

## Article

# Development of a Maturity Model for Software Quality Assurance Practices

Ahmad Al MohamadSaleh and Saeed Alzahrani \*

Department of Management Information Systems, College of Business Administration, King Saud University, Riyadh 11587, Saudi Arabia; 442106196@student.ksu.edu.sa

\* Correspondence: salhariri@ksu.edu.sa

**Abstract:** The advancements in the technology landscape and software development in recent years mandate paying attention to Software Quality Assurance (SQA) because it is becoming significantly important and complex. SQA is a set of activities within the software development lifecycle that aims at reducing development and testing costs, improving the quality of the software systems, and increasing customer satisfaction. Thus, the objective of this paper is to build a SQA maturity model, particularly in the telecommunication industry. To achieve this, this research identified perspectives and factors based on a comprehensive literature review and experts' inputs using Hierarchical Decision Modeling (HDM) as the methodology. The proposed model consists of five perspectives, which are requirements validation, testing, software change management control, technology, and organization and culture with every perspective containing relevant factors. The factors and perspectives are validated and quantified using SQA inputs from subject matter experts. The findings of this study suggest that requirements validation is the most important perspective. Two case studies were conducted to identify the maturity score for each case, demonstrate the practicality of the research model, identify areas of deficiencies, and propose corrective actions. This paper provides an in-depth look at software quality factors and their relative importance, targeting to help SQA practitioners understand and assess their SQA practices.

**Keywords:** software quality assurance; maturity model; Hierarchical Decision Modeling



**Citation:** Al MohamadSaleh, A.; Alzahrani, S. Development of a Maturity Model for Software Quality Assurance Practices. *Systems* **2023**, *11*, 464. <https://doi.org/10.3390/systems11090464>

Academic Editor: William T. Scherer

Received: 16 August 2023

Revised: 1 September 2023

Accepted: 3 September 2023

Published: 5 September 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Developing successful software systems requires paying close attention to software quality assurance (SQA). SQA encompasses various activities that are carried out throughout the software development process. When these activities are performed well, the outcomes are software systems that meet requirements and expectations with quality and, in return, greater user satisfaction. There are various benefits of developing well-established SQA practices such as reducing development and testing costs, improving the quality and productivity of the software systems developed, and increasing customer satisfaction [1]. Organizations pay close attention to software development as high-quality products meet customer needs, protect their reputation, and improve business operations [2]. A recent study suggests that the success rate in software projects is 37%, in which further efforts should be placed during software development [3]. SQA activities include planning, defining and implementing quality standards, reviewing and auditing, testing, and reporting on the outcomes. Requirement validation is considered the cornerstone, with prototyping, inspection, knowledge-oriented, test-oriented, modeling and assessment, and formal models as important elements [4]. The trend in validation techniques is toward solutions that combine machine-learning techniques with knowledge from dictionaries and ontologies [4]. Machine learning has been used for quality risk assessment and error detection [5]. The majority of difficulties in software development revolve around how to express requirements and respond to client feedback. Quality assurance involves other areas such as SQA

techniques, tools, programming languages used, professional demographics, certification, capabilities, and education [6,7]. Other studies suggest that software systems are evaluated based on their efficiency, effectiveness, compatibility, and traceability while factors such as security, availability, performance, conceptual integrity, and usability may go unperformed [8]. Another study looked at the new product delivery and new service delivery based on TQM, which is divided into four categories: strategic, tactical, operational, and product with the goal of tracking the progress of software development in order to achieve the desired outcomes [9]. The literature suggests that the majority of issues are related to human factors and that there are significant knowledge gaps due to the rise of the IT market, technological advancement, and the rising number of individuals with non-IT backgrounds [7]. Various tools and standards have been developed in recent years to cope with the advancements in software development such as agile delivery methodologies and white box testing approaches [6,10].

This research has examined SQA practices and models as the foundation of this research. At first glance, this may imply that business divergence from an industrious background would result in a variety of points of view and a disjoint of quality needs and objectives. For some organizations, it is either jurisprudence or partial effort of the software quality aspect. For many reasons, some organizations lack knowledge and awareness; for other organizations, due to severe competition and product delivery, they may have to skip many steps or only apply the minimum; for others, the main barrier is applying SQA as a standalone function [11]. In some instances, the SQA team may only emphasize the development scope but overlook overall activities in the entire software development life cycle [2]. Lastly, for some organizations, a lack of resources would consider quality assurance a luxury, and a preliminary level of quality practices would be sufficient. The authors of [12] support the fact that, in IT overall, there is uncertainty when it comes to quality assurance of the return of investments of quality factors such as investing in automation testing. As a result, there has not been enough literature and models investigating the SQA practices and models, both comprehensively and in a structured way. The literature review of this paper provided a wide overview of SQA, and where it is standing in both the point of view of the research and the industry. It is noticeable that most software quality application practices focus on one area and disregard others, such as completing the technical aspect and neglecting the human aspect such as leadership or focusing on the standard or the process and not taking testing metrics seriously. We promote this as the research gap and our expected outcome was to deliver a model that can assess SQA maturity, which will cover various aspects and perspectives. In this research, we have considered the telecommunication industry as the main segment and the lens of this study. Thus, the goal of this research is to develop a structured and comprehensive model to assess the gap, help telecommunication companies understand the most important factors, and pinpoint areas where they need to focus their effort to improve their SQA practices' maturity. The foundation of this research is based on an extensive literature review and the real-world experience of SQA experts, demonstrating the gap of not all quality practitioners being unified or, perhaps, coming to the same standing when it comes to perusing SQA. The expected outcome of this research is to structure a maturity model for SQA that can be used in assessing SQA practices in the telecommunication industry.

#### *Paper Organization*

The paper is organized as follows: the literature review is the starting phase of this research framework. Secondly, the methodology is presented. Following this, an initial model based on the literature has been structured, followed by the formation of expert panels that include a variety of experts' backgrounds and knowledge bases. Then, the model development phase includes validation and quantification. The paper will then test the model against case studies to examine its practicality. Finally, the result will be analyzed and the findings showcased.

## 2. Related Work

The literature review covers the SQA topics in both theoretical and practical aspects, which includes quality assurance standards, testing techniques, testing tools and platforms, quality assurance metrics, quality assurance models, quality assurance process, procedures, and practices. Also, it has been extended to cover maturity models.

### 2.1. Software Quality Assurance Overview

There have been various models and standards that have been researched in the literature. ISO standards have been noted as a major foundation on which researchers have built their studies in various areas and perspectives of software quality [13,14]. ISO has defined the metrics for SQA. Based on these metrics, researchers have built models and studied software quality to comply with these software quality ISO metrics. In [13], the author states that modern software development increasingly relies on software testing to regularly deliver high-quality software, based on the authors' systematic literature reviews. The author has created a complete model for capturing test case/suite quality dimensions that are important from a variety of perspectives based on the ISO/IEC 25010:2011 version. The test artifact quality model given in the study, as suggested by the author, can be utilized in practice to enable test artifact quality assessment and improvement. Furthermore, the model can be used to record context characteristics to make study findings more accessible to academics and practitioners. The demographics and diversity of the IT market and industry can have a market-specific impact on ISO quality adoption. In [14], the authors have spotted the Chinese market in the quality assurance study as a reflection of the rapid growth that has taken place in the Chinese economy and the massive opening to both promising potential and challenges at the same time, due to the huge population count with higher growth, which increases the demand for technology and services. In China, ISO/IEC 25010:2011, Systems and software engineering: Systems and Software Quality Requirements and Evaluation (SQuaRE), is in use, which is a localization due to the need for Chinese market-specific standards and compliance. The Chinese local model promoted the need to reexamine the software code, which ultimately proved successful in ensuring the quality of the software with higher quality assurance. The author also advised considering more scientific research, the definition and improvement of quality standards, and the need to further explore new testing approaches and automation software testing tools, along with improving testing process mechanisms. On the other hand [15], the author has examined the Indian version of ISO 9000's software engineering policies, processes, quality assurance procedures, and measurement techniques. Even though each of the associated operations has been computerized, the author is highly considering the tester's ability to remain a requirement for reasonably accurate solutions.

The agile delivery method is becoming very trendy in modern software engineering practice and product developments [6]. Agile can support late and urgent ad-hoc delivery with limited time and cost because agile offers more visibility of the developments in case any risk or issue occurs halfway through the development of features or deliverables, since feedback can be detected at the earliest and fixed at that time. Agile can improve team communication and collaboration in large organizations to eliminate any gaps or misalignments. Agile is an enabler for the five dimensions of project development, which are organizational, personal, technical, process-oriented, and project-oriented. In addition, agile would significantly influence software quality due to the implication of teamwork practices, engineering practices, documenting practices, management practices, and testing practices. One of the best-known agile processes that contributes to ensuring quality is Scrum. This process can offer a high welcome to changes of the requirements on late and short notice to keep competitor advantage toward the customer, while also paying attention to technical excellence and quality enhancements.

In [16], the authors studied quality assurance from a process and practice viewpoint and believe that the agile methodology will help with software quality. The author studied and examined five aspects of software quality (quality model, quality attributes, quality

metrics, critical success factor, and agile practices) in order to validate what are the key success characteristics for an agile development project, what measures are utilized to assess the quality of software, and which agile principles were utilized by the investigated models. The results show that agile characteristics are most closely related to quality attributes such as “maintainability” and the ISO metric, offering great support for testability and reusability in the development process. This aligns with the finding of [17]. The authors underline that software processes are continuously calibrated and enhanced for better performance using software process improvement methodologies. It includes improving the quality of the software product, decreasing time, and reducing changes, among other things. Lean software development supports process improvement by detecting process waste and evaluating interactions in the software development process. The authors of [18] emphasized that in the new software development industry, a mix of agile methodology and automation testing in the software quality process has resulted in what is known as the Agile Genome, which may provide full use of agile methodologies as well as automation technologies to enhance the finest outcome of quality delivery in the software industry. However, not every segment can fully benefit from a given quality assurance method and technique; it may be narrow in scope. Further study can be conducted to investigate the most frequent quality assurance procedures in order to make them universal and applicable to a wide range of industries. Also, as indicated by [19], Development and Operation (DevOps), by its definition, is a quality-oriented approach that takes culture, collaboration, automation, measurement, and monitoring as enablers for software quality. DevOps is mainly focused on using automation and increasing feedback to reduce the number of resources involved in the release, which should reduce the failure rate. The analysis showed that DevOps features have a stronger impact on the quality and success of software development and offers fast feedback to developers which in return provides better quality. This analysis shows stronger links between DevOps software architecture and quality assurance.

There are many factors to consider when using technology for SQA, mainly the technology setups that can be in alignment with standards and practices. Several concepts that would fulfill the quality factors can be considered. For instance, infrastructure setup is an important element where it requires a developer environment (DEV) for the developer’s tasks, User Acceptance Test (UAT) for pre-release testing to see how clients interact with the software, and a pre-production environment to be in high sync with production. Taking up this measure will fulfill maintainability, portability, and so on. Putting automation in place will be an enabler here. Automation tools are gaining popularity, and they are trustworthy in pursuing quality, where the reputation exclusion of the test cases is no longer the case, due to the high coverage of probability in both happy and negative scenarios. Also, a performance tool can be used that can run a large number of concurrent transactions that measure performance and compatibility. This comes in alignment with the finding of [12]. According to the authors, automation testing is challenging to achieve in organizations, with only 6% of professionals believing it is possible. Executives are looking forward to automating the execution of test cases in order to save money, time, and boost reliability by discovering defects early on. In conducting the survey, the author focused on the return on investment in automated testing, taking into account the following criteria: test coverage, test efficacy, test strategy, tools, cost of test automation, and people, which were distributed to professionals from various product and service organizations. According to the findings of this study, testing accounts for 28% of total IT spending for the last five years due to product and service management, as well as the ability to automate the appropriate test [12]. According to the study, unit testing with adequate automation should deliver technical richness that provides faster, cleaner, and higher quality outputs with very minimal maintenance costs. In [20], according to the author, white box testing approaches have grown in recent years to automate and plan the design and execution of test cases. The author investigated the search-based technique, which provides an opportunity to test the explosion of conditions and pre-condition failures. However, the majority of the research



work has gone toward unit testing and integration testing, which defines the requirement to solve technical problems when transitioning from the unit level to the integration level. Based on the above literature review, we can conclude the three dimensions of our study: First, SQA factors and standards; second, SQA in using management and delivery methods; and finally, SQA in using technology.

## 2.2. Maturity Model's Overview

Maturity models are vital tools that help organizations evaluate the effectiveness of various aspects, and they can be applied to improve business processes. The authors of [21] outline that maturity models are representations of an organization's ability to manage continuous improvement in a particular discipline to achieve growth and further prosperity. There are several maturity models outlined in this paper, and they are vital for several reasons. First, they allow organizations to assess and evaluate their internal performance to adopt the best practices that can be implemented to close performance gaps and improve maturity. Secondly, maturity models are important because they help organizations striving for competitive advantage and identify areas of weaknesses to attain their business goals. Also, maturity models assist organizations in making better and well-informed decisions because they determine what resources are needed to move from one level of maturity to the next. As this study investigates the maturity of the SQA practices, we shed light on the maturity models applied in the technology and software spaces.

### 2.2.1. Business Process Maturity Model (BPMM)

The business process maturity model (BPMM) is considered an effective framework to help organizations achieve success. According to [22], BPMM is a framework that assists an organization in comparing its current practices against the market's standard, allowing it to initiate plans for improving its business procedures. By using BPMM, an organization can efficiently manage its procedures, allowing it to accomplish its set targets and objectives. The business environment is constantly changing, and organizations must innovate and improve their business processes to keep up with these changes. The BPMM framework helps an organization implement a business strategy that enhances the work performance and productivity of its workforce. Most unsuccessful software applications cost organizations significant resources, and these failures are often associated with technology issues. However, the main cause of these failures is inefficiencies in the business processes. The BPMM was developed to eliminate flaws in business processes by allowing organizations to attain uniform standards, establish standardized processes, and determine weaknesses in workflows. BPMM was developed using the process maturity framework (PMF) [22]. Regardless of these benefits, BPMM has its limitations. There are limited empirical studies that confirm the usefulness and validity of BPMM [23]. As a result, the applicability of the model may be limited since it may not meet the security standards imposed by some organizations. Besides that, the model does not consider certain requirements that some organizations might have in place, such as data security.

### 2.2.2. Capability Maturity Model (CMM)

CMM is a framework utilized to refine and develop software advancement procedures. According to [24], CMM outlines a five-stage evolutionary path of maturity and identifies key areas of software development. The framework was developed to evaluate the capability of an organization to develop software, but the model has been applied in other areas, including safety design. The five-stage evolutionary path of CMM helps an organization prioritize its improvement and development efforts. When an organization accomplishes each level of maturity outlined under the CMM framework, it establishes a different aspect of the software process. Through CMM, an organization increases its process capability. Organizations can implement CMM to attain their goals and objectives for cost, schedule, quality, and functionality. The paradigm of CMM encompasses a five-level developmental path. The first level is the initial level. According to [25], most procedures at the initial

level are not recorded and are changed constantly. The second developmental stage is the repeatable level where organizational processes can be repeated and tend to provide steady results. The defined level is the third developmental stage and all processes in this level are defined and recorded. The fourth developmental stage is the managed level where organizations tend to have significant control over their processes and can monitor them. The last developmental stage is the optimizing level in which organizations tend to implement continuous improvements to their processes by monitoring their activities and developing new and creative processes that enhance organizational efficiency. CMM is beneficial to an organization since the framework allows it to identify problems in process development operations. Also, the model allows an organization to reduce the cost of software development since it facilitates the efficient management of data. Regardless of these benefits, CMM has its limitations. When an organization implements CMM, it evaluates each level as a target [25]. In doing so, the organization can lose its perspective which is to improve its processes, by becoming fixated on reaching the next level. Another drawback of CMM is that it does not outline a specific method of attaining its set targets. Just because one organization has achieved success by implementing CMM, it does not guarantee that the framework will be successful since other factors are involved.

#### 2.2.3. Information Quality Management Maturity Model (IQM3)

IQM3 is applied to an organization's information systems to evaluate its level of information quality management maturity. According to [26], the model identifies the most critical information management processes (IMPs) of an organization, especially those processes that cause problems that impact the performance of the organization. Once these critical IMPs have been identified, they are assessed using the Methodology for the Assessment and Improvement of Information Quality Management (MAIMIQ), which is based on the continuous improvement idea. The main aim of MAIMIQ is to address the gaps between the IMP under evaluation and the ideal IMP outlined by IQM3 [26]. In doing so, the methodology improves the information quality of an IMP and requires higher maturity levels. IQM3 is developed using several maturity levels that address a specific information quality management goal and is used by organizations to improve the quality of their information, enhancing their business performance. The limitation of the model is that it is too complex since various key performance areas must be satisfied for it to work efficiently [27]. Another downside of the model is that various techniques related to human resources management and coaching may be required.

#### 2.2.4. Complex Product Systems (CoPS) Maturity Model

According to [28], CoPS are vital projects and they represent a significant amount of annual capital investments in most countries. Various competencies, skills, and capabilities are required in CoPS, including business and innovation capabilities, among others. Under the CoPS maturity model, business capabilities are essential because they allow an organization to identify and solve strategic problems, exploit new technologies, and develop a business plan. The other requirement of CoPS is innovation capabilities [28]. CoPS maturity model enhances the strategic networking process and improves innovation across organizational boundaries. With innovation capabilities, an organization using the CoPS maturity model can manage vital environmental and technological changes and increased global competition. The CoPS maturity model allows leaders to improve their leadership skills, enabling them to adapt effectively to environmental and technological changes that impact their organizations. In addition to improving leadership skills, the CoPS maturity model allows project managers to be able to manage change in complex environments [28]. Regardless of these benefits, the model has some limitations. One of these limitations is that project managers using the model must be knowledgeable to create teams that can work innovatively and productively toward achieving set targets in changing environments. Also, some other project manager capabilities such as delegation skills, among others, are necessary for CoPS projects.

#### 2.2.5. Organizational Change Readiness Maturity Model (OCRMM)

OCRMM outlines the level of capacity that an organization has to change. According to [29], the model allows managers and other involved people to assess an organization's level of maturity and the organization's capacity to implement new practices either through technological or structural change. The model allows organizations to measure their readiness when it comes to accomplishing their organizational goals through structural and technological improvements. In the current and contemporary business environment, organizations are facing uncertainty and complexity on a larger scale than before due to constant changes in market dynamics. Despite these constant changes in market dynamics, some businesses operate as usual without implementing strategies to manage the change [30]. Thus, OCRMM is developed within organizations to help them adapt to market dynamics as well as manage competitive challenges. Through OCRMM, organizations can become agile, allowing them to counter various challenges. The reason is that OCRMM facilitates informed leadership and coordination among the entire workforce. Since the model describes various practices, behaviors, and processes of an organization, it can be used by organizations to manage change and attain sustainable outcomes [29]. Lastly, the model is developed by establishing a strong organizational culture, effective leadership, monitoring of changes, and implementing change initiatives founded on the collaboration and feedback of the whole organization. One of the limitations of the model is that it may not address all the readiness gaps that are critical to implementing change. On most occasions, organizations are faced with poor change readiness and the model does not outline strategies to address these gaps. Due to this limitation, the model may result in unsuitable change implementations, resulting in poor change efforts, and incomplete realization of the benefits associated with the change.

#### 2.2.6. Service Systems Maturity Model

The service systems maturity model presents organizations with the opportunity to create additional value by merging smart products with smart services. With increasing advancements in technology, organizations are facing many challenges to finding the most suitable solutions for planning [31]. The reason is that these organizations are experiencing many strategic challenges when it comes to minimizing operating costs while fulfilling industrial service demands. In comparison to other models, the service systems maturity model involves the application of various information technology solutions in product and service development. It is beneficial to an organization since it influences its business processes. Also, the model is important to organizations because it allows them to collect ideas for new services. With those ideas, they can implement strategies to improve their business objectives. One of the main limitations of this model is that there is a high risk of collecting low-quality data. Although the model allows for the collection of ideas for new services, low-quality data can negatively influence the effectiveness of the product-service system.

Table 1 shows the differences between various maturity models, highlighting the purpose, benefits, and limitations of each one:

Table 1. Maturity Models review.

Model	Purpose	Benefits	Limitation	Reference
Business Process Maturity Model (BPMM)	<ul style="list-style-type: none"> <li>Developed to eliminate flaws in business processes by allowing organizations to attain uniform standards, establish standardized processes, and determine weaknesses in workflows.</li> <li>BPMM provides organizations with a reliable method for measuring the maturity of business workflows and processes.</li> </ul>	<ul style="list-style-type: none"> <li>Allows organizations to gain greater insights into their current business processes.</li> <li>Allows organizations to define the processes and capabilities needed to improve.</li> <li>Minimizes the cost of transactions since it helps simplify tasks within an organization</li> </ul>	<ul style="list-style-type: none"> <li>The model does not consider certain requirements that some organizations might have in place.</li> <li>It may not be compatible with other organizational frameworks and business processes</li> </ul>	[22,23]
Capability Maturity Model (CMM)	<ul style="list-style-type: none"> <li>Was developed to evaluate the capability of an organization to develop software.</li> <li>CMM is implanted in Org to attain its goals and objectives for cost, schedule, quality, and functionality.</li> </ul>	<ul style="list-style-type: none"> <li>CMM framework allows it to identify problems in process development operations.</li> <li>Allows an organization to reduce the cost of software development since it facilitates efficient management of data.</li> <li>The framework allows an organization to minimize post-release defects.</li> </ul>	<ul style="list-style-type: none"> <li>An organization can lose perspective, which is to improve its processes by becoming fixated on reaching the next level.</li> <li>CMM does not outline a specific method of attaining its set targets.</li> </ul>	[24,25]
Information Quality Management Maturity Model (IQM3)	<ul style="list-style-type: none"> <li>IQM3 is applied in an organization's information systems to evaluate information quality management level of maturity.</li> <li>IQM3 can be used by organizations to improve the quality of their information enhancing their business performance.</li> </ul>	<ul style="list-style-type: none"> <li>IQM3 can improve information quality using MAIMIQ.</li> <li>IQM3 is beneficial as it enables information quality managers to solve specific information quality issues by matching them with the appropriate key performance areas.</li> </ul>	<ul style="list-style-type: none"> <li>The model is complex due to various key performance areas that are needed for the model to work efficiently.</li> <li>It has various techniques related to human resources management and coaching may be required.</li> </ul>	[26,27]
Complex Product Systems (CoPS) Maturity Model	<ul style="list-style-type: none"> <li>CoPS maturity model describes various competencies that project managers need to have to be successful in different industries.</li> <li>The model is applied in project-based organizations that require people's capabilities to handle different forms of CoPS.</li> </ul>	<ul style="list-style-type: none"> <li>CoPS maturity model allows leaders to improve their leadership skills, enabling them to adapt effectively to environmental and technological changes that impact their organizations.</li> <li>CoPS allows project managers to manage change in complex environments.</li> </ul>	<ul style="list-style-type: none"> <li>Project managers using the model must be knowledgeable to create teams that can work together and productively toward achieving a set of targets in a changing environment along with other capabilities such as delegation skills.</li> </ul>	[28]

Table 1. Cont.

Model	Purpose	Benefits	Limitation	Reference
Organizational Change Readiness Maturity Model (OCRMM)	<ul style="list-style-type: none"> <li>The model is developed to deal with changes and to help to adapt to the market dynamics as well as manage competitive challenges within organizations.</li> <li>The model is developed by establishing a strong organizational culture, effective leadership, monitoring of change, and implementing change initiatives founded on the collaboration and feedback of the whole organization</li> </ul>	<ul style="list-style-type: none"> <li>The model allows an organization to assess its readiness to implement change.</li> <li>It presents organizations with the opportunity to foster successful change by achieving broad commitment from all the involved parties, including employees.</li> <li>It allows organizations to identify problems that obstruct the implementation of change management programs.</li> </ul>	<ul style="list-style-type: none"> <li>The model may not address all the readiness gaps that are critical to implementing change.</li> </ul>	[29,30]
Service Systems Maturity Model	<ul style="list-style-type: none"> <li>The model presents organizations with the opportunity to create additional value by merging smart products with smart services.</li> <li>This model is commonly used by manufacturing companies aiming to exploit the opportunities of digital transformation.</li> </ul>	<ul style="list-style-type: none"> <li>It is beneficial to an organization since it influences its business processes.</li> <li>Allows organizations to collect ideas for new services.</li> </ul>	<ul style="list-style-type: none"> <li>There is a high risk of collecting low-quality data</li> </ul>	[31,32]

### 3. Research Methodology

#### 3.1. Hierarchal Decision Modeling (HDM)

In this paper, we propose a maturity model to assess SQA maturity in the telecommunication industry by examining the software quality factors, which were identified based on the literature review that had a rigorous assessment and point of view of the quality assurance practices within the IT industry. Each factor is embedded within perspectives, and the weight of each factor should be evaluated to understand the level of importance in the SQA model [33–35]. In order to address the multidimensionality of developing a robust maturity model of SQA practices, this research uses the multi-criteria decision approach, namely the Hierarchical Decision Model (HDM), to accomplish this task. In 1981, Cleland and Kocaoglu introduced HDM [36]. HDM is used to structure the decision/maturity model into a hierarchy of assessment perspectives and factors that require the elicitation and evaluation of the subjective judgment of the subject matter experts in the software development space [37,38]. Depending on the complexity and logical sequence of the problem, the levels of the decision tree are determined and constructed. It evaluates the factors against each other in order to rank them based on their level of importance to the maturity model and goal of this research using pairwise comparison [37,39,40]. The subjective judgments expressed in pairwise comparisons are used to derive numerical weights to identify the relative importance of each factor and provide a systematic approach to decision making and improve the decision and assessment quality. Pairwise comparisons are performed using a constant-sum measurement scale (1–99 scale) for comparing each two decision factors. HDM assigns numerical values to the perspectives and factors in which each factor will have a global weight and local weight within their respective perspective.



HDM is validated and proven to be a reliable approach in addressing the multi-criteria decision problem such as the one under investigation and has been applied at case from different industries [41–44].

The following are the steps of the methodology applied:

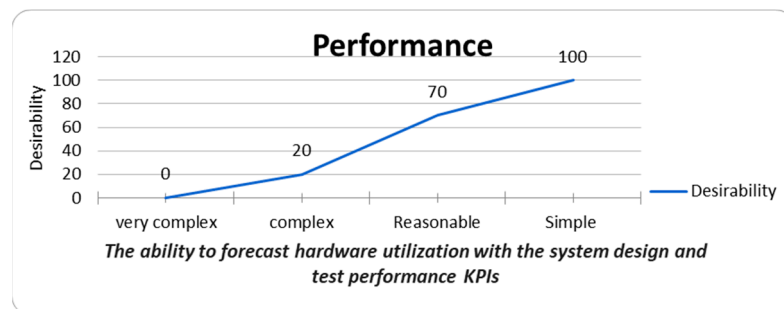
1. With the definition of HDM, we established the model perspectives and factors for assessing SQA practices' maturity. The objective of this research is to develop and maturity model to assess SQA practices. The perspectives identified are the testing perspective, requirements validation perspective, technology perspective, software quality management control perspective, and organization and culture perspective with 25 factors under these perspectives.
2. Create a survey using Qualtrics, first for the validation phase then for the quantification phase to collect the inputs from the experts.
3. Use the Pairwise Comparison tool to determine the weights for perspectives and factors from the survey responses.
4. Considering the relative importance of the various perspectives and factors in determining SQA. The perspectives and factors with the highest weights are viewed as being the most important to SQA. Perspectives and factors are ranked against one another to determine which would be preferred based on experts' points of view and their responses.
5. Develop the desirability curves to understand the dynamics of each factor that shows the different levels where companies can fall into. Desirability curves are used to calculate the final maturity score.
6. Then, we apply the model to assess the maturity of the two well-established telecommunication companies and calculate the maturity levels of each case. Using the factors' weights and the company's weights in the desirability curves, the final maturity score is calculated.

