AN AUTOMATED APPROACH TO VALIDATE CORRECTNESS AND COMPLETENESS OF REQUIREMENTS SPECIFICATION

By

Yazan Mahmoud Al-Kasabrah

Supervisor

Dr. Aysh Al-Hroob

This thesis was submitted in partial fulfillment of the requirements for the Master’s Degree of Science in Software Engineering

Faculty of Graduate Studies

ISRA University

January 2018
An Automated Approach to Validate Correctness and Completeness of Requirements Specification

By

Yazan Mahmoud Al-Kasabrah

Supervisor

Dr. Aysh Al-Hroob

This thesis was submitted in partial fulfillment of the requirements for the Master’s Degree of Science in Software Engineering

Faculty of Graduate Studies

ISRA University

January 2018
The undersigned have examined the thesis entitled ‘An Automated Approach to Validate Correctness and Completeness of Requirements Specification’ presented by Yazan Al-Kasabrah, a candidate for the degree of Master of Science in Software Engineering and hereby certify that it is worthy of acceptance.

_______________________
Date

Dr. Aysh Al-Hroob

_______________________
Date

Dr. Adi Maaita

_______________________
Date

Dr. Thamer Al-Rawashda
DEDICATION

I dedicate this thesis to my Father and Mother who encouraged me too much.

To my brothers and sisters for supporting me.

To my wife who stood with me side by side to accomplish this work, not forgetting my two little kids.

To my colleagues at ISRA University

To my faithful friends in Master degree program

Yazan M. Al-Kasabrah
2017 / 2018
ACKNOWLEDGMENT

After eight months of hard work of research and development, at last, I can say: I DID IT. I had a great experience during this research, even in my business work.

So at first I would kindly present my thanks to my supervisor DR. Aysh Al-Hroob, the Dean of Information Technology faculty, he put the foundation stone for me by accepting and acknowledging me, and for his help and support.

Special thanks, gratitude, and I stand with respect, giving a salute to my encyclopedic advisor Dr. Wael Al-Ziadat, for standing with me step by step to accomplish this work, and for enduring my too many questions. Honestly, I will not be able to accomplish the thesis as perfect as it is, without his support, review and corrections.

A great gratitude for Dr. Ayad Al-Zubaidy for his help, especially in the tool implementation, I was stuck there!

My gratitude and thanks from there, deep in the heart, to my parents for keeping supporting and encouraging me to proceed, and my family who was considering my comfort as her serious business.

Great and lovely emotional thanks to my friends in the Master degree program and so to my business colleagues at ISRA University, being together living good and bad moments as a family.
# TABLE OF CONTENTS

DEDICATION .......................................................................................................................... IV

ACKNOWLEDGMENT ............................................................................................................. V

TABLE OF CONTENTS .......................................................................................................... VI

LIST OF TABLES ................................................................................................................... VIII

LIST OF FIGURES ................................................................................................................ IX

LIST OF ABBREVIATIONS ................................................................................................... X

CHAPTER ONE ....................................................................................................................... 1

INTRODUCTION .................................................................................................................... 1

1.1 Problem Statement ......................................................................................................... 2

1.2 Research Questions ....................................................................................................... 2

1.3 Research Aim ................................................................................................................ 2

1.4 Research Objectives ..................................................................................................... 2

1.5 General Methodology .................................................................................................. 3

1.5.1 Conceptual approach to evaluate RS from C&C criteria ........................................... 4

1.6 Motivation ..................................................................................................................... 6

1.7 Significance ................................................................................................................... 6

CHAPTER TWO ..................................................................................................................... 7

BACKGROUND AND RELATED WORK ............................................................................. 7

2.1 Overview ....................................................................................................................... 7

2.2 Keywords ..................................................................................................................... 7

2.2.1 Correctness ............................................................................................................... 7

2.2.2 Completeness: ......................................................................................................... 7

2.2.3 Quality (based on requirements) ............................................................................ 8

2.3 Related Work ................................................................................................................. 9

2.4 Similar Approaches ..................................................................................................... 12

2.4.1 Approaches ............................................................................................................. 12

2.4.2 Tools ........................................................................................................................ 13

2.4.3 Frameworks and Techniques .................................................................................. 14

2.5 Conclusion .................................................................................................................... 16
CHAPTER THREE ................................................................................................................. 17

METHODOLOGY ...................................................................................................................... 17

3.1 Overview .......................................................................................................................... 17

3.2 Proposed Correctness and Completeness in Requirements Specification Approach ......................................................................................................................... 17

3.2.1 Level 1: Structured Document .................................................................................... 18

3.2.2 Level 2: Dynamic Language ....................................................................................... 18

3.2.3 Level 3: Completeness .............................................................................................. 18

3.3 Map-Rules ....................................................................................................................... 18

3.3.1 Relationship .............................................................................................................. 19

3.4 Conclusion ....................................................................................................................... 23

CHAPTER FOUR ..................................................................................................................... 24

EXPERIMENT AND DISCUSSIONS ......................................................................................... 24

4.1 Experiment ....................................................................................................................... 24

4.2 Case studies: .................................................................................................................... 24

4.2.1 Case study 1: Hospital Reception .............................................................................. 24

4.2.1 Case study 2: Online Shopping ................................................................................. 32

CHAPTER FIVE ....................................................................................................................... 40

CONCLUSIONS and FUTURE WORKS .................................................................................. 40

5.1 Conclusions ..................................................................................................................... 40

5.2 Answering the Research Questions: .............................................................................. 41

5.3 Achieved Objectives ....................................................................................................... 42

5.4 Limitations and Future work .......................................................................................... 42

REFERENCES ......................................................................................................................... 43

Appendix A ............................................................................................................................ 46

Appendix B ............................................................................................................................ 47

Appendix C ............................................................................................................................ 48
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1: Comparison among Approaches</td>
<td>13</td>
</tr>
<tr>
<td>Table 2.2: Comparison among Tools</td>
<td>14</td>
</tr>
<tr>
<td>Table 2.3: Comparison among Frameworks and Techniques</td>
<td>16</td>
</tr>
<tr>
<td>Table 3.1: Represents the rules between two Actors in the Association relation</td>
<td>19</td>
</tr>
<tr>
<td>Table 3.2: Represents the rules between Actor and Use case in the Association relation</td>
<td>20</td>
</tr>
<tr>
<td>Table 3.3: Represents the rules between two Use cases in the Association relation</td>
<td>21</td>
</tr>
<tr>
<td>Table 3.4: Represents the rules between two Use cases in the Include relation</td>
<td>21</td>
</tr>
<tr>
<td>Table 3.5: Represents the rules between Use case and Use case in the Extend relation</td>
<td>22</td>
</tr>
<tr>
<td>Table 3.6: Represents the rules between two Actors in the Generalization relation</td>
<td>22</td>
</tr>
<tr>
<td>Table 3.7: Represents the rules between two Use cases in the Generalization relation</td>
<td>23</td>
</tr>
<tr>
<td>Table 4.1: Case studies rules discussion table</td>
<td>39</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1: The General Methodology of research design</td>
<td>3</td>
</tr>
<tr>
<td>Figure 1.2: Proposed Conceptual Approach</td>
<td>5</td>
</tr>
<tr>
<td>Figure 3.1: Proposed Correctness and Completeness in Requirements Specification Approach</td>
<td>17</td>
</tr>
<tr>
<td>Figure 4.1: Hospital Reception use case diagram</td>
<td>25</td>
</tr>
<tr>
<td>Figure 4.2: Extracting in-boundary items to acquire use case diagram elements</td>
<td>26</td>
</tr>
<tr>
<td>Figure 4.3: Figure 4.3: Filtered form for the in-boundary items</td>
<td>27</td>
</tr>
<tr>
<td>Figure 4.4: Result of applying C&amp;C (Hospital Reception)</td>
<td>28</td>
</tr>
<tr>
<td>Figure 4.5: Changing some elements of Hospital Reception case study (in red color)</td>
<td>29</td>
</tr>
<tr>
<td>Figure 4.6: Error results after changing elements of (Hospital Reception)</td>
<td>30</td>
</tr>
<tr>
<td>Figure 4.7: Sample of applying rule 1 (Hospital Reception)</td>
<td>31</td>
</tr>
<tr>
<td>Figure 4.8: Sample of applying rule 2 (Hospital Reception)</td>
<td>31</td>
</tr>
<tr>
<td>Figure 4.9: Online shopping use case diagram</td>
<td>32</td>
</tr>
<tr>
<td>Figure 4.10: Extracting in-boundary items to acquire use case diagram elements</td>
<td>33</td>
</tr>
<tr>
<td>Figure 4.11: Filtered form for the in-boundary items</td>
<td>34</td>
</tr>
<tr>
<td>Figure 4.12: The result of applying C&amp;C (Online Shopping)</td>
<td>35</td>
</tr>
<tr>
<td>Figure 4.13: Changing some elements of Online Shopping case study (in red color)</td>
<td>36</td>
</tr>
<tr>
<td>Figure 4.14: Error results after changing elements of (Online Shopping)</td>
<td>37</td>
</tr>
<tr>
<td>Figure 4.15: Sample of applying rule 1 (Online Shopping)</td>
<td>38</td>
</tr>
<tr>
<td>Figure 4.16: Sample of applying rule 2 (Online Shopping)</td>
<td>38</td>
</tr>
<tr>
<td>Figure A: An output based on XML exported to excel sheet (Hospital Reception)</td>
<td>46</td>
</tr>
<tr>
<td>Figure B: An output based on XML exported to excel sheet (Online Shopping)</td>
<td>47</td>
</tr>
<tr>
<td>#</td>
<td>Abbreviation</td>
</tr>
<tr>
<td>---</td>
<td>--------------</td>
</tr>
<tr>
<td>1</td>
<td>3Cs</td>
</tr>
<tr>
<td>2</td>
<td>CBR</td>
</tr>
<tr>
<td>3</td>
<td>C&amp;C</td>
</tr>
<tr>
<td>4</td>
<td>DBR</td>
</tr>
<tr>
<td>5</td>
<td>FSF</td>
</tr>
<tr>
<td>6</td>
<td>HLR</td>
</tr>
<tr>
<td>7</td>
<td>NLP</td>
</tr>
<tr>
<td>8</td>
<td>NQQR</td>
</tr>
<tr>
<td>9</td>
<td>PBR</td>
</tr>
<tr>
<td>10</td>
<td>POS</td>
</tr>
<tr>
<td>11</td>
<td>QQR</td>
</tr>
<tr>
<td>12</td>
<td>RDF</td>
</tr>
<tr>
<td>13</td>
<td>RQA</td>
</tr>
<tr>
<td>14</td>
<td>RS</td>
</tr>
<tr>
<td>15</td>
<td>SOFL</td>
</tr>
<tr>
<td>16</td>
<td>SR</td>
</tr>
<tr>
<td>17</td>
<td>SRS</td>
</tr>
<tr>
<td>18</td>
<td>UML</td>
</tr>
<tr>
<td>19</td>
<td>VBA</td>
</tr>
<tr>
<td>20</td>
<td>V&amp;V</td>
</tr>
</tbody>
</table>
An Automated Approach of Correctness and Completeness to Validate Requirements Specification

Abstract

Requirements is the first phase in software development process that should be taken into consideration, which is a milestone to achieve software success and quality, the difficulty in requirements is requirements changeability, which is hard to control by following the services and functions.

Meanwhile, it is important to pin down a set of formal customer requirements, which points out Correctness and Completeness, addressing them through customer satisfaction that revolves around precise services requests, to be conveyed to deep Completeness, through Consistency. Mentioning that Correctness and Completeness point at Quality, while enhancing the Requirements Specification increases its quality, which in turn reflects positively on the product quality.

This research addresses the problem of achieving Correctness, Completeness and Consistency in Requirements Specification through UML use case diagram. Applying an automated approach and a (VBA/MS excel) programmed tool based on standard rules to evaluate the approach. The evaluation is established using two standard use case diagrams from the UML site, resulted in improving the Correctness and Completeness criteria in Requirements Specification, and so the quality.

Keywords: Correctness, Completeness, Consistency, Quality-based requirements, Use case diagram, Requirements Specification.
CHAPTER ONE

INTRODUCTION

The terms; Correctness and Completeness (C&C) are two sides of one coin, which are involved in several phases of software processes. Meanwhile, they will be defined relying on the processes that exist. The initial phase in software development process is requirements that deals between customer and software developer, intended to system domain, and how to establish the product. Issues involved are abstraction and invisibility that appear clearly in quantification and measurement. Likewise, there are also complicated efforts to describe and organize software, in ways that will facilitate services change during the processes of their design, implementation, testing and maintenance.

Addressing incompleteness negatively influences the quality of produced artifact via requirements model, design model or the rest of software components via methodologies that are applied (Eckhardt et al., 2016; Fernández & Wagner, 2015; Hadad et al., 2015; Zowghi & Coulin, 2005; Coughlan & Macredie, 2002;). The investigated Completeness issue in software engineering especially, in requirements phase is that quality requirements Engineers; incorrectly describe how to build the system more than the functionality of the system (Firesmith, 2007).

The requirement phase consists of many stages, considering three of them in this research. Firstly, user requirements, which is collecting the services that the customer needs in the system presented by high-level requirements, as well called raw requirements (Swarnalatha et al., 2014; Düchting et al., 2007). Secondly, Requirements elicitation, is capturing the requirements which rely on emergent and collaborative view of requirements, elicitation and communication are both required to encompass the user to reduce error-prone requirements, that came from early stage from the user, as well as the purpose of requirements elicitation is to ensure successful requirements gathering (Coughlan & Macredie, 2002).

Thirdly, Requirements Specification (RS) stage named system requirements, is a detailed description of what the system should do, which are derived from the user requirements, and they are modeled using formal or semi-formal methods and languages (dos Santos Soares & Vrancken, 2008). In addition to Requirements Evaluation and Requirements Prioritizing stages of the requirements phase.
1.1 Problem Statement
One of the RS issues is Correctness facing the formal and informal view, which occurs effectively from customer requested services to control hierarchy (Larsson & Borg, 2014). On another hand, the Completeness issue is an upper level of Correctness to consider, taking Consistency among Correctness services (Kamalrudin & Sidek, 2015). So the problem is how to achieve a complete Requirements Specification, entailing correct and consistent requirements, through the UML use case diagram.

1.2 Research Questions
1) How to understand requirements C&C challenges in RS?
   The question will be answered by following the first objective, especially in chapter two, defining the keywords and the previous study.

2) How to address and control requirements challenges in the RS?
   This question will be answered through the second objective, in chapter three, by introducing the approach and the standard rules from UML and IBM sites.

3) Are requirements modifications able to be automated in the RS?
   The answer to this question will be found in chapter four, by applying the Correctness and Completeness in Requirements Specification Approach, and the programmed tool.

4) How does the approach conduct?
   The answer is in chapter four, experimenting the approach on two standard case studies from UML site, and making changes to falsify them to evaluate the approach.

1.3 Research Aim
To reach a level of C&C in the RS that minimizes the requirements modifications, to achieve clear and consistent functions due to quality.

1.4 Research Objectives
1) To investigate C&C in RS using historical review inner to Approaches, Tools, Frameworks and Techniques.
2) To introduce a conceptual approach, covering RS challenges from function aspect, using C&C criteria.
3) To develop the approach using a system application “Correctness and Completeness in Requirements Specification” that applied the C&C.
4) To evaluate the approach based on measurements of C&C through quality.
1.5 General Methodology

The general methodology of research design will be the basis of this research, to organize the steps and identify the processes to achieve the output (Hevner & Chatterjee, 2010).

![Diagram of general methodology]

Figure 1.1: The General Methodology of research design

Resource: (Hevner & Chatterjee, 2010)

**Step 1: Awareness of Problem**

Customers cannot get the big picture of the product even if it is according to their needs, further, they keep changing and modifying the requirements until they are satisfied with them. The awareness of the problem is how to minimize the modifications from customers, to achieve clear requirements using Correctness, Completeness and Consistency in requirements based on quality.

**Step 2: Suggestion**

In this step, the related work will be investigated in similar challenges and aim. Here, the investigation is through two factors, firstly, by keywords and terminology. Secondly, by historical publish. On the other hand, to find the elements that are involved in the new conceptual approach named, Correctness and Completeness in Requirements Specification.
Step 3: Development
To apply the conceptual approach based on a set of instructions using programming languages, as well as adopting tools to reach specific results.

Step 4: Evaluation
To determine the measurements for the conceptual approach, and compare them with other work in the same scope and objective.

Step 5: Conclusion
To find out the research contribution, which is minimizing the modifications in RS to evaluate it via C&C, and improve the quality of the product with less effort, cost and time.

1.5.1 Conceptual approach to evaluate RS from C&C criteria
The proposed conceptual approach shows an overview of the process that should be followed to achieve the preferable output with minimal customer modifications, it includes three levels: structural document, dynamic language and completeness, with two decisions which are: match (positive) and mismatch (negative), where the match decision determines the C&C of use case diagram elements (Actor, Use case and Relation), otherwise, it goes with C&C to the mismatch decision. The RS must be modified according to the match and mismatch decisions to establish Correctness, the cycle goes on until all the requirements are processed to Correctness, which leads to achieving Completeness as shown in figure 1.2.
Level 1: Structured Documents

In this level, the RS is presented first as a formal document of requirements phase, and contains what the customer needs to achieve from the system after negotiating with the developer, so it is the contract between the developer and the customer that is assumed as a reference, to check if needed requirements in the specification stage as a narrow document, are according to the customer needs.

A use case diagram is represented from the specification document, including their formal descriptions in a specific template.
Level 2: Dynamic Language
A dynamic language such as XML\(^{(1)}\) is used, to transform the formal data (flow of events) from the template into a textual understandable format, then the Actors, Use cases and Relations between them are acquired.

When comparing the flow of events in the use case description template to those in the specification document, the result will show that if they are similar then, we got a match which acquires correct requirement, but if mismatched then, the specification document or the use case diagram must be modified. Drawing to attention that use case diagrams will be used, in the first place, from benchmark to ensure the process quality.

Level 3: Completeness
The cycle of Correctness (match, mismatch) goes on, until all requirements needed by the customer in the RS are matched with the output of the approach, which means the achievement of a Complete RS.

1.6 Motivation
Requirements Specification is typically irregular, with specified requirements that are ambiguous, incomplete, inconsistent, incoherent, infeasible, obsolete, unable to be tested or validated and not usable by all of their intended audiences (Mead & Stehney, 2005), that should be controlled.

The C&C are millstones for life cycle inner to requirements engineering, which is important to point at in several aspects, as: reach the customer satisfied state as much as possible, minimize the effort and cost during phases development, achieving quality.

1.7 Significance
To achieve clear and correct requirements in the specification stage, taking into consideration the Consistency between each requirement and the other, to establish complete requirements of what the customer requested, which minimizes the potential customer modifications, getting satisfied as much as possible, and that contributes to reduction in cost, effort and time especially, from the later phases depending on the requirements phase.

\(^{(1)}\) XML language is suggested so that it is understandable by both users and computers (w3schools, 1998)
2.1 Overview
This chapter describes the issues regarding Correctness, Completeness and Quality in the requirements scope. It is divided into sections that incorporate discussion for a further explanation of C&C, as well as included similar challenges with different perspectives.

2.2 Keywords

2.2.1 Correctness
- Correctness is an application perspective that can be defined as, the adherence to the specifications that regulate how users can interact with the application, and how the application should behave when it is used appropriately, meanwhile, that application planned tasks as defined by its specification (Naeem et al., 2017).
- Software Requirements Specification (SRS) is correct, just if every requirement stated therein is one the software shall meet (Committee & Board, 1998).
- Incorrect: Some information in the SRS contradicts with other information, about the same or relevant information in the domain knowledge, or conflicts with preceding documents (Alshazly et al., 2014)
- According to Sommerville (2011) he defined Correctness term as “Ensuring the system services are as specified”.

2.2.2 Completeness:
- Functional Completeness: is how much does a set of functions covers all the specified tasks and user objectives (ISO/IEC, 2017).
- The term of Completeness is which subject data associated with an entity, has values for all expected attributes, and related entity instances in a specific context of use (ISO/IEC, 2008).
- The SRS is Complete (Committee & Board, 1998), if it includes the following elements:
  i. All the requirements related to functionality attributes in RS should be treated.
ii. Definition of the responses of the software to all realizable classes of input data, in all realizable classes of situations. Note that it is important to specify the responses to both valid and invalid input values.

iii. All the diagrams, labels, figures, term definitions and measures should be referenced and labeled.

- According to Sommerville (2011) he defined Completeness as "all services required by the user should be defined".

2.2.3 Quality (based on requirements)

- The standard name ISO/IEC 25021-Quality measure elements defined a set of recommended base and derived measures, which are intended to be used during the whole software development lifecycle. The document describes a set of measures that can be used as an input for the software product quality, or software quality in use measurement (ISO/IEC, 2007).

- The standard name ISO/IEC 2503n-Quality Requirements Division is one of the international standards that formed this division, to help specify quality requirements based on quality models and quality measures. These quality requirements can be used in the process of quality requirements elicitation, for a software product to be developed or as input for an evaluation process (ISO/IEC, 2016).

- The SRS is verifiable, if every requirement stated therein is verifiable. A requirement is verifiable, if there exists some finite cost-effective process, with which a person or machine can check that the software product meets the requirement. In general, any ambiguous requirement is not verifiable (Committee & Board, 1998).

- Non-verifiable requirements include statements such as “works well,” “good human interface,” and “shall usually happen.” These requirements cannot be verified because it is impossible to define the terms “good”, “well,” or “usually.” The statement that “the program shall never enter an infinite loop” is non-verifiable because the testing of this quality is theoretically impossible (Committee & Board, 1998).

- For traditional up-front RS, a number of standards have been defined on what constitutes good quality: requirements should be complete, unambiguous, specific, time-bounded, consistent, etc. (Heck & Zaidman, 2016).
• A quality requirement (Svensson et al., 2013) is further detailed into two types:
  i. Quantified Quality Requirements (QQR) type is a quality requirement with a direct quantification within the requirement.
  ii. Non-Quantified Quality Requirements (NQQR) type is a quality requirement without metrics.
• Quality has been defined as fitness for use, or the extent to which a product successfully serves the purposes of consumers (Jahanshani et al., 2014).
• Quality has been an elusive concept in the automotive industry. It is the totality of features and characteristics of a product or service, that bears on its ability to satisfy given needs (Jahanshani et al., 2014).
• According to Sommerville (2011) he defined Quality as: when talking about professional software quality, the customer and developer use the system separately from each other, so the quality is not only what the software does, it must include the behavior of the software while executing, the structure of the programs and the associated documents, which is reflected on what is called quality or nonfunctional requirements.

The conclusion, the three main keywords (Correctness, Completeness, and Requirements-Based Quality), were giving a brief description of their definitions and domains, these keywords as criteria will be verified and validated during the Proposed Approach, to be the measurements that are used to validate RS later on.

2.3 Related Work

According to Firesmith (2003) defined many characteristics that make requirements better, such as Correctness, Completeness, Consistency, feasibility, usability and many others, in a way that each characteristic benefits in RS and quality.

The research Zowghi and Gervasi (2004) introduced C&C in two points of view, (1) formal, that Correctness is a combination between Completeness and Consistency, (2) practical, that Correctness is satisfaction of a specific business goals needed by the customer, and presented review papers about Correctness, Completeness and Consistency relating them with real-life practice. Yet, proved that formally with some equations. Furthermore stated an automated tool from related work (Zowghi et al., 2001) and resulted proving C&C formally.
**Firesmith (2007)** detailed five common problems of requirements, explaining their negative consequences and solutions introducing the C&C manner, such as poor requirements quality, requirements not traced, inadequate requirements process and unprepared requirements engineers, as well he stated that these problems are interleaved with each other, that means if a company had one of the problems, it is probably had another interrelated problem.

An experiment of **Larsson and Borg (2014)** proposed ten challenges facing requirements engineering in aligning them to Verification and Validation (V&V), explaining each one and its effect on quality, one of them is defining complete requirements, stating that requirements changes continuously as new requirements arrive, so an audit is done which is documented in an audit log, the audit contains the challenges in the requirements like requirements conflicts and missing requirements, audits include representatives from developers, business analysts and testers at least. While change happens in audit like requirements conflict, for example, the business analysts perform a new audit for the same representatives to revise and refine the audit, documenting them in a new log, and this iteration is repeated until no change is found. Other examples of challenges proposed, are defining good verification process and verifying quality requirements.

**Jahanshani et al. (2014)** introduced the importance of the quality through the current research on Tata motors industrial company for automobiles in India, doing a questionnaire of more than 60 questions, some are derived from previous researches, the else is designed to evaluate customer service, product and loyalty, and analysing them with ANOVA test and SPSS16. They stated the relation between good quality and customer loyalty for the product and company, that customer service quality and product quality mostly affects the customer loyalty, yet when the customer gets what he wants as he wants he will be satisfied which leads to better quality, the reason for getting customer loyalty. And according to Gooofin and Price (1996), the importance of customer service comes from achieving better quality, more sales and income and competitive level in the market.

A systematic review of **Kamalrudin and Sidek (2015)** stated a review discussion on Correctness, Completeness and Consistency (3Cs) in the requirements validation process defining each criterion, then introduced traceability and its importance of keeping in touch with requirements, forward to analysis approach and divided it into two parts (1)
heuristic analysis which is subjective, (2) formal analysis which is mathematical, then after all produced a heat map to reveal the most commonly used approaches and methodologies.

Kalinowski et al. (2016) presented a study on incomplete and hidden requirements based on a survey called Naming the Pain in Requirements Engineering (NaPiRE) in Austria and Brazil from 14 and 74 companies respectively both using plan-driven (waterfall) and agile methodology (scrum), the stated many problems in requirements some of them are, missing qualifications of requirements engineering team members, lack of experience, missing domain knowledge, unclear business needs and poorly defined requirements and introduced solutions to them, but the most famous problem was incomplete and hidden requirements, occupied the top of the list in Austria and the second in Brazil.

Kuchta (2016) claimed that no one can guarantee absolute Completeness, so he stated metrics to measure weakness in SRS Completeness, the measurements were divided into direct measures counted in quasi-graph storage, and indirect measures calculated by some formulas. He took a sample of SRS document preparing three incremented versions and evaluated them; the first version was prepared very thoroughly of 69 functional requirements, and the evaluation results in valuing to Completeness, Consistency and Correctness from low to high respectively. Based on the first version, the second version with 36 functional requirements and the evaluation results in increasing Completeness, Consistency but decreasing of Correctness value and critical, exceptional and breakdown situations existed, which needed a functional requirement for each situation to prevent or fix, that resulted in 99 new functional requirements. The third version, formally unresolved goals such as needs, tasks and problems existed, the reason that laid Consistency increasing, which results in 30 new requirements, that evaluated in increasing Completeness, Consistency and Correctness. So Consistency has an impact on Completeness yet without quality or Consistency evaluation, about 2/3 of the functional requirements will be ignored, that affects the requirements specification to be incomplete.

Naeem et al. (2017) defined C&C under the quality umbrella, stating three problems of requirements with examples in web applications, and defined three strategies to solve them, furthermore stated a benchmark table as a guide for requirements engineers of the predefined problems and their solutions, which affects the quality through requirements negotiation.
2.4 Similar Approaches

2.4.1 Approaches

Swathi (2011) focused on software quality since it is a key value to the final product. Miss. G. Swathi and Dr. Ch GVN Prasad stated an interpretation of many requirements, each set has a kind of a problem to solve clarifying that the quality of the product, depends on the requirements initialized to specification document, and how the requirements volatility affects the Software Development Life Cycle by impacting the time, cost, effort and final product quality. They introduced some examples of poor requirements and clarifying them, coming to a result that requirements can be improved by paying attention to requirements activities, so “Effective Requirements Practices” is suggested to improve writing requirements in SRS.

Femmer et al. (2014) stated the requirements smells by doing an analysis a so-called light-weight which was based on the Natural Language of standard ISO/IEC/IEEE 29148(2) (ISO/IEC/IEEE, 2017), and applied the approach on two case studies taken from two different companies, which contained 339 requirements and 53 use cases extracted from 9 specifications of previous companies.

“Ambiguities or incomplete requirements specifications, can lead to time and cost overrun in the project. Requirements (bad) smells, which are concrete symptoms for a requirement artefact's quality defect.”

Gigante et al. (2015) stated quality and its importance by defining Correctness, Completeness and Consistency, and their effect on quality, yet proving Completeness against Natural Language (external context, or unstructured textual requirements), by adopting ontologies beside the Resource Description Framework (RDF) triplets, and other heuristics and Natural Language Processing (NLP) tasks, to verify the High-Level Requirements (HLR) against System Requirements (SR) representing external context, so if the difference between them is 0 then, a redundancy exists which doesn’t add any value, so the smaller the distance between them the more Completeness is achieved.

(2) https://edisciplinas.usp.br/pluginfile.php/1077344/mod_folder/content/0/iso-iec-ieee-29148-2011.pdf?forcedownload=1
Table 2.1: Comparison among Approaches

<table>
<thead>
<tr>
<th>Factor</th>
<th>Writing Software Requirements Specification Quality</th>
<th>A semantic driven approach for requirements verification</th>
<th>Rapid Requirements Checks with Requirements Smells: Two Case Studies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Domain</strong></td>
<td>Software Requirements Specification (SRS)</td>
<td>Unstructured textual requirements (Natural Language)</td>
<td>User Requirements</td>
</tr>
<tr>
<td><strong>Target</strong></td>
<td>Focusing on the relation between the requirements and project outcomes</td>
<td>Adopt ontologies to verify requirements Completeness against a so-called software “external context”</td>
<td>Applying techniques such as light-weight for detecting (bad) smells in requirements</td>
</tr>
<tr>
<td><strong>Tool</strong></td>
<td>HTML tags</td>
<td>Using ontologies and tools RDF triplets measure</td>
<td>light-weight technique (based on the ISO 29148(ISO/IEC/IEEE, 2017)natural language criteria)</td>
</tr>
<tr>
<td><strong>Result</strong></td>
<td>Software quality can be improved by paying attention to requirements activities</td>
<td>Achieving sufficient Completeness</td>
<td>Detect the violating requirements of some certain principles of Requirements Engineering, develop an implementation that use Part-Of-Speech (POS) tagging to detect these violating requirements</td>
</tr>
</tbody>
</table>

2.4.2 Tools

Nigam et al. (2012) developed a tool for identifying ambiguous words called “Ambiguity Detector”, for three types: 1) lexical ambiguity, 2) syntactic ambiguity, 3) syntax ambiguity, introducing that ambiguity problem rises from changeable or incomplete requirements from the customer, which affects the Software Requirements Specification (SRS), furthermore stated that unresolved ambiguities become bugs with fourteen times costlier to be fixed. The tool matches the Software Requirements Specification with the predefined ambiguous words, and results them with a column diagram and ambiguity percentage of the three types mentioned earlier, that helps reasoning the SRS Correctness and Completeness, and so the quality of the output product.

(3) https://edisciplinas.usp.br/pluginfile.php/1077344/mod_folder/content/0/iso-iec-ieee-29148-2011.pdf?forcedownload=1
Fu (2015) introduced a tool named Structured Object-Oriented Formal Language (SOFL) that does test cases based on Functional Scenario Form (FSF), which play the same path role in the program, and by setting input values to satisfy the pre-condition and applying the Functional Scenario, the evaluation is done on the conjunction and disjunction formally to get the result of Completeness, which means that all the requirements of the customer should be covered by the Functional Scenarios. An example of Flight Ticket Management system is introduced as a case for testing the Completeness, applying SOFL and FSF in a formal way setting two formulas; the first is for Correctness, and the second is for Completeness to rule the test, during the process of the tool, when the two formulas are set to true in the test cases the Completeness is achieved.

Table 2.2: Comparison among Tools

<table>
<thead>
<tr>
<th>Factor</th>
<th>Tool for Automatic Discovery of Ambiguity in Requirements</th>
<th>A tool supported testing approach to automatic checking if the Completeness of formal specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>SRS</td>
<td>Requirements Specification (RS)</td>
</tr>
<tr>
<td>Target</td>
<td>Highlight ambiguous words and gauge Correctness graphically by a snapshot</td>
<td>Achieve Completeness formally with two equations for C&amp;C</td>
</tr>
<tr>
<td>Tool</td>
<td>Ambiguity Detector</td>
<td>SOFL, FSF</td>
</tr>
<tr>
<td>Result</td>
<td>Catching the three types of ambiguities: Lexical, syntax and syntactic, and helps the analysts decide the ambiguity priority by providing them a percentage of the ambiguities.</td>
<td>If SOFL resulted in a positive value then there is a Completeness, else it is not, with an error message</td>
</tr>
</tbody>
</table>

2.4.3 Frameworks and Techniques

Génova et al. (2013) introduced a framework for measuring textual requirements, stating some indicators with the goal of “measure to improve” for some desirable requirements properties, such as (Verifiability, Validatability, Completeness, Consistency
and traceability), and the indicators are: Morphological indicators, Lexical indicators, Analytical indicators and Relational indicators explaining their elements like size, readability and some terms they use, classifying the measurements of these indicators into discrete level sets: High, medium and low, good, medium and bad, and so. He gave a weight to each indicator that he can evaluate, mentioning that it is more useful to consider “global quality measure” than a specific set supporting his theoretical part by a tool: Requirements Quality Analyzer (RQA) and compared it with other tools he included in the research.

Alshazly et al. (2014) introduced many techniques and case studies to detect defects in Requirements Specification (RS) to improve quality. Firstly, they headed to classifying the defect types in RS such as Margarido et al. did in his research, then to classifying requirements errors such as the research of Walia et al., furthermore defined many requirements defect detection techniques some of them are; 1) Checklist-Based-Reading technique (CBR), 2) Defect-Based-Reading technique (DBR), 3) Perspective-Based-Reading technique (PBR), compared between them by applying them on three Software Requirements Specification; 1) Automated Teller Machine Network, 2) Online National Election Voting system 3) Web Publishing System, and experimenting them on four case studies, coming out with results that each technique has its strengths and weaknesses in detecting defects in RS beside that, it is difficult to depend on a single technique for detecting defects, the reason a new technique has been proposed called combined-readig technique that dected more defects despite its disadvantages. What concerns the current research from the previous techniques is (PBR) technique, which concentrates on the use cases from many different perspectives of users in RS, to minimize faults overlaps found from the reviewers, although it doesn’t consider the requirements document components and the relations between them, anyway, PBR technique is not the best one to detect incorrect or inconsistent defects.
Table 2.3: Comparison among Frameworks and Techniques

<table>
<thead>
<tr>
<th>Factor</th>
<th>A Framework to Measure and Improve the Quality of Textual Requirements</th>
<th>Detecting defects in software requirements specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>User requirements</td>
<td>Software Requirements Specification (SRS)</td>
</tr>
<tr>
<td>Target</td>
<td>Measure textual requirements using some indicators to improve their quality</td>
<td>Detect defects in SRS and present a taxonomy of defect detecting techniques</td>
</tr>
<tr>
<td>Tool</td>
<td>Requirements Quality Analyzer (RQA)</td>
<td>CBR, PBR, RVTs, DVR, UBR</td>
</tr>
<tr>
<td>Result</td>
<td>Increase the Quality of requirements and so V&amp;V, increase the writing skills of Requirements Engineers</td>
<td>Connect the defects with their sources to ensure SRS quality, support the SRS writers with good techniques of detection and prevention</td>
</tr>
</tbody>
</table>

Eventually, Correctness, Completeness and Quality are valuable and important criteria to argue and prove, many definitions are assigned to them in standards and researches yet in different domains, the reason that encourages the research in this field, the way that improves the quality of requirements by collecting every need of the customer as he wanted.

### 2.5 Conclusion

Many standards and a lot of researches stated Correctness, Completeness and requirements Quality, taking into consideration the domain to define them, further to many approaches and tools which were introduced and developed, e.g. (Fu, 2015; Gigante et al., 2015) to improve requirements and final product quality. After validating the Correctness, Completeness and Quality (based on requirements) they will become measurements to use as a validation process in practice.

Refer to table 2.1 based on the comparison among approaches, all the researches achieved their scope results from the natural requirements or SRS domain, further to that some results were not the Completeness of the RS.

Refer to table 2.2 comparison among tools, the researches the targets were to achieve Completeness and pinpoint at quality through RS or SRS domain using equations.

Refer to the comparison in table 2.3 the researches concentrated on achieving quality.
CHAPTER THREE

METHODOLOGY

3.1 Overview
This section introduces the followed process of the Proposed Approach describing each level, starting with use case diagram and ending with the achievement of Completeness, using a programmed tool to verify the inner process of each level, to ensure clear achievement of C&C through the use case diagram based on UML.

3.2 Proposed Correctness and Completeness in Requirements Specification Approach
The proposed approach illustrated in figure 3.1 shows the levels that use case diagram will go through, to reach a result that C&C in the RS document is proved.

Figure 3.1: Proposed Correctness and Completeness in Requirements Specification Approach
The Proposed Approach consists of three levels:

3.2.1 **Level 1: Structured Document**
The initial execution in this level is a use case diagram presented by UML which is a dynamic form, each use case in the diagram has its scenarios entailing the flow of events, connected with Actors with specific types of relations.

3.2.2 **Level 2: Dynamic Language**
The use case diagram elements (Actor, Use case and Relation) will be extracted from the use case diagram, and whilst the process is to transform use case diagram as dynamic to text as a static representation that is possible to acquire these elements, the language XML⁴ is suggested. So the use case diagram as an initial for this level is decomposed into its elements, as mentioned earlier, which are Actor, Use case and Relation, identified by their IDs according to XML.

3.2.3 **Level 3: Completeness**
At the beginning of this level, each Use case is checked according to its formal description and by whom it is communicated, further to what is the type of relation that interacts between the two classifiers of the use case diagram, which is based on rules from “IBM Knowledge Center”⁵ (IBM, 2017) and the UML⁶ (UML, 2017) sites, so according to these rules, if the acquired elements match the use case diagram then the Correctness criterion will be achieved, if not then a feedback must be returned indicating to the developer to check the use case diagram or the RS, the cycle continues to matching all the elements, then the achievement of the Completeness criterion will be reached.

3.3 **Map-Rules**
The rules in UML models are used from “IBM Knowledge Center”⁵ (IBM, 2017) and the UML⁶ (UML, 2017) sites as standards, with instructions represented by pseudo code and a tool implemented code in Visual Basic for Application (VBA) on Microsoft Excel, to check the Correctness of the use case diagram elements:

i. **Actor:** in UML, is a “behaviored classifier” that plays a role interacting with the system boundary and connected to the Use case to gain services from. It can be a hardware or software or human(UML, 2017).

---

⁴ XML language is suggested so that it is understandable by both users and computers (w3schools, 1998)
⁵ https://www.ibm.com/support/knowledgecenter/SS8PJ7_9.5.0/com.ibm.xtools.modeler.doc/topics/crelsme_ucd.html
⁶ https://www.uml-diagrams.org/use-case-diagrams.html
ii. **Use case:** in UML, is a “behaviored classifier” that specifies a full useful functionality of an action that collaborates with one or more Actors, each Use case has a service to each Actor connected to it (IBM, 2017; UML, 2017).

iii. **Relationship:** In UML is a relation that connects two classifiers interacting with each other like use cases or classifiers, that describes the nature of the relationship rules, and appears as a solid line between the connected classifiers, arrow-headed with a dashed line or triangular arrowhead (IBM, 2017; UML, 2017).

### 3.3.1 Relationship

Contains three types of relations:

**I. Association:**

In UML, the association is a relation connecting two classifiers describing the relation reasons and rules, and it is represented as a straight line between the two classifiers (IBM, 2017).

i. Only binary associations are allowed.

ii. The association is between Actor and Use case or Use case and Use case.

iii. The actor could connect one or more Use cases.

iv. The use case may be connected to one or more Actors.

The table below shows the association between two Actors describing that no rule between two Actors is accepted but the Generalization.

**Table 3.1:** Represents the rules between two Actors in the Association relation

<table>
<thead>
<tr>
<th>Pseudo code (Actor to Actor)</th>
<th>Implemented VBA (Actor to Actor)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A is the first classifier</td>
<td>For xx1 = LBound(Sheet3.Range(&quot;A1:A200&quot;),Value) To UBound(Sheet3.Range(&quot;A1:A200&quot;),Value)</td>
</tr>
<tr>
<td>B is the second classifier</td>
<td>If (UCase(Left(Sheet4.Range(&quot;B&quot; &amp; xx1), Len(Sheet4.Range(&quot;B&quot; &amp; xx1)))) = &quot;ACTOR&quot; And UCase(Left(Sheet4.Range(&quot;E&quot; &amp; xx1)), Len(Sheet4.Range(&quot;E&quot; &amp; xx1)))) = &quot;ACTOR&quot;) Then</td>
</tr>
<tr>
<td>If A = “Actor” and B = “Actor” then check if Relation between A and B = “Generalization” then print “Right Generalization” Else Print “Wrong Relation or Classifier” End if</td>
<td>If UCase(Left(Sheet4.Range(&quot;C&quot; &amp; xx1)), Len(Sheet4.Range(&quot;C&quot; &amp; xx1)))) = &quot;GENERALIZATION&quot; Then</td>
</tr>
<tr>
<td></td>
<td>Sheet4.Range(&quot;F&quot; &amp; xx1) = &quot;Right Generalization&quot; Else</td>
</tr>
<tr>
<td></td>
<td>Sheet4.Range(&quot;G&quot; &amp; xx1) = &quot;Wrong Relation or Classifier&quot; End If</td>
</tr>
<tr>
<td></td>
<td>End If</td>
</tr>
<tr>
<td></td>
<td>End If</td>
</tr>
</tbody>
</table>
The relation between the Actor and the Use case is limited to Association named “Communicate” in the programmed tool:

Table 3.2: Represents the rules between Actor and Use case in the Association relation

<table>
<thead>
<tr>
<th>Pseudo code (Actor to Use case)</th>
<th>Implemented VBA (Actor to Use case)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A is the first classifier</td>
<td>If UCase(Left(Sheet4.Range(“B” &amp; xl), Len(Sheet4.Range(“B” &amp; xl)))) = “ACTOR” And UCase(Left(Sheet4.Range(“E” &amp; xl), Len(Sheet4.Range(“E” &amp; xl)))) = “USE CASE” Then</td>
</tr>
<tr>
<td>B is the second classifier</td>
<td>If UCase(Left(Sheet4.Range(“C” &amp; xl), Len(Sheet4.Range(“C” &amp; xl)))) = “COMMUNICATE” Then</td>
</tr>
<tr>
<td>If A = “Actor” and B = “Use case” then check if Relation between A and B = “Communicate” then print “right Communicate” Else Print “Wrong Relation or Classifier” End if</td>
<td>Sheet4.Range(“F” &amp; xl) = “Right Communicate” Else Sheet4.Range(“G” &amp; xl) = “Wrong Relation or Classifier” End If</td>
</tr>
</tbody>
</table>

In the case of the relation between two Use cases, there are no forbidden relations according to the pre-mentioned rules (IBM, 2017; UML, 2017), and table 3.3 shows the “Communicate” association, table 3.4 shows the “Include” association and table 3.5 shows the “Extend” association.
II. **Include**

In UML, it is a directed relation where one base use case includes the functionality of the included use case, and it appears as a dashed line with an open arrowhead pointing to the included use case (IBM, 2017; UML, 2017).

i. Do not have names, only the keyword “Include”, and if a name is added it appears beside the include connector.

ii. The relation is only between Use cases only, no Actors involved.

iii. Split large use cases into other use cases to simplify them.

iv. Extract the same behavior of more than one use case.

III. **Extend**

In UML, it is a directed relation between two use cases that the extension use case extends the behavior of the base use case. If the base use case is meaningful by itself then there is no need for the extension use case (IBM, 2017; UML, 2017).
i. Part of a use case is an optional system behavior.

ii. Executing a sub-flow under specific conditions.

iii. The possibility of inserting many behavior segments in a base use case.

iv. The relation is between use cases only.

Table 3.5: Represents the rules between Use case and Use case in the Extend relation

<table>
<thead>
<tr>
<th>Pseudo code (Use case to Use case)</th>
<th>Implemented VBA (Use case to Use case)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A is the first classifier</td>
<td>If UCase(Left(Sheet4.Range(&quot;C&quot; &amp; xx1), Len(Sheet4.Range(&quot;C&quot; &amp; xx1)))) = &quot;EXTEND&quot; Then</td>
</tr>
<tr>
<td>B is the second classifier</td>
<td>If UCase(Left(Sheet4.Range(&quot;B&quot; &amp; xx1), Len(Sheet4.Range(&quot;B&quot; &amp; xx1)))) = &quot;USE CASE&quot; And</td>
</tr>
<tr>
<td>check if the Relation between A and B</td>
<td>UCase(Left(Sheet4.Range(&quot;E&quot; &amp; xx1), Len(Sheet4.Range(&quot;E&quot; &amp; xx1)))) = &quot;USE CASE&quot; Then</td>
</tr>
<tr>
<td>=&quot;Extend&quot; then check if A=&quot;Use case&quot; and B=&quot; Use case&quot; then print “Right Extend”</td>
<td>Sheet4.Range(&quot;F&quot; &amp; xx1) = &quot;Right Extend&quot;</td>
</tr>
</tbody>
</table>

IV. Generalization

In UML it is a relation between at least two classifiers like generalization between classes, the child use case is a type of the parent (general) use case having the same relations and operations of the general use case and it is represented as a big triangular arrowhead pointing to the general use case (IBM, 2017; UML, 2017).

i. Can be between Actors only.

ii. Can be between Use cases only.

iii. Cannot be between Actor and Use case.

So table 3.6 shows the “Generalization” between two Actors, and table 3.7 shows the “Generalization” between two Use cases.

Table 3.6: Represents the rules between two Actors in the Generalization relation

<table>
<thead>
<tr>
<th>Pseudo code (Actor to Actor)</th>
<th>Implemented VBA (Actor to Actor)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A is the first classifier</td>
<td>If (UCase(Left(Sheet4.Range(&quot;B&quot; &amp; xx1), Len(Sheet4.Range(&quot;B&quot; &amp; xx1)))) = &quot;ACTOR&quot; And UCase(Left(Sheet4.Range(&quot;E&quot; &amp; xx1), Len(Sheet4.Range(&quot;E&quot; &amp; xx1)))) = &quot;ACTOR&quot;) Then</td>
</tr>
<tr>
<td>B is the second classifier</td>
<td>If UCase(Left(Sheet4.Range(&quot;C&quot; &amp; xx1), Len(Sheet4.Range(&quot;C&quot; &amp; xx1)))) = &quot;GENERALIZATION&quot; Then</td>
</tr>
<tr>
<td>If A= “Actor” and B=“Actor” then check if the Relation between A and B =&quot;Generalization&quot;) then print “Right Generalization” else print &quot;Wrong Relation or Classifier &quot;</td>
<td>Sheet4.Range(&quot;F&quot; &amp; xx1) = &quot;Right Generalization&quot; Else</td>
</tr>
<tr>
<td></td>
<td>Sheet4.Range(&quot;G&quot; &amp; xx1) = &quot;Wrong Relation or Classifier&quot; End If</td>
</tr>
<tr>
<td></td>
<td>End If</td>
</tr>
</tbody>
</table>

22
Table 3.7: Represents the rules between two Use cases in the Generalization relation

<table>
<thead>
<tr>
<th>Pseudo code (Use case to Use case)</th>
<th>Implemented VBA (Use case to Use case)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A is the first classifier</td>
<td>If UCase(Left(Sheet4.Range(&quot;B&quot; &amp; xx1), Len(Sheet4.Range(&quot;B&quot; &amp; xx1)))) = &quot;USE CASE&quot; And UCase(Left(Sheet4.Range(&quot;E&quot; &amp; xx1), Len(Sheet4.Range(&quot;E&quot; &amp; xx1)))) = &quot;USE CASE&quot; And UCase(Left(Sheet4.Range(&quot;C&quot; &amp; xx1), Len(Sheet4.Range(&quot;C&quot; &amