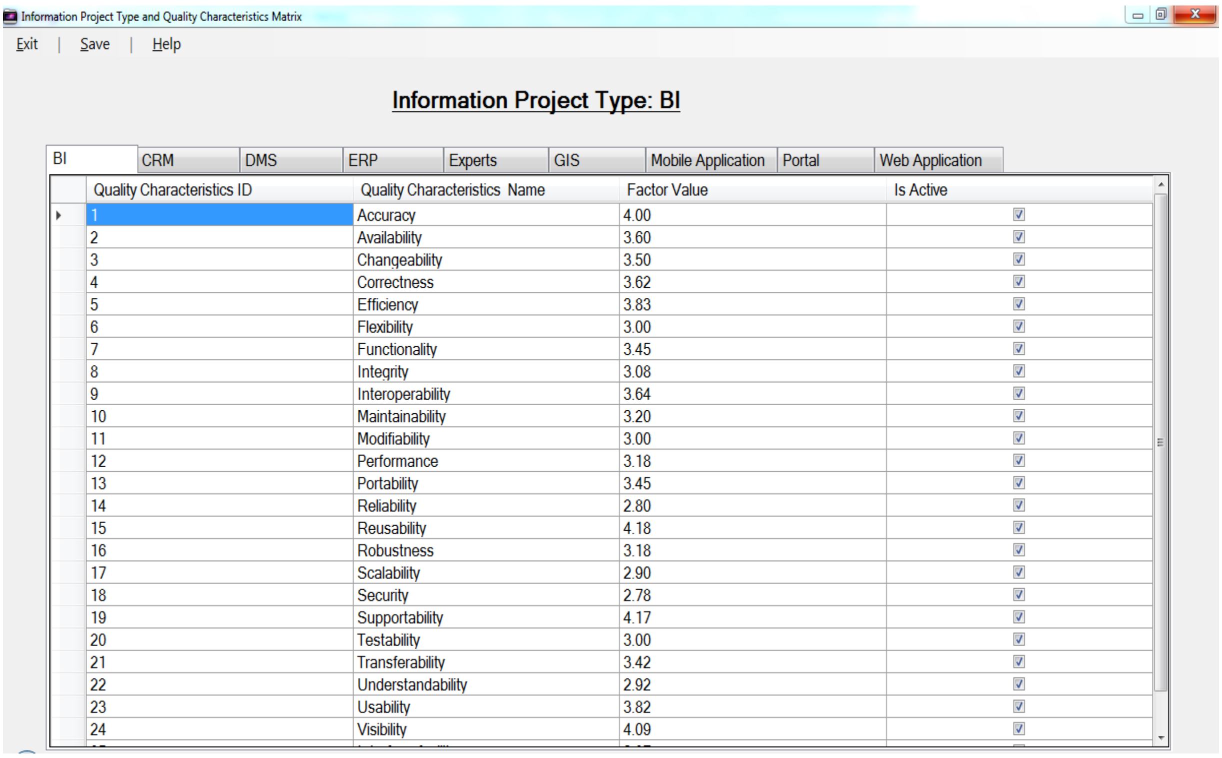

Figure A9.11. Information project type and quality characteristics matrix

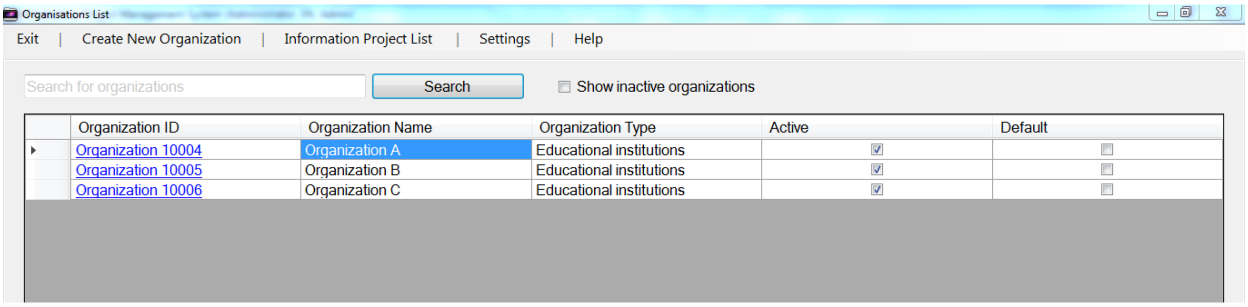

Figure A9.12. Example of Organizations list

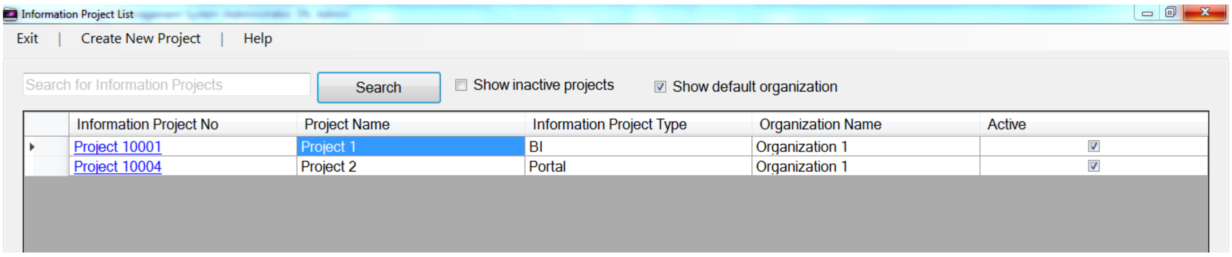

Figure A9.13. Example of Information Project list

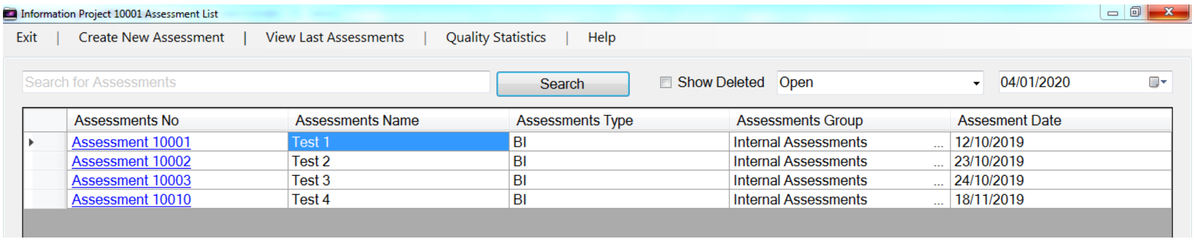

Figure A9.14. Example of Assessment list

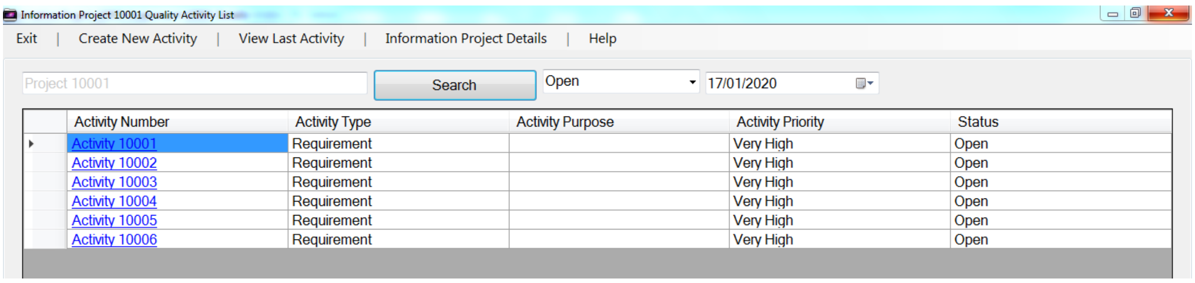

Figure A9.15. Example of Quality Activity list

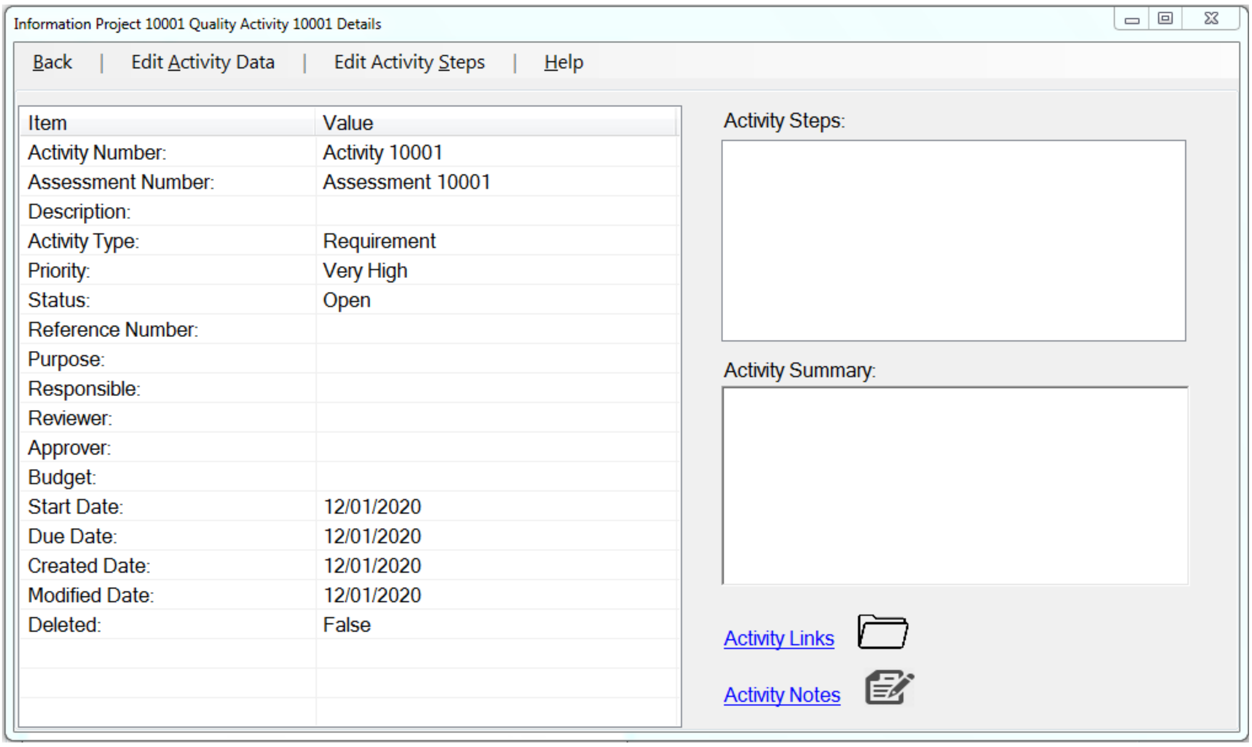

Figure A9.16. Example of Quality Activity details

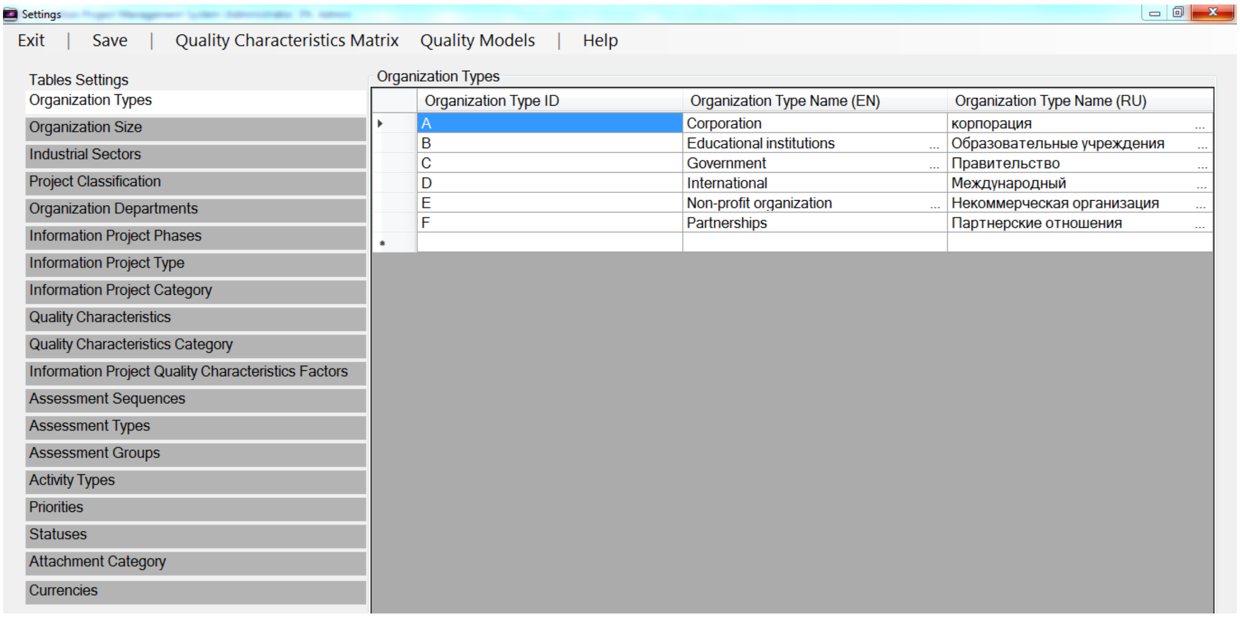

Figure A9.17. Tables Settings

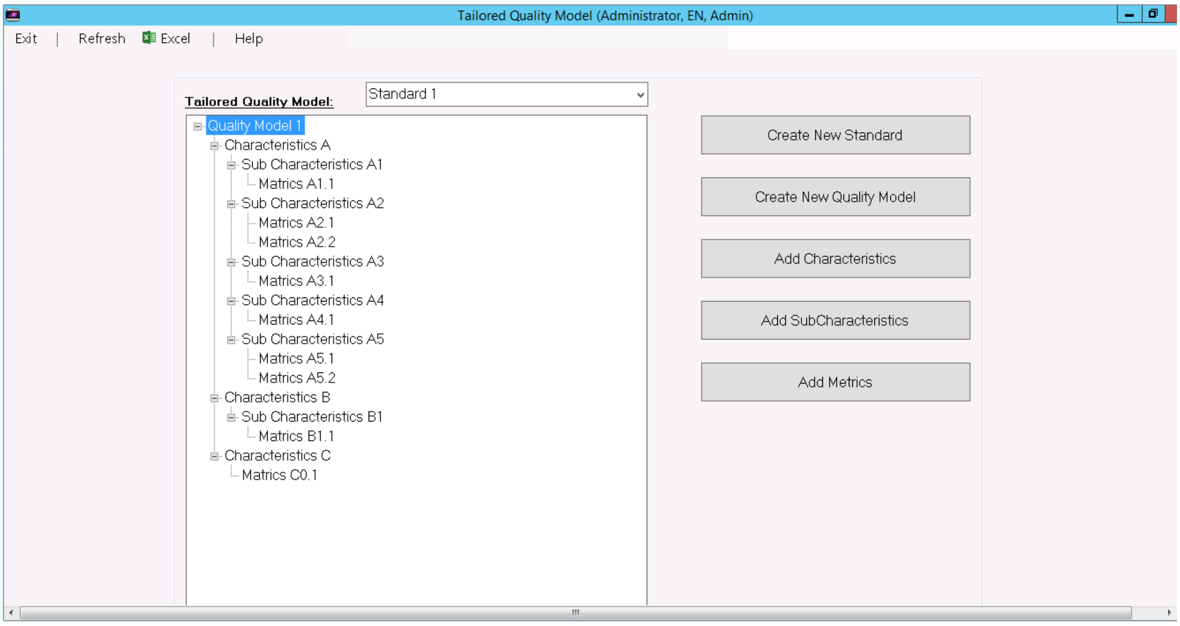

Figure A9.18. Example of Tailored Quality Model