The methodology employed in this research has the ability to fulfill the research objective of developing a maturity model for SQA practices by developing a scoring model that assesses SQA practices' maturity in the telecommunication industry. HDM, as the methodology selected for this study, has the ability to break down complex and multidimensional problems such as the one in this research into smaller and manageable tasks. The HDM emphasizes the incorporation of subject matter experts' inputs in order to provide a dynamic and best representation of reality. It allows for the development of a comprehensive and structured model, ensuring all SQA practice maturity elements are captured and accounted for. This approach provides the necessary tool for telecommunication companies to assess their SQA practice maturity with a score, pinpoint areas of strengths and weaknesses, and decide where to focus their efforts.

### 3.2. Desirability Curves

The desirability curves are used to understand the dynamic of each factor by developing different levels for each factor and determining where companies fall concerning that specific factor. Experts are to identify possible statuses and levels where an organization might fall into based on their experience. Then, the expert assigns values for the typical situations. It helps decision makers understand where they are now and where they need to reach in terms of maturity. Experts are asked to assign a value between 0 and 100 points to each level with the factors for the basis of how desirable the category is. Desirability curves are used to identify how desirable or valuable a metric is for a decision maker. Companies will be evaluated for their maturity and tested using their performance level based on assessing their maturity on the desirability metrics scale. In terms of how companies are going to use the desirability curves to assess their existing situation, the companies will evaluate the company's current situation and capabilities for each factor in the model. Each company can be assigned to a level that best fits each factor. Desirability curves provide significant flexibility to the model where it can help evaluate multiple companies and situations.

Figure 1 below presents an example of a desirability curve for the Performance factor:



**Figure 1.** Desirability curve for the Performance factor.

Further descriptions of the desirability curve definitions and values for each factor are presented in the Appendix A in the end of the document.

### 3.3. Maturity Score

In the case of this research, the maturity score is calculated by multiplying the weight of each factor with its desirability value using the equation below:

$$M = \sum_{k=1}^K \sum_{j=1}^{J_k} P_k \times C_{jk} \times D_{jk} \quad (1)$$

where:

$M$  is the 'Maturity Score',  $K$  is the 'Number of Perspectives',  $J$  is the 'Number of Criteria',  $P_k$  is the 'Weight of Perspective ( $k$ )',  $k = 1 \dots k'$ ,  $C_{jk}$  is the 'Relative importance of Criterion ( $j$ th) for Perspective ( $k$ th) ( $k$ )',  $j = 1 \dots j$  and  $k = 1 \dots k'$ ,  $D_{jk}$  is the 'Desirability value (Maturity Assessment Value) of Criterion ( $j$ th) for Perspective ( $k$ th)' [39,45–48].

### 3.4. Inconsistency and Disagreement

In HDM, the inputs from the experts were tested and validated for input consistency and panel disagreement. Inconsistency in an expert's judgment arises when the expert's judgments or comparisons are not consistent. The inconsistency is calculated using the average standard deviation method and is presented by the HDM software (Version: Beta 2.0). On the other hand, disagreements among experts occur when the experts in the panel show different quantifications to the same analysis [49]. The acceptable threshold for inconsistency and disagreement is 0.10 as indicated by previous studies [47,50].

### 3.5. Expert Panel Selection and Formation

In order for this research to validate and quantify the model elements and to rank the relative importance of each factor, inputs from subject matter experts are needed to validate the model and then quantify the factors using the pairwise comparison method. Experts were identified using direct connections from the academic research community and the professional environment in the IT industry with special expertise in software development. Professional social network channels such as LinkedIn also offer accessibility to experts from several IT organizations, especially from the telecommunication industry. This will add more value to the research contribution and the survey result.

### 3.6. Expert Panel Formation

After identifying all experts, the experts were distributed across multiple panels for participation in validating and quantifying different aspects of the proposed maturity model. The experts can participate in one or more panels. The following Table 2 shows the experts' involvement.

**Table 2.** Expert Panels.

Panel Number	Task	Number of Experts	Instrumentation
Panel 1	Personal Interview	2	Direct interview on the model validation
Panel 2	Validation of the model	19	Online survey (Microsoft forums)
Panel 3	Model perspective level Quantification	23	Pairwise comparison using Qualtrics, analyzed by HDM tools
Panel 4	Testing Perspective Quantification	5	Pairwise comparison using Qualtrics, analyzed by HDM tools
Panel 5	Requirement Validation Perspective Quantification	9	Pairwise comparison using Qualtrics, analyzed by HDM tools
Panel 6	Software change management control Perspective Quantification	10	Pairwise comparison using Qualtrics, analyzed by HDM tools
Panel 7	Technology Perspective Quantification	5	Pairwise comparison using Qualtrics, analyzed by HDM tools
Panel 8	Organization and Culture Perspective Quantification	10	Pairwise comparison using Qualtrics, analyzed by HDM tools

Experts were distributed across multiple panels based on their area of expertise as shown in Table 3 below:

**Table 3.** Experts' contributions across panels.

Number	Expert Designation	Panel 1	Panel 2	Panel 3	Panel 4	Panel 5	Panel 6	Panel 7	Panel 8
1	Sr Quality assurance leader	•		•	•	•		•	
2	Head of Delivery		•	•	•			•	
3	Software Quality Assurance Expert		•	•	•			•	
4	Software Quality Assurance Expert			•	•			•	
5	Software Quality Assurance Expert			•	•			•	
6	Software Quality Assurance Expert			•			•		•
7	Senior ICT Solution Design		•	•			•		•
8	Software Quality Assurance Expert			•			•		•
9	CISCO Service Manager		•	•			•		•
10	CTO		•	•			•		•
11	Solutions Architect		•	•			•		•
12	BSS Director		•	•			•		•
13	Business analyst			•			•		•
14	Digital Executive GM		•	•			•		•
15	ICT expert			•			•		•
16	software Quality Assurance engineer			•		•			
17	software Quality Assurance engineer			•		•			
18	software Quality Assurance engineer			•		•			
19	Sr Business Analyst			•		•			
20	Sr system Analyst			•		•			
21	system Analyst			•		•			
22	system Analyst			•		•			
23	Sr system Analyst			•		•			
24	Sr Quality assurance Engineer		•						
25	IT system analyst		•						
26	Expert Tester		•						

Table 3. Cont.

Number	Expert Designation	Panel 1	Panel 2	Panel 3	Panel 4	Panel 5	Panel 6	Panel 7	Panel 8
27	Sr software quality engineer		•						
28	VAS and integration GM		•						
29	Technology director		•						
30	Regulatory Affairs VP		•						
31	System Integration Manager		•						
32	Architecture and DevOps		•						
33	Applications Operations head		•						
34	Digital products director		•						
35	Digital products director	•							

• The expert participated in the panel.

#### 4. Research Model Development and Results

Based on an extensive literature review of the SQA domain, the model's important factors were classified into five perspectives, namely:

1. Testing Perspective
2. Requirements validation Perspective
3. Technology Perspective
4. Software quality management control Perspective
5. Organization and Culture Perspective

##### 4.1. Proposed Maturity Model

The following Tables 4–8 present the model factors extracted from the literature review.

Table 4. First Perspective: Testing.

Factors	Definition	References
Test artifact	This factor measures the coverage of test cases, test scenarios (Test Suites) to the newly developed function, or applied change to the systems.	[12]
Test level	The mix of the testing levels to reach an accuracy of the implementation and efficiency	[12,51]
Testing objective	This factor measures the degree of precision of the expected result of the applied testing activity to achieve the testing objective, such as acceptance testing, compatibility testing, execution time testing, penetration testing, quality of service testing, regression testing, robustness testing, safety testing, security testing, UI testing, usability testing.	[12,52,53]
Testing activity	This factor measures the organization's application of the needed sequential testing activities and required efforts of the software testing (test case design, test case execution, test case generation, test case prioritization, test case selection, test coding, test data generation, test script generation, test script repair).	[12]
Testing approach	The clear objective of selecting and understanding the testing approach, either block box testing, white box testing, or both, and being aware of the implications and necessity of this selection.	[12,52]

**Table 5.** Second Perspective: Requirement validation.

Factors	Definition	References
Competency	Level of certainty that requirement documents contain all the requirements and updates and their accompanying constraints.	[13,54]
Consistency	The level of the measurements and procedures in place for the required items to prevent contradicting other requirements related to the existing and/or other software features and functions.	[8,54–58]
Correctness	This factor measures the acceptable degree of mutual understanding of the requirements, which implies to be mapped with compliance to policies, standards, and laws.	[13,52,54]
Validity	This factor measures the alignment on how the system functions and what needs to be performed based on what is proposed by stakeholders.	[13,52,53]
Realism	This factor measures the awareness of projects or changes constraints, defining the achievable requirements.	[13,59]
Verifiability	This factor measures the precision level of what the demonstrated and tested has been implemented as per the specified requirements	[13,53]

**Table 6.** Third Perspective: Technology.

Factors	Definition	References
Automation level	The level of automation within the testing methods, with the level of ability to perform automation activity such as test case execution, test case generation, test data generation, test script execution, test script generation and repair as well as the automation degree.	[4,19,60]
Performance	The ability to forecast hardware utilization with the system design and test performance KPIs.	[12,19]
Testing Tools	The availability of testing tools used in test case generation, testing execution, testing tracing, and defect logging.	[12,51]
Framework and environment structure	The level of integration and the sync of the components during all delivery phases (Dev Env, UAT Env, pre-prod Env, and production).	[12,56]

**Table 7.** Fourth Perspective: software change management control.

Factors	Definition	References
Agile	The organization's readiness to adopt agile delivery methodology.	[4,55]
DevOps	The organization's adoption of the DevOps delivery methodology, which offers fast feedback as the main DevOps features	[19]
Release	This factor measures the level of awareness of the relevant stakeholders about the changes and the implementations: it can be fast track or a 4–5 week cycle	[54]
Internal Process	the facilitation level of the internal process, approval, SLAs to the change management.	[54]



**Table 8.** Fifth Perspective: Organization and Culture.

Factors	Definition	Reference
Leadership	The degree of top management support of quality assurance and the role of leaders in enabling SQA	[29]
Team building	The organization's ability to put together a dedicated/appropriate team with efforts and support toward quality assurance activities and approaches, which include the availability of developers, software engineers, testers, and product owners.	[6,61]
Reporting quality	The reporting efforts and level of communication undertaken between stakeholders such as testing reports, defect reports, performance reports, frequency of reports, and management exposure to the reports.	[51]
Documentation	The awareness of the importance of keeping the traceability of the documentation versions among teams, and accessibility to the document's repository and templates.	[55,62,63]
Certification and technical skills	The availability of clear definitions of the required technical skills, with the ability to provide suitable training and quality assurance-related certificates such ISQTB and programming and technical skillset.	[7,64,65]
Quality standard	Does the organization keep up and be well informed about the quality assurance standards and practices such as ISO and ISQTB?	[1,4,11,13–15,57,66]

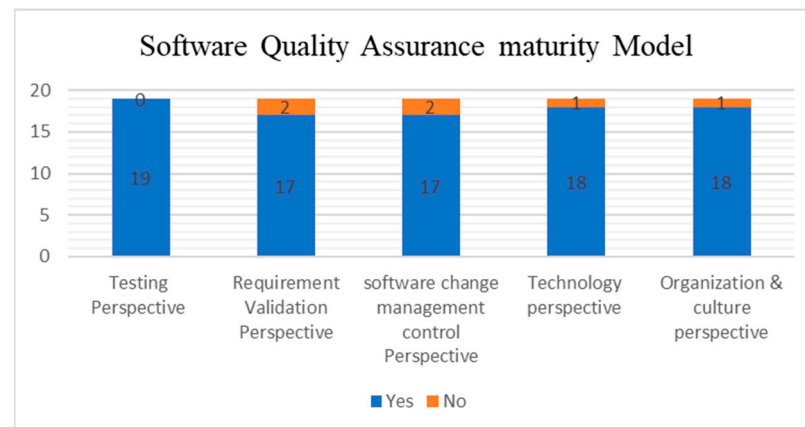
#### 4.2. Result: Validation and Quantification

Subject matter experts were invited to validate and quantify the model construct. A total of 35 experts participated in the validation and quantification phase of this research. Experts were distributed based on their area of expertise across eight expert panels as explained in the methodology section.

##### 4.2.1. Validation Phase

Based on the constructive review of the literature review, we identified the factors believed to serve as the foundation to develop the initial maturity model. This paper then validates the factors throughout the selected experts in the SQA field. If 2/3 of the experts agree that the specific factor is important and should be included in the model, then it is kept otherwise, the factor is deleted. The experts were given the chance to propose new perspectives and factors to be added. In this stage, the goal is to capture the expert's opinions, which will ensure that the model considers the most important perspectives and their embedded factors that can offer the best representation of the reality. The objective of this validation process is to include all the important factors for developing the SQA practice maturity model and ensure a generalizable and reliable model that can be used in different companies in the telecommunication industry. Thus, the researcher distributed the validation survey to 19 subject matter experts. The following section presents the findings.

In the overall model perspectives, the 2/3 threshold of the experts' validations conclude the degree of agreement that testing, requirement validation, software change management control, technology, and organization and culture perspectives are helpful and significant in assessing the maturity level of the SQA practices within their organization. This can be seen in Figure 2 for the perspectives:



**Figure 2.** Result of the Model Validation.

The validation of the factors included in the model and the list of experts who participated in this phase are shown in Tables 9 and 10 below:

**Table 9.** Expert Validation result.

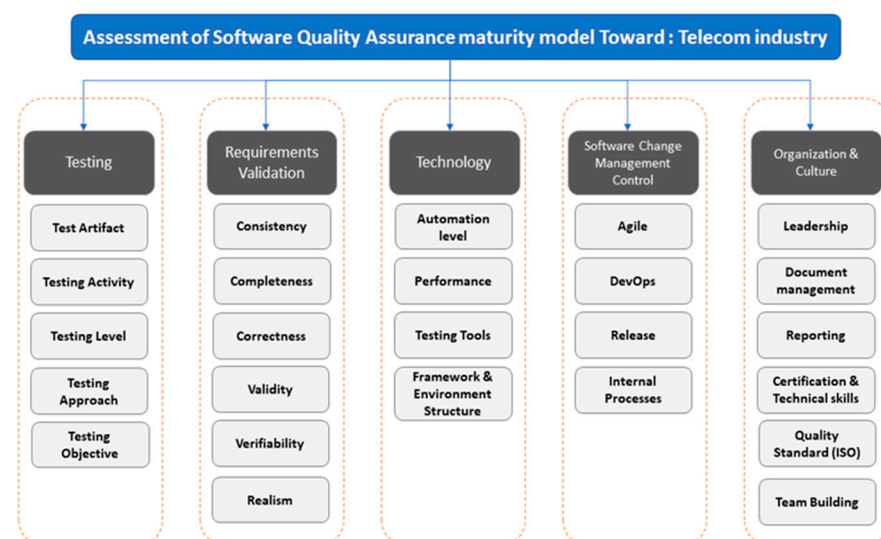
Perspective	Factors	Validation %
Testing Perspective (100.0%)	Test artifact	84.2%
	Test level	89.5%
	Testing objective	100%
	Testing activity	100%
	Testing approach	78.9%
Requirement Validation Perspective (89.5%)	Competency	89.5%
	Consistency	100%
	Correctness	94.7%
	Validity	94.7%
	Realism	73.7%
	Verifiability	84.2%
Technology perspective (89.5%)	Automation level	89.5%
	Environment Performance	100.0%
	Testing Tools	94.7%
	Framework and environment structure	94.4%
Organization and culture perspective (94.7%)	Leadership	94.7%
	Team building	94.7%
	Reporting	100%
	Documentation	73.7%
	Certification and technical skills	94.7%
	Quality standard	89.5%
Software change management control Perspective (94.7%)	Agile	100%
	DevOps	94.7%
	Release	78.9%
	Internal Process	89.5%

**Table 10.** Panel (2): Experts participated in the model validation.

Number	Expert Code	Expert Designation
01	02-01	Sr Quality assurance Engineer
02	02-02	IT system analyst
03	02-03	Expert Tester
04	02030608-05	BSS Director
05	02-04	Sr software quality engineer
06	02-05	VAS and integration GM
07	02-06	Technology director
08	02030608-06	Digital Executive GM
09	02030407-01	Head of Delivery
10	02030608-02	CISCO Service Manager
11	02-07	Regulatory Affairs VP
12	02-08	System Integration Manager
13	02-09	Architecture and DevOps
14	02030608-01	Senior ICT Solution Design
16	02-10	Applications Operations head
16	02-11	Digital products director
17	02030608-04	Solutions Architect
18	02030608-03	CTO
19	02030407-02	Software Quality Assurance Expert

The outcomes of the validation phase are shown in the finalized model. The experts provided constructive feedback that included validating the proposed factors and redefining some factors to meet the objective of the research. All the feedback and comments received from the experts were incorporated into the final model. The following tables show some of the feedback received and the figure below shows the finalized model.

Based on the validation phase, the model has not been changed in structure. However, the experts have provided insightful feedback on the definitions with respect to the telecommunication industry. Figure 3 shows the validated model:

**Figure 3.** SQA Finalized model.

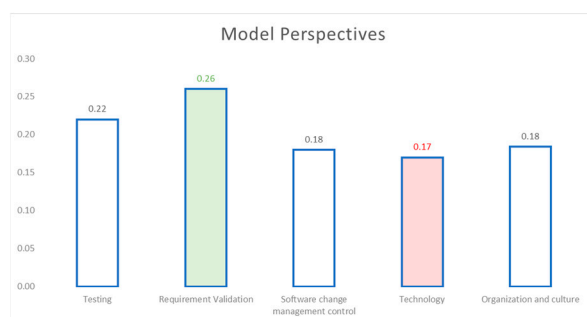
#### 4.2.2. Quantification Phase

The goal of this phase is to define the relative importance of all the model factors within their main perspectives, in addition to applying the assessment of quality assurance maturity. A total of 23 experts participated in the quantification phase. Qualtrics survey is used to capture the expert judgments and they are analyzed using the HDM tool. The HDM tool helps in conducting and analyzing the pairwise comparison against every factor

within the same perspective from the importance point of view of the expert judgment. The outcome of this phase is a ranking of the factors important in the maturity of the SQA practices.

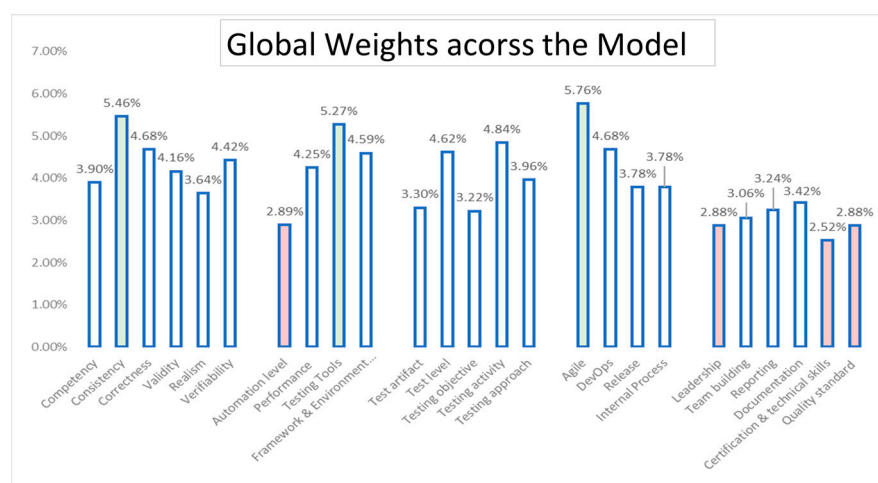
The inputs from the experts were tested and validated for input consistency and panel disagreement. All experts showed consistency in their judgment and all experts within each expert panel show a high level of agreement on the ranking of the maturity factors.

At the perspectives level, requirement validation is the most important perspective to assess the quality assurance maturity within the telecommunication industry, and this indicates that more attention should be drawn in addition to level-up quality. On the other hand, technology was the perspective that has the least influence on reaching quality, meaning that other practices that have more human tendency, such as organization and culture or product management methods, have more weight in terms of software quality influence. Figure 4 shows the quantification results for the perspectives. The green color shows the highest ranked perspective and the red color for the lowest ranked perspective. Figure 5 shows the overall ranking of the factors with the green color for the highest ranked factors and the red color for the lowest ranked factors:



**Figure 4.** Overall ranking of the perspectives.

Final Model Weights:



**Figure 5.** Overall ranking of the model factors.

## 5. Case Studies

### 5.1. Telecommunication Industry in Saudi Arabia

The goal of this study is to identify and rank the factors impacting the maturity of SQA practices in the telecommunication industry and help companies realize the areas of strength and weakness. The developed model is applied to two real-world cases of telecommunication companies, which helps in ensuring the robustness of the research model. The telecommunication industry in Saudi Arabia has grown and developed at a

rapidly increasing rate in recent years. The telecommunication market is dominated by three companies namely: the Saudi Telecom Company (STC), Mobily, and Zain. The competition among these companies has increased and contributed positively to the economic growth of the Kingdom of Saudi Arabia. The government has been putting much effort into boosting the telecommunication industry and companies as well as seeking international growth. This research applied the maturity model at two of these top three companies. Both cases have been developing successful software systems, and there is much potential to improve their maturity in terms of software development. In assessing their maturity, the results of the factors' quantifications and desirability curves will remain constant; however, the two cases will be tested against these results using their performance based on the desirability metrics scale. The goal is to develop a model that can help telecommunication companies assess their SQA practices' maturity and offer recommendations on how to improve their practices. Tables 11 and 12 shows the maturity scores for the cases:

**Table 11.** Maturity Scores for case studies.

Perspective	Factor	Global Weight	Case 1 VS *	Case 1 FS **	Case 2 VS	Case 2 FS
Testing Perspective	Test artifact	3.430%	95	3.26	45	1.54
	Test level	4.893%	95	4.65	45	2.20
	Testing objective	5.168%	95	4.91	75	3.88
	Testing activity	5.031%	80	4.02	20	1.01
	Testing approach	4.390%	70	3.07	30	1.32
Requirement validation Perspective	Competency	3.650%	95	3.47	35	1.28
	Consistency	5.171%	75	3.88	30	1.55
	Correctness	4.289%	90	3.86	85	3.65
	Validity	3.863%	90	3.48	65	2.51
	Realism	3.285%	90	2.96	70	2.30
Software Change Management Control Perspective	Verifiability	4.076%	90	3.67	85	3.46
	Agile	5.253%	50	2.63	20	1.05
	DevOps	4.347%	50	2.17	75	3.26
	Release	3.458%	95	3.29	45	1.56
	Internal Process	3.392%	90	3.05	35	1.19
Technology Perspective	Automation level	2.979%	75	2.23	20	0.60
	Performance	4.504%	40	1.80	70	3.15
	Testing Tools	5.426%	95	5.16	10	0.54
	Framework and environment structure	4.823%	90	4.34	40	1.93
Organization and Culture Perspective	Leadership	2.985%	95	2.84	20	0.60
	Team building	3.148%	95	2.99	30	0.94
	Reporting quality	3.332%	80	2.67	45	1.50
	Documentation	3.435%	75	2.58	10	0.34
	Certification and technical skills	2.535%	70	1.77	20	0.51
	Quality Standards	3.026%	85	2.57	45	1.36
Final Result		100%		81.31		43.22

\* VS: Value Score, \*\* FS: Final Score.



**Table 12.** Case studies performance.

Perspectives	Case 1	Case 2
Testing Perspective	19.91	9.94
Requirement validation Perspective	21.31	14.75
Software Change Management Control Perspective	11.14	7.05
Technology Perspective	13.53	6.22
Organization and Culture Perspective	15.42	5.25
Maturity Scores	81.31	43.22

### 5.2. Results of the Case Studies

- i Maturity Scores
- ii Improvement Simulation

Enhancement scenarios for both cases based on their performance in the maturity scores as proposed are shown in the following table. Companies can follow conservative or moderate approaches or even go all in to make changes. The IT team should realize their existing status and plan where they need to make improvements. Table 13 shows the enhancement areas for selected factors where improvements are needed:

**Table 13.** Enhancements for the case studies.

Perspective	Factor	Global Weight	New VS CS1 Score	Case 1 FS	New VS CS2 Score	Case 2 FS	Recommendations and Improvement Suggestions
Testing Perspective	Testing activity	5.031%	80	4.02	50.00	2.52	Case 2: Needs to achieve at least average-to-high application of the needed sequential testing activities and required efforts of the software testing.
	Testing approach	4.390%	70	3.07	60.00	2.63	Case 2: Needs considerable awareness and a clear objective of selecting and understanding the testing approach, either block box testing, white box testing, or both, with awareness of the implications and necessity of the selection.
Requirement validation Perspective	Competency	3.650%	95	3.47	55.00	2.01	Case 2: Needs to operate at the level where the medium-majority of requirements are documented and available.
	Consistency	5.171%	75	3.88	60.00	3.10	Case 2: Needs to have at least a medium level of the measurements and procedures in place for the requirements items to prevent contradicting other requirements related to the existing and/or other software features and functions.
Software Change Management Control Perspective	Agile	5.253%	80	4.20	60.00	3.15	Case 1 and 2: The ability of the telecommunication companies to perform the test artifact procedure at the highest level should be performed at least at a medium-to-high level.
	DevOps	4.347%	80	3.48	75.00	3.26	Case 1: the DevOps delivery methodology should be utilized and adopted frequently.
	Internal Process	3.392%	90	3.05	55.00	1.87	Case 2: should have at least medium support/facilitation level of the internal process, approval, SLAs to the new implementation and change management.

Table 13. Cont.

Perspective	Factor	Global Weight	New VS CS1 Score	Case 1 FS	New VS CS2 Score	Case 2 FS	Recommendations and Improvement Suggestions
Technology Perspective	Automation level	2.979%	75	2.23	45.00	1.34	Case 2: should improve the level of automation within the testing methods, with a high level of ability to perform automation activity.
	Performance	4.504%	60	2.70	70.00	3.15	Case 1: should have the ability to forecast hardware utilization with the system design and test performance KPIs.
	Testing Tools	5.426%	95	5.16	55.00	2.98	Case 2: should have the dedication to ensure the availability of the testing tools used in test case generation, testing execution, testing tracing, and defects logging.
Organization and Culture Perspective	Leadership	2.985%	95	2.84	75.00	2.24	Case 2: should seek the support and involvement of top management in the quality assurance activities and practices
	Team building	3.148%	95	2.99	65.00	2.05	Case 2: should have a medium-to-high ability to put together a dedicated/appropriate team with efforts and support toward quality assurance activities and approaches.
	Documentation	3.435%	75	2.58	70.00	2.40	Case 2: should have a high level of awareness of the importance of keeping the traceability of the documentation versions among teams, and accessibility to the document's repository and templates.
	Certification and technical skills	2.535%	70	1.77	70.00	1.77	Case 2: should have a high level of availability of clear definitions of the required technical skills, with the ability to provide suitable training.
Final Improved Results		100%		85.09		60.37	

## 6. Discussion

In this section, the 35 SQA and IT Expert responses were utilized. All of them have agreed with a positive level of assertiveness that the proposed model will be useful and helpful to assess the SQA and there is a significant need for such a model due to the fact there is no standing point to pursue software quality end-to-end in some organizations.

As a starting point, experts in the field of SQA have been personally interviewed to enhance the understanding of the SQA practices landscape. The objective of this interview was to determine two preliminary inquiries: first, whether there is a need to develop a SQA maturity model within the telecommunication industry. Furthermore, to initially confirm the usefulness of the proposed model through the full review of the proposed model. Both inquiries come with affirmative responses, confirming the high need to have a maturity model for software quality assessment practices, which the proposed model would help achieve. Then, 19 subject matter experts in the field went through the proposed model, validating twenty-five factors embedded within the five perspectives. The validation phase objective is to confirm the main concept, structure, and model elements as well as ensure that the model provides a great representation of the landscape. The result for this phase confirmed that the model is appropriate and useful for assessing the maturity of the SQA practices.

Furthermore, 23 subject matter experts were asked to contribute to quantify the research model, assign weights to all perspective's level, and assign weights for the included factors within the different perspectives. In the quantification phase, the objective is to gain more insight into the importance of both perspective level and factor level. The result declares that the requirement validation perspective as the most important perspective across the model. This perspective holds the highest relative weight of 25.75%. The requirement validation is a common practice but is not necessarily empowered during the

delivery phases. The testing perspective comes next with a relative weight of 22.19%. The third perspective is the software change management control perspective with a relative weight of 17.57%, followed by organization and culture as fourth with a relative weight of 17.81%. The last important perspective is technology, with a relative weight of 16.81%. Practices that have more human tendency such as organization and culture or product management methods showed higher importance in terms of software quality influence.

In the testing perspective, the testing objective factor is ranked as the most important factor with a relative importance of 23%. This is consistent with what the author emphasized in [12], which is that the testing objective is mainly focused on defining the purpose of the testing, such as acceptance testing, quality of service testing, and regression testing. The testing objective has been marked as a quality factor 20 times in a study count, as per the finding in this paper. The second most important factor in this perspective is a testing activity with a relative importance of 22%. The third ranking factor is the testing level with a relative importance of 21%. At the testing level, the main interest for this factor is how far testing should be conducted, either unit testing, system testing, or integration testing. As indicated by [51], the ideal testing level should be mostly focusing on unit testing, then integration testing, followed by system testing “GUI”. Supported by automation testing tools, the fourth-ranking factor is the testing approach with a relative importance of 18%. Lastly, the test artifact factor takes the lowest rank in the testing perspective with 15%.

For the requirement validation perspective, consistency is ranked as the most important factor, with a relative importance of 21%. Correctness comes second, with a relative importance of 18%. This result comes in alignment with the findings of [13], where the author described these two factors as the main quality requirements. Also, those two factors will jointly come into tandem in favor of the verifiability factor, which is the third-ranking factor with a relative importance of 17%. The validity factor comes as the fourth-ranked factor with a relative importance of 16%. Lastly, with a relative importance of 14%, realism comes as the lowest factor of validation requirement perspective. Adaptive behaviors of keeping requirement validation factors presented across the delivery process and technique. The author in [13] believed that starting from an initial requirement gathering, passing by prototyping a testing-oriented activity such as test case generations, will deliver a decent and acceptable threshold in validating the software requirements.

In the technology perspective, testing tools are the most important with a relative importance of 31%. From a technology perspective, the testing tool is the largest enabler of software quality. According to [12], using testing tools and products with service management tools along with the right test will adhere to quality. The study shows that using testing tools in unit testing will deliver technical wealth that will offer faster, cleaner, and better-quality deliverables, which require a very low cost of required maintenance. The second-ranked factor is the framework and environment structure with a relative importance of 27% and the third-ranked factor is performance with relative importance of 25%. The last-ranked factor is automation level with 17% of relative importance. As concluded by the author in [12], based on a survey, automation testing is difficult to achieve in organizations and only 6% of professionals think it is doable. Executives are looking forward to executing the test cases in an automated manner to save cost and time and increase reliability by detecting defects in early stages. Automation testing needs to be adopted more and more in organization practices. On the other hand, the authors in [12] indicated in their research the need to set a proper return on investments while applying automated testing, and this should be considered by following the metrics: test coverage, test effectiveness, test strategy, tools, cost of test automation and peoples. As for designing the survey, which was distributed among professionals from products and service organizations, the result showed that testing represents 28% of overall IT spending in 5 years [12], and automation has a low contribution of return on investment, which is a sign for baring more attention into this factor.

In a software change management control perspective, agile has ranked as the most important factor with a relative importance of 32%. This finding comes in alignment

with the result presented in [16], which states that agile nowadays is associated with a productive method of delivery and is directly enabling testability, maintainability, and reusability of the development process of software. The study shows that the agile method has a positive impact on the organization's targets and measurement of the level of satisfaction. Furthermore, the result of analyzing the quantitative data of 325 participants in [67] showed the real added value through the use of agile, which has been accepted in many corporate cultures in various areas such as structure, methods discipline, direction, or work instruction, cooperation executive and team. The second important factor is DevOps with a relative importance of 26%. DevOps has a stronger impact on the quality assurance concept. According to DevOps features, automation process, sharing, and direct measurements contribute to quality favors. In DevOps, fast feedback assuring quality can be reached by practices not only in a theatrical context [19]. The next factor in rank is the internal process with a relative importance of 21%. The author in [54] concluded that constant review and management of internal process improvement attempts make the software process more operational and raise the quality of the software product. On the other hand, the release factor has the same level of importance with 21%. According to [68], the sequence of project changes can be viewed as small projects because they may have their own impact and business case feasibility, as well as a unique method of execution and testing, business acceptance, and documentation. All the preceding changes are essential to keep the quality of this change. Furthermore, any changes should not be regarded as separate units when they are connected with other units to provide collaborative activities or duties. A priority for the scope of modifications, essential and noncritical, must be determined and scheduled within the releases. A release may include more than one change request. A change request scope must be signed, developed, and accepted by the change requester before it can be documented.

In the organization and culture perspective, the documentation factor is the most important in this perspective with a relative importance of 19%. The documentation aspect has become a more and more emerging aspect when it comes to quality. The finding in [68] showed that the quality of documenting user stories for testing, requirements, changes, documentation of code, and amendment comes as an insightful practice of the ISO/ICE 19761 guide, and with other ISO version guides as well. In addition, the result in [55] emphasized that documentation is becoming a required aspect in rapid delivery methods such as agile. The second factor is reporting with a relative importance of 18% and the third factor is team building with a relative importance of 17%. According to [6], team building is a very important topic in quality as a team should consist of business analysis developers and software testers. The "know-how" is a primary driver to improve the quality. Quality standards and leadership are the two factors that come in the fourth rank with 16% of relative importance. The quality standard refers to the followed quality foundation and practices within an organization, as referred to ISO with the verity of versions or ISQTB, in many factors such as agile and DevOps, which we have studied above. To those factors, human interpersonal skill is a game-changer when it comes to enabling those factors in favor of quality assurance. Interpersonal skills and lack of leadership is an area that needs further studies and examination in the existing models as highlighted by [16]. As indicated by [7], the educational background would be an accelerator to the software quality domain, since the tendency of professionals emerging with non-IT domain practices and backgrounds due to high growth and demand in the IT industry. The study showed that this might produce a knowledge gap because of the differences, which might cause a risk of quality due to these challenges between two aspects: technology and humans. On the other hand, another human factor that comes in the lowest rank in this perspective is technical skills, which is a certificate with a relative importance of 14%. This factor is the lowest factor in this model.

The model has been applied in two real-world case studies to test its reliability and applicability. Thus, it has proved its capability to measure the maturity of the SQA practices. A few insights can be stated from the cases to show how these two cases performed in the

model. Case 1 scored/performed better in all perspectives (Testing, Requirement validation, Software Change Management Control, Technology, and Organization and Culture) compared to Case 2. This is due to that Case 1 is a well-established telecommunication company with a history of developing software systems as well as it being considered as the leading telecommunication company in the telecommunication industry in Saudi Arabia and has benefited much from government support in the early days of its launch. Case 2 was established decades after Case 1 and the competition was intense with Case 1. Yet, Case 2 has much potential in improving and benefiting from this research outcome.

## 7. Conclusions

The intensive literature review on the SQA practices along with conducting interviews with IT experts and survey applications showed the high need for having a quality assurance maturity model within the telecommunication industry practices with respect to software development. A complete view of the maturity is established by incorporating multiple perspectives into the model for the benefit of software development. Therefore, this study is intended to propose a maturity model that assesses SQA maturity in the telecommunication industry. The proposed maturity model classifies the important factors into five perspectives: Testing, Requirements validation, Technology, Software quality management control, and Organization and Culture, in which the importance level of each factor in the SQA model was determined based on its evaluated weights. This proposed model has been reviewed and validated with the help of 35 SQA and IT experts, in which the responses have confirmed that the model is appropriate and useful to assess the maturity of the SQA practices. Moreover, the model has been quantified by 23 of the involved experts where weights have been assigned to all the perspectives level, along with the included factors within these perspectives. The result of the quantification phase showed that the most important perspective across the model is the requirement validation perspective with a relative weight of 25.75%. Technology is the least important perspective with a relative weight of 16.81%, which means that technology is the least to worry about when it comes to the maturity of the practices than other perspectives. This research allows for a better understanding of the different dimensions of SQA practices. It helps in increasing the knowledge of how telecommunication companies assess their software development maturity. With respect to the research gaps, it provides a structured and comprehensive investigation of the important factors and assesses their impact on the SQA practice maturity in a multi-perspective approach using both quantitative and qualitative metrics. On the practical side, this research helps software developers and decision makers in the telecommunication industry to classify and organize their priorities and pinpoint areas of strength and weakness in order to develop successful software systems.

It is important to note the limitations of this study. The outcomes of this research are time and context dependent, in which the dimensions examined may change in the future due to technological advancements and the software development landscape as well as the continuous advances in the SQA practices. This model was applied to two telecommunication companies, yet applying it to more cases should certainly improve its accuracy and ensure a robust and more generalizable model. Furthermore, finding knowledgeable and qualified experts is one of this approach's challenges. The quality of the outcomes depends on the quality of the expert's inputs. While experts invited to participate in this research were selected carefully to ensure better results, experts' judgments may be impacted by human biases, interests, and errors. This issue was controlled by conducting proper selection procedures for the expert panels. Also, the model considers a large number of factors that were grouped into perspectives. The factors' weights are calculated using pairwise comparisons performed by experts. The more factors an expert is asked to perform pairwise comparisons for, the more they tend to feel bored and lose concentration, resulting in the accuracy being impacted. This research design tried to avoid this issue by constructing multiple expert panels and breaking down the model into smaller and manageable tasks to facilitate the experts' judgments.



For future research, this model can be expanded to go beyond the telecommunications industry that it was intended for by incorporating input from subject matter experts from other industries. The model can be applied to other industries and sectors to test its validity and make the necessary changes accordingly. Other SQA maturity factors can be added to the model as the SQA maturity models advances. In the future, pairwise comparisons can be repeated by new experts in order to keep the model weights relevant and up-to-date as SQA practices advance.

**Author Contributions:** Conceptualization, A.A.M. and S.A.; Methodology, A.A.M. and S.A.; Software, A.A.M. and S.A.; Formal analysis, A.A.M.; Data curation, A.A.M.; Writing—original draft, A.A.M.; Writing—review and editing, S.A.; Visualization, A.A.M.; Supervision, S.A.; Project administration, S.A.; Funding acquisition, S.A. All authors have read and agreed to the published version of the manuscript.

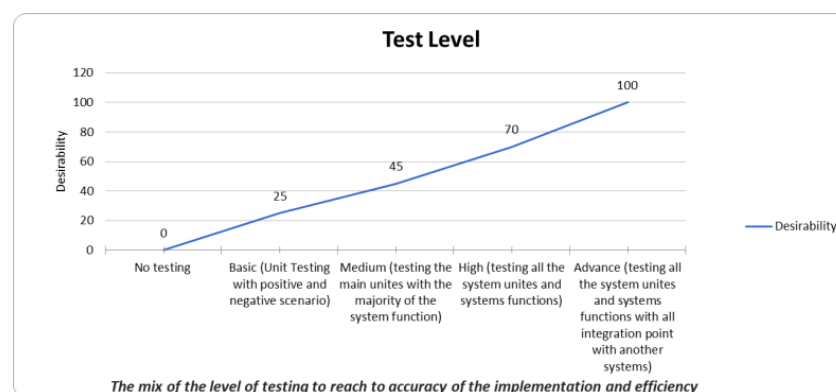
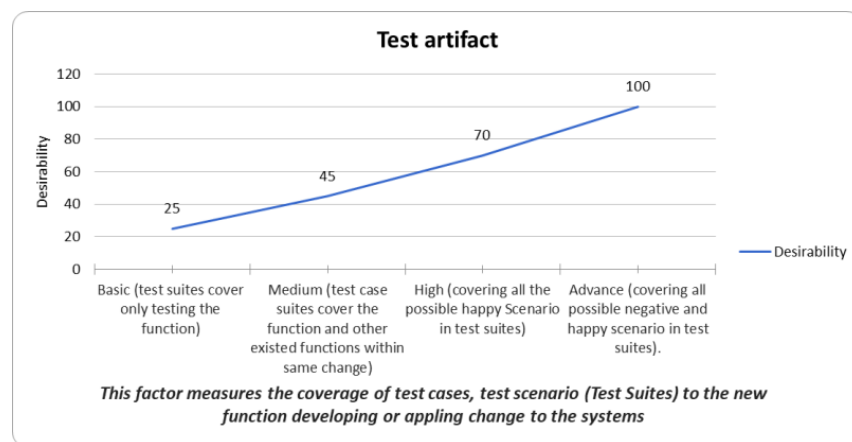
**Funding:** This research was funded by the Researchers Supporting Program at King Saud University. Researchers Supporting Project number (RSPD2023R867), King Saud University, Riyadh, Saudi Arabia.

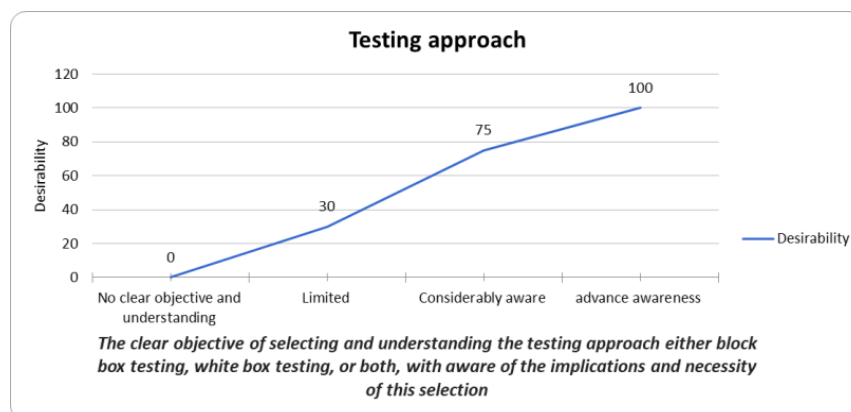
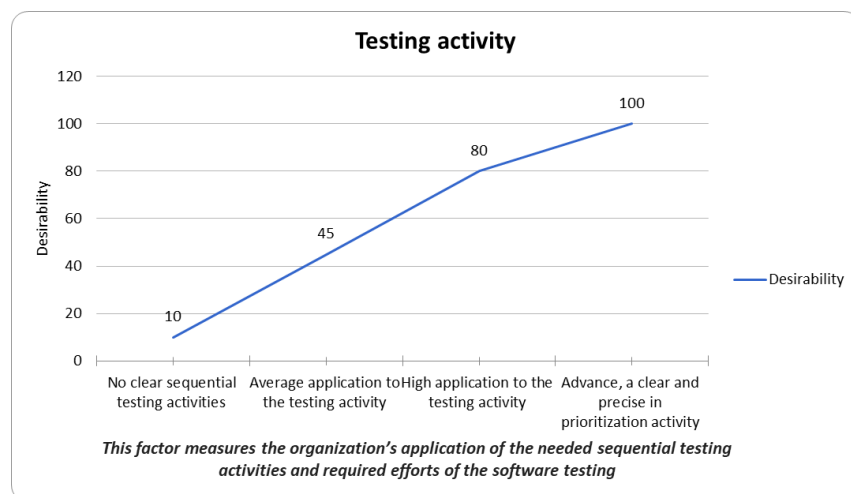
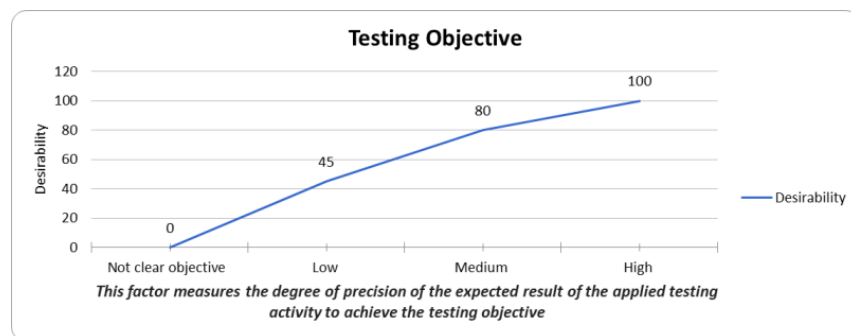
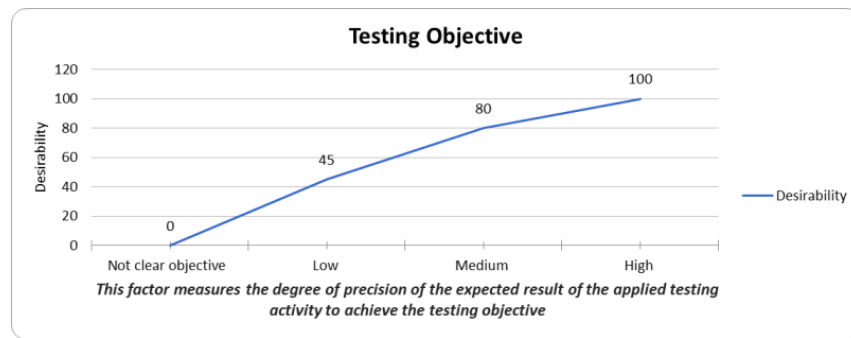
**Data Availability Statement:** The data are available from the corresponding author on reasonable request.

**Acknowledgments:** The authors extend their appreciation to the Researchers Supporting Program at King Saud University. Researchers Supporting Project number (RSPD2023R867), King Saud University, Riyadh, Saudi Arabia.

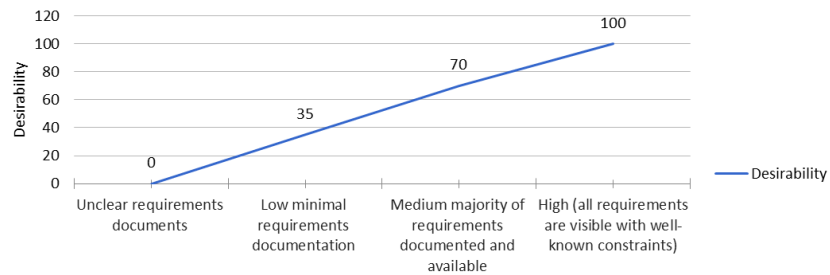
**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A



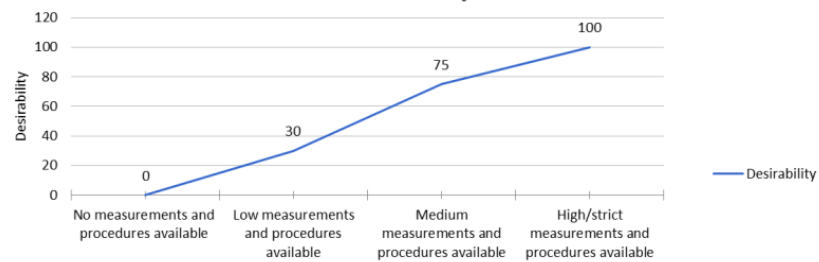


### Competency



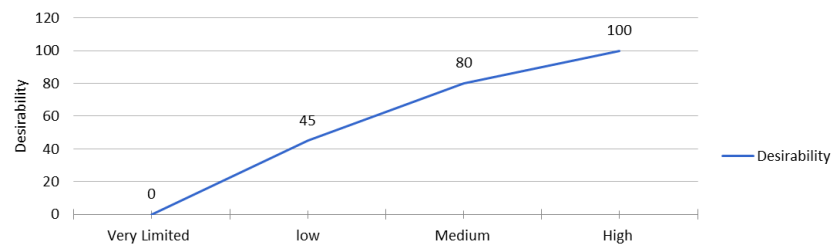
*This factor to Level of certainty which requirements documents contain, all the requirements and updates and their accompanying constraints*

### Consistency



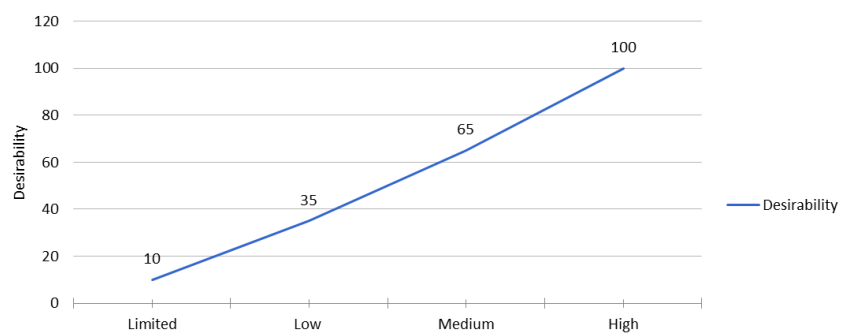
*The level of the measurements and procedures in place for the requirements items to prevent contradicting other requirements related to the existing and/or other software features and functions*

### Correctness

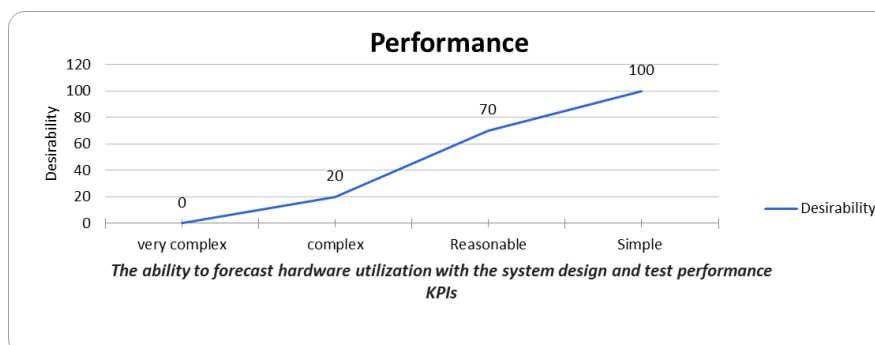
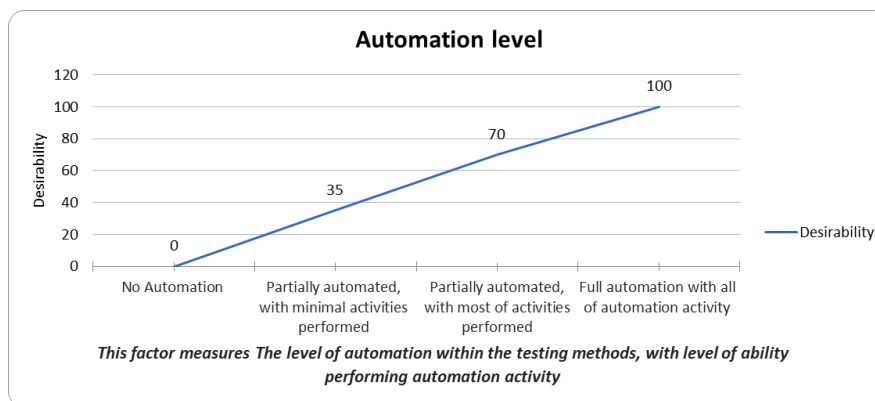
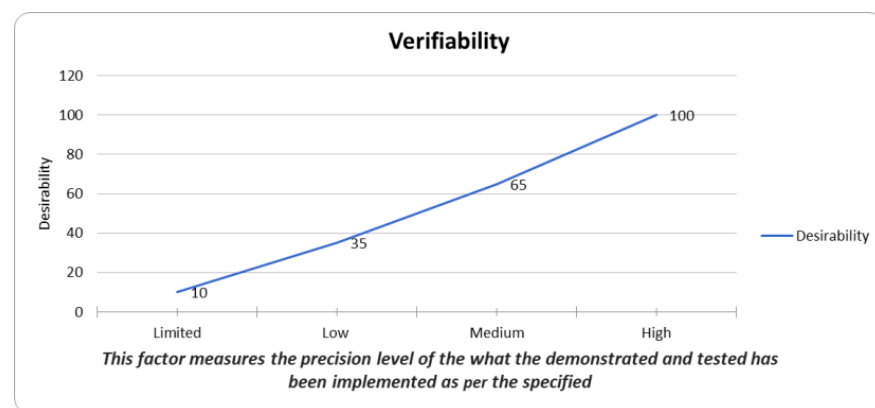
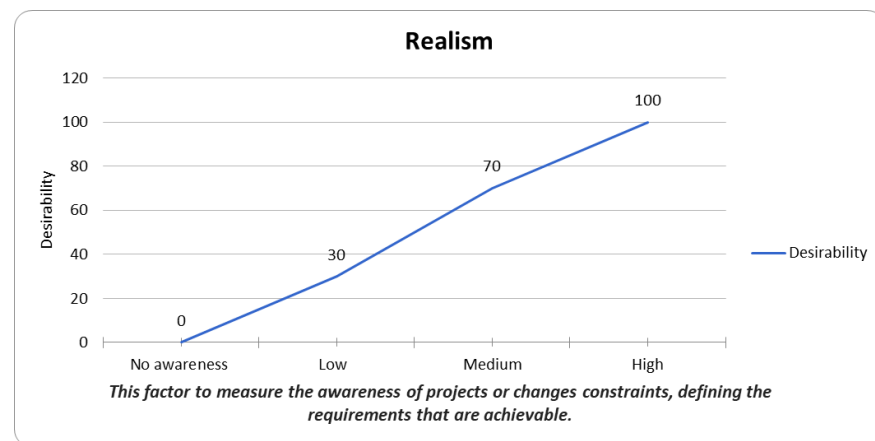


*This factor measures the acceptable degree of mutual understanding to the requirements which implies to be mapped with the compliance to policies, standards, and laws*

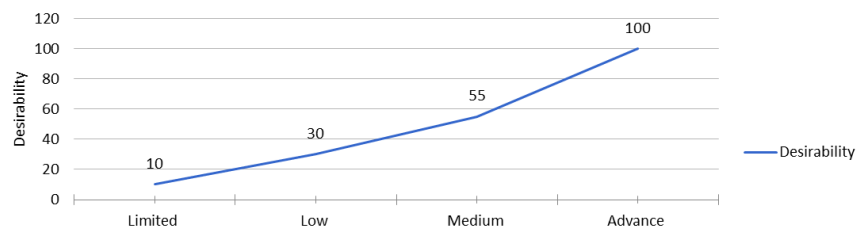
### Validity



*this factor measure the alignment on what the system function and what needs to perform based on what is proposed by stakeholders.*

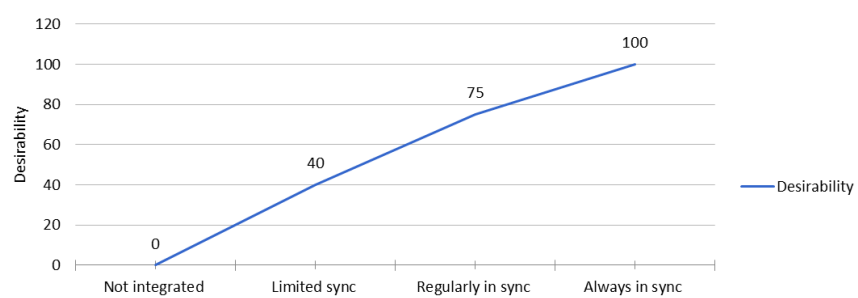


### Testing Tools



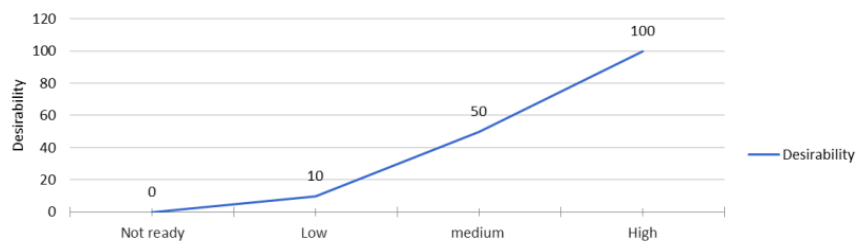
*This Factor to measure the availability of testing tools which used in test cases generation, testing execution, testing tracing, defects logging.*

### Framework & Environment Structure



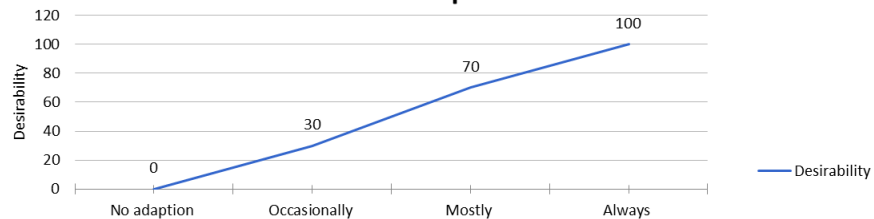
*this factor measure the level of integration and the sync of the components during all delivery phases (Dev Env, UAT Env, pre-prod Env and production)*

### Agile

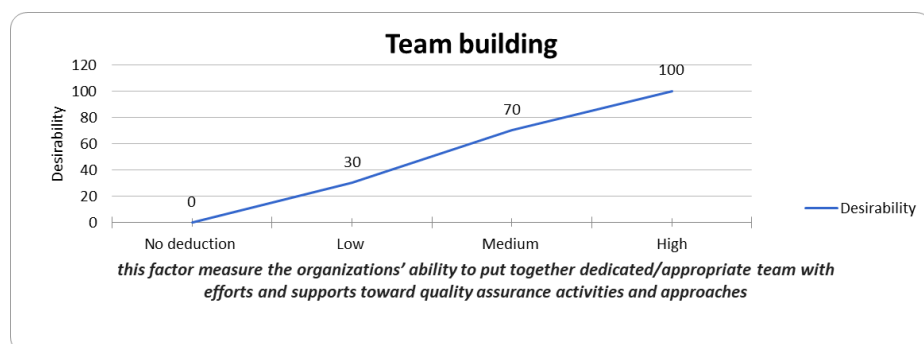
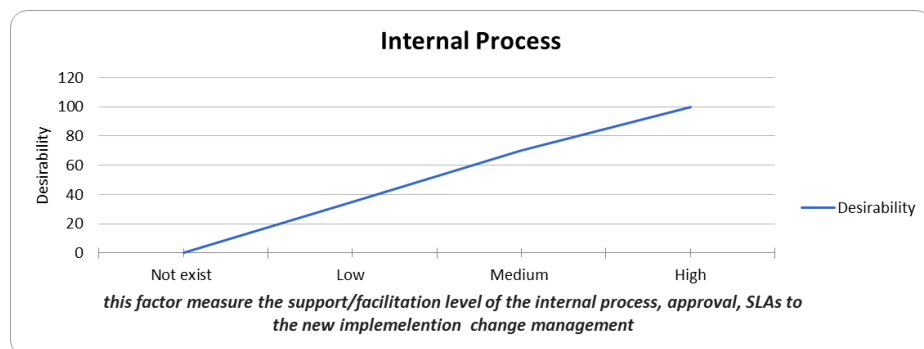
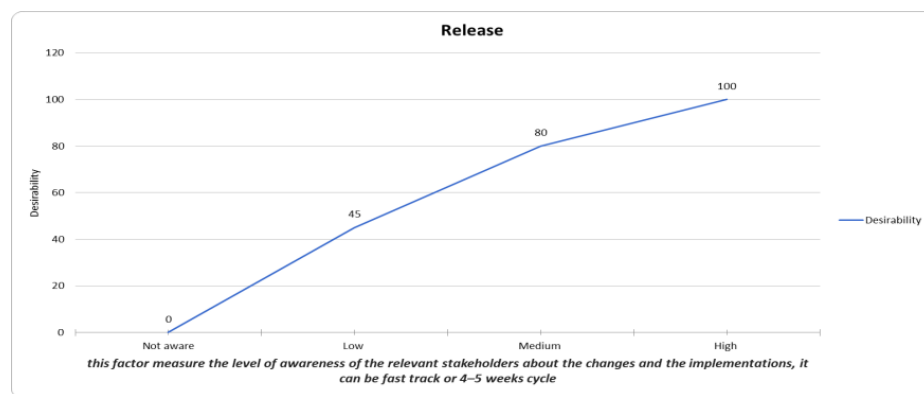


*This factor measures the ability of the telecom companies to perform the test artifact procedure at the highest level*

### DevOps

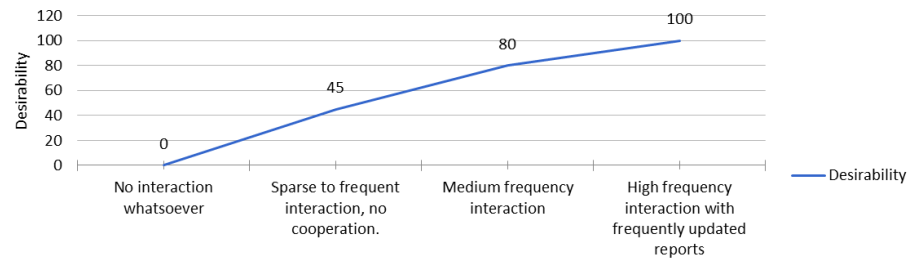


*this factor measure the organization adoption to the DevOps delivery methodology*



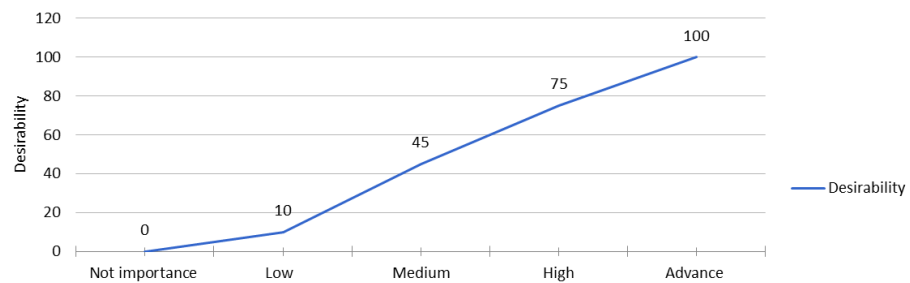


### Reporting Quality



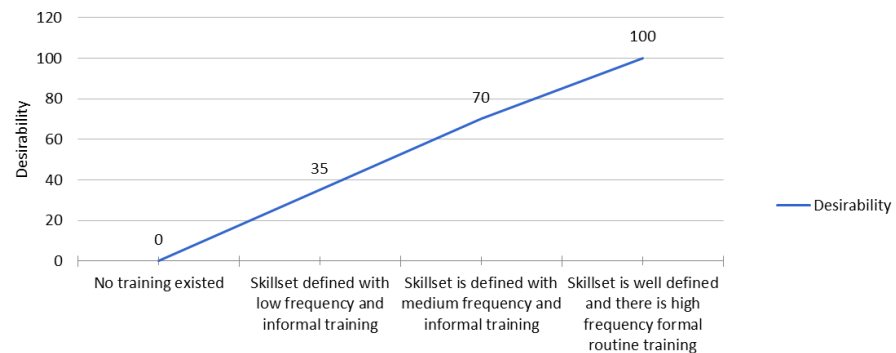
*the factor to measure the reporting efforts and level communication that undertaken between stakeholders such Testing reports, defect report, performance report, frequency of reports, management expose to the reports*

### Documentation



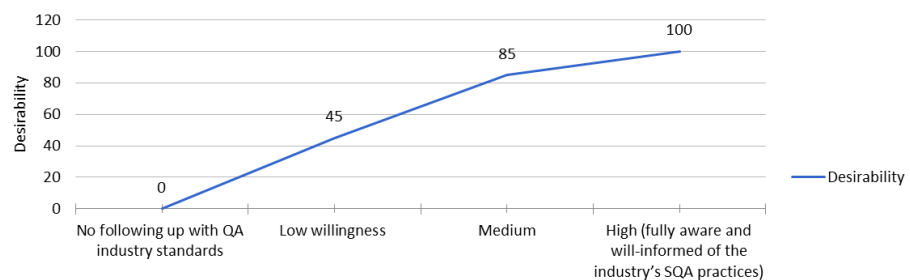
*this factor to measure the awareness level of the importance to keep the traceability of the documentation versions among teams, and accessibility to the document's repository and templates..*

### Certification and technical skills



*this factor measure the availability of the clear definitions to the required technical skills, with the ability to provide suitable training .*

### Quality Standards



*Does the organization keep up and being well informed about the quality assurance standards and practices such as ISO, ISQTB*

## References

1. Zhao, Y.; Hu, Y.; Gong, J. Research on International Standardization of Software Quality and Software Testing. In Proceedings of the 2021 IEEE/ACIS 20th International Fall Conference on Computer and Information Science (ICIS Fall), Xi'an, China, 13–15 October 2021; pp. 56–62. [\[CrossRef\]](#)
2. Wong, W.Y.; Hai Sam, T.; Too, C.W.; Fong Pok, W. Software Quality Assurance Plan: Setting Quality Assurance Checkpoints within the Project Life Cycle and System Development Life Cycle. In Proceedings of the 2022 IEEE 18th International Colloquium on Signal Processing & Applications (CSPA), Selangor, Malaysia, 12 May 2022; pp. 214–219. [\[CrossRef\]](#)
3. Ibarra, S.; Munoz, M. Support tool for software quality assurance in software development. In Proceedings of the 2018 7th International Conference On Software Process Improvement (CIMPS), Guadalajara, Mexico, 17–19 October 2018; pp. 13–19. [\[CrossRef\]](#)
4. Atoum, I.; Baklizi, M.K.; Alsmadi, I.; Ootom, A.A.; Alhersh, T.; Ababneh, J.; Almalki, J.; Alshahrani, S.M. Challenges of Software Requirements Quality Assurance and Validation: A Systematic Literature Review. *IEEE Access* **2021**, *9*, 137613–137634. [\[CrossRef\]](#)
5. Poth, A.; Meyer, B.; Schlicht, P.; Riel, A. Quality Assurance for Machine Learning—An approach to function and system safeguarding. In Proceedings of the 2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS), Macau, China, 11–14 December 2020; pp. 22–29. [\[CrossRef\]](#)
6. Poth, A.; Kottke, M.; Riel, A. Evaluation of Agile Team Work Quality. In *Agile Processes in Software Engineering and Extreme Programming—Workshops*; Paasivaara, M., Kruchten, P., Eds.; Lecture Notes in Business Information Processing; Springer International Publishing: Cham, Switzerland, 2020; Volume 396, pp. 101–110. [\[CrossRef\]](#)
7. Sabev, P.; Grigorova, K. A Survey on State of Software Quality Assurance in Bulgaria. In Proceedings of the 20th International Conference on Computer Systems and Technologies, Ruse, Bulgaria, 21–22 June 2019; pp. 124–130. [\[CrossRef\]](#)
8. Saleem, G.; Azam, F.; Younus, M.U.; Ahmed, N.; Li, Y. Quality assurance of web services: A systematic literature review. In Proceedings of the 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 14–17 October 2016; pp. 1391–1396. [\[CrossRef\]](#)
9. Kettunen, P. Bringing Total Quality in to Software Teams: A Frame for Higher Performance. In *Lean Enterprise Software and Systems*; Fitzgerald, B., Conboy, K., Power, K., Valerdi, R., Morgan, L., Stol, K.-J., Eds.; Lecture Notes in Business Information Processing; Springer: Berlin/Heidelberg, Germany, 2013; Volume 167, pp. 48–64. [\[CrossRef\]](#)
10. López, L.; Burgués, X.; Martínez-Fernández, S.; Vollmer, A.M.; Behutiye, W.; Karhapää, P.; Franch, X.; Rodríguez, P.; Oivo, M. Quality measurement in agile and rapid software development: A systematic mapping. *J. Syst. Softw.* **2022**, *186*, 111187. [\[CrossRef\]](#)
11. Sophocleous, R.; Kapitsaki, G.M. Examining the Current State of System Testing Methodologies in Quality Assurance. In *Agile Processes in Software Engineering and Extreme Programming*; Stray, V., Hoda, R., Paasivaara, M., Kruchten, P., Eds.; Lecture Notes in Business Information Processing; Springer International Publishing: Cham, Switzerland, 2020; Volume 383, pp. 240–249. [\[CrossRef\]](#)
12. Reine De Reanzi, S.; Ranjit Jeba Thangaiah, P. A survey on software test automation return on investment, in organizations predominantly from Bengaluru, India. *Int. J. Eng. Bus. Manag.* **2021**, *13*, 184797902110620. [\[CrossRef\]](#)
13. Tran, H.K.V.; Unterkalmsteiner, M.; Börstler, J.; Ali, N. bin Assessing test artifact quality—A tertiary study. *Inf. Softw. Technol.* **2021**, *139*, 106620. [\[CrossRef\]](#)
14. Shen, P.; Ding, X.; Ren, W.; Yang, C. Research on Software Quality Assurance Based on Software Quality Standards and Technology Management. In Proceedings of the 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Busan, Republic of Korea, 27–29 June 2018; pp. 385–390. [\[CrossRef\]](#)
15. Reddy, M.P.; Reddy, K.L.R. Policies, Processes, Procedures and Measurement in Software Quality Assurance: A State of Art Survey. *Int. J. Innov. Sci. Eng. Technol.* **2017**, *4*, 8.
16. Arcos-Medina, G.; Mauricio, D. Aspects of software quality applied to the process of agile software development: A systematic literature review. *Int. J. Syst. Assur. Eng. Manag.* **2019**, *10*, 867–897. [\[CrossRef\]](#)
17. Sun, L.; Nazir, S.; Hussain, A. Multicriteria Decision Making to Continuous Software Improvement Based on Quality Management, Assurance, and Metrics. *Sci. Program.* **2021**, *2021*, 9953618. [\[CrossRef\]](#)
18. Gonen, B.; Sawant, D. Significance of Agile Software Development and SQA Powered by Automation. In Proceedings of the 2020 3rd International Conference on Information and Computer Technologies (ICICT), San Jose, CA, USA, 9–12 March 2020; pp. 7–11. [\[CrossRef\]](#)
19. Mishra, A.; Otaiwi, Z. DevOps and software quality: A systematic mapping. *Comput. Sci. Rev.* **2020**, *38*, 100308. [\[CrossRef\]](#)
20. Panichella, A. Beyond Unit-Testing in Search-Based Test Case Generation: Challenges and Opportunities. In Proceedings of the 2019 IEEE/ACM 12th International Workshop on Search-Based Software Testing (SBST), Motreal, QC, Canada, 26–27 May 2019; pp. 7–8. [\[CrossRef\]](#)
21. Backlund, F.; Chronéer, D.; Sundqvist, E. Project Management Maturity Models—A Critical Review. *Procedia-Soc. Behav. Sci.* **2014**, *119*, 837–846. [\[CrossRef\]](#)
22. Lee, J.; Lee, D.; Kang, S. An Overview of the Business Process Maturity Model (BPMM). In *Advances in Web and Network Technologies, and Information Management*; Chang, K.C.-C., Wang, W., Chen, L., Ellis, C.A., Hsu, C.-H., Tsoi, A.C., Wang, H., Eds.; In Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2007; pp. 384–395. [\[CrossRef\]](#)

23. Tarhan, A.; Turetken, O.; Reijers, H.A. Business process maturity models: A systematic literature review. *Inf. Softw. Technol.* **2016**, *75*, 122–134. [\[CrossRef\]](#)
24. Strutt, J.E.; Sharp, J.V.; Terry, E.; Miles, R. Capability maturity models for offshore organisational management. *Environ. Int.* **2006**, *32*, 1094–1105. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Paulk, M. Capability Maturity Model for Software. In *Encyclopedia of Software Engineering*; Marciniak, J.J., Ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2002; p. sof589. [\[CrossRef\]](#)
26. Caballero, I.; Caro, A.; Calero, C.; Piattini, M. IQM3: Information Quality Management Maturity Model. *J. Univers. Comput. Sci.* **2008**, *14*, 29. [\[CrossRef\]](#)
27. Kim, S.; Pérez-Castillo, R.; Caballero, I.; Lee, D. Organizational process maturity model for IoT data quality management. *J. Ind. Inf. Integr.* **2022**, *26*, 100256. [\[CrossRef\]](#)
28. Yeo, K.T.; Ren, Y. Risk management capability maturity model for complex product systems (CoPS) projects. *Syst. Eng.* **2009**, *12*, 275–294. [\[CrossRef\]](#)
29. Zephir, O.; Minel, S.; Chapotot, E. A maturity model to assess organisational readiness for change. *Int. J. Technol. Manag.* **2011**, *55*, 286. [\[CrossRef\]](#)
30. Skelsey, D.; King, D.; Sidhu, R.; Smith, R.; Perkins, C.; Change Management Institute; APMG International. *The Effective Change Manager: The Change Management Body of Knowledge*; Kogan Page: London, UK, 2014.
31. Neff, A.A.; Hamel, F.; Herz, T.P.; Uebernickel, F.; Brenner, W.; vom Brocke, J. Developing a maturity model for service systems in heavy equipment manufacturing enterprises. *Inf. Manage.* **2014**, *51*, 895–911. [\[CrossRef\]](#)
32. Rapaccini, M.; Saccani, N.; Pezzotta, G.; Burger, T.; Ganz, W. Service development in product-service systems: A maturity model. *Serv. Ind. J.* **2013**, *33*, 300–319. [\[CrossRef\]](#)
33. Alzahrani, S.; Daim, T.U. Evaluation of the Cryptocurrency Adoption Decision Using Hierarchical Decision Modeling (HDM). In Proceedings of the 2019 Portland International Conference on Management of Engineering and Technology (PICMET), Portland, OR, USA, 25–29 August 2019; pp. 1–7. [\[CrossRef\]](#)
34. Lavoie, J.R.; Daim, T. Towards the assessment of technology transfer capabilities: An action research-enhanced HDM model. *Technol. Soc.* **2020**, *60*, 101217. [\[CrossRef\]](#)
35. van Blommestein, K.C.; Daim, T.U. Residential energy efficient device adoption in South Africa. *Sustain. Energy Technol. Assess.* **2013**, *1*, 13–27. [\[CrossRef\]](#)
36. Cleland, D.I.; Kocaoglu, D.F. *Engineering Management*; McGraw-Hill series in industrial engineering and management science; McGraw-Hill: New York, NY, USA, 1981.
37. Kocaoglu, D.F. A participative approach to program evaluation. *IEEE Trans. Eng. Manag.* **1983**, *EM-30*, 112–118. [\[CrossRef\]](#)
38. Turan, T.; Amer, M.; Tibbot, P.; Almasri, M.; Fayez, F.A.; Graham, S. Use of Hierarchal Decision Modeling (HDM) for selection of graduate school for master of science degree program in engineering. In Proceedings of the PICMET '09-2009 Portland International Conference on Management of Engineering Technology, Portland, OR, USA, 2–6 August 2009; pp. 535–549. [\[CrossRef\]](#)
39. Barham, H. Development of a Readiness Assessment Model for Evaluating Big Data Projects: Case Study of Smart City in Oregon, USA. Ph.D. Thesis, Portland State University, Portland, OR, USA, 2019. [\[CrossRef\]](#)
40. Barham, H.; Daim, T.U. The use of readiness assessment for big data projects. *Sustain. Cities Soc.* **2020**, *60*, 102233. [\[CrossRef\]](#)
41. Hogaboam, L.; Ragel, B.; Daim, T. Development of a Hierarchical Decision Model (HDM) for health technology assessment (HTA) to design and implement a new patient care database for low back pain. In Proceedings of the PICMET '14 Conference: Portland International Center for Management of Engineering and Technology, Infrastructure and Service Integration, Kanazawa, Japan, 27–31 July 2014; pp. 3511–3517.
42. Alanazi, H.A.; Daim, T.U.; Kocaoglu, D.F. Identify the best alternatives to help the diffusion of teleconsultation by using the Hierarchical Decision Model (HDM). In Proceedings of the 2015 Portland International Conference on Management of Engineering and Technology (PICMET), Portland, OR, USA, 2–6 August 2015; pp. 422–432. [\[CrossRef\]](#)
43. Phan, K. Innovation Measurement: A Decision Framework to Determine Innovativeness of a Company. Ph.D. Thesis, Portland State University, Portland, OR, USA, 2013. [\[CrossRef\]](#)
44. Alzahrani, S.; Daim, T.U. Technology Adoption: Case of Cryptocurrency. In *Recent Developments in Individual and Organizational Adoption of ICTs*; Yildiz, O., Ed.; IGI Global: Hershey, PA, USA, 2021; pp. 96–119. [\[CrossRef\]](#)
45. Lavoie, J. A Scoring Model to Assess Organizations' Technology Transfer Capabilities: The Case of a Power Utility in the Northwest USA. Ph.D. Thesis, Portland State University, Portland, OR, USA, 2019. [\[CrossRef\]](#)
46. Estep, J. Development of a Technology Transfer Score for Evaluating Research Proposals: Case Study of Demand Response Technologies in the Pacific Northwest. Ph.D. Thesis, Portland State University, Portland, OR, USA, 2017. Available online: [https://pdxscholar.library.pdx.edu/open\\_access\\_etds/3479](https://pdxscholar.library.pdx.edu/open_access_etds/3479) (accessed on 18 December 2022).
47. Gibson, E. A Measurement System for Science and Engineering Research Center Performance Evaluation. In Proceedings of the 2016 Portland International Conference on Management of Engineering and Technology (PICMET), Honolulu, HI, USA, 4–8 September 2016. [\[CrossRef\]](#)
48. Abotah, R. Evaluation of Energy Policy Instruments for the Adoption of Renewable Energy: Case of Wind Energy in the Pacific Northwest U.S. Ph.D. Thesis, Portland State University, Portland, OR, USA, 2014. [\[CrossRef\]](#)

49. Estep, J.; Daim, T. A framework for technology transfer potential assessment. In Proceedings of the 2016 Portland International Conference on Management of Engineering and Technology (PICMET), Honolulu, HI, USA, 4–8 September 2016; pp. 2846–2852. [\[CrossRef\]](#)
50. Chen, H.; Kocaoglu, D.F. A sensitivity analysis algorithm for hierarchical decision models. *Eur. J. Oper. Res.* **2008**, *185*, 266–288. [\[CrossRef\]](#)
51. Sabev, P.; Grigorova, K. A Comparative Study of GUI Automated Tools for Software Testing. In Proceedings of the Third International Conference on Advances and Trends in Software Engineering, Lisbon, Portugal, 21–25 February 2016; p. 9.
52. Felderer, M.; Ramler, R. Quality Assurance for AI-Based Systems: Overview and Challenges (Introduction to Interactive Session). In *Software Quality: Future Perspectives on Software Engineering Quality*; Winkler, D., Biffl, S., Mendez, D., Wimmer, M., Bergsmann, J., Eds.; Lecture Notes in Business Information Processing; Springer International Publishing: Cham, Switzerland, 2021; Volume 404, pp. 33–42. [\[CrossRef\]](#)
53. Tao, C.; Gao, J.; Wang, T. Testing and Quality Validation for AI Software—Perspectives, Issues, and Practices. *IEEE Access* **2019**, *7*, 120164–120175. [\[CrossRef\]](#)
54. Ji, S.; Li, Q.; Cao, W.; Zhang, P.; Muccini, H. Quality Assurance Technologies of Big Data Applications: A Systematic Literature Review. *Appl. Sci.* **2020**, *10*, 8052. [\[CrossRef\]](#)
55. Rashidi, H.; Sadeghzadeh Hemayati, M. Software Quality Models: A Comprehensive Review and Analysis. *J. Electr. Comput. Eng. Innov.* **2018**, *6*, 59–76. [\[CrossRef\]](#)
56. Rashid, J.; Nisar, M.W. How to Improve a Software Quality Assurance in Software Development—A Survey. *Int. J. Comput. Sci. Inf. Secur.* **2016**, *14*, 11.
57. Lee, M.-C. Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance. *Br. J. Appl. Sci. Technol.* **2014**, *4*, 3069–3095. [\[CrossRef\]](#)
58. Gan, M.; Yucel, Z.; Monden, A. Improvement and Evaluation of Data Consistency Metric CIL for Software Engineering Data Sets. *IEEE Access* **2022**, *10*, 70053–70067. [\[CrossRef\]](#)
59. Thota, M.K.; Shajin, F.H.; Rajesh, P. Survey on software defect prediction techniques. *Int. J. Appl. Sci. Eng.* **2020**, *17*, 331–344. [\[CrossRef\]](#)
60. Jaskolka, J.; Hamid, B.; Kokaly, S. Software Design Trends Supporting Multiconcern Assurance. *IEEE Softw.* **2022**, *39*, 22–26. [\[CrossRef\]](#)
61. Nazir, M. Software Quality Assurance and Android Application Development: A Comparison among Traditional and Agile Methodology. *Res. J. Comput. Sci. Inf. Technol.* **2020**, *4*, 1–29. [\[CrossRef\]](#)
62. Slaughter, A.E.; Permann, C.J.; Miller, J.M.; Alger, B.K.; Novascone, S.R. Continuous Integration, In-Code Documentation, and Automation for Nuclear Quality Assurance Conformance. *Nucl. Technol.* **2021**, *207*, 923–930. [\[CrossRef\]](#)
63. Khan, S.U.; Khan, A.W.; Khan, F.; Khan, M.A.; Whangbo, T.K. Critical Success Factors of Component-Based Software Outsourcing Development From Vendors’ Perspective: A Systematic Literature Review. *IEEE Access* **2022**, *10*, 1650–1658. [\[CrossRef\]](#)
64. Huang, F.; Strigini, L. HEDF: A Method for Early Forecasting Software Defects Based on Human Error Mechanisms. *IEEE Access* **2023**, *11*, 3626–3652. [\[CrossRef\]](#)
65. Lee, T.; Nam, J.; Han, D.; Kim, S.; Peter In, H. Developer Micro Interaction Metrics for Software Defect Prediction. *IEEE Trans. Softw. Eng.* **2016**, *42*, 1015–1035. [\[CrossRef\]](#)
66. Tomar, A. The Survey of Metrics on Software Quality Assurance and Reuse. In Proceedings of the National Conference on Innovative Paradigms in Engineering & Technology (NCIPET-2013), Nagpur, India, 17 February 2013; p. 5.
67. Heimicke, J.; Kaiser, S.; Albers, A. Agile product development: An analysis of acceptance and added value in practice. *Procedia CIRP* **2021**, *100*, 768–773. [\[CrossRef\]](#)
68. Teah, T.-S.; Wong, W.-Y.; Beh, H.-C. The Practical Implication of Software Quality Assurance of Change Control Management: Why Overall IT Project Activities Matters? In Proceedings of the 2019 IEEE 7th Conference on Systems, Process and Control (ICSPEC), Melaka, Malaysia, 13–14 December 2019; pp. 131–136. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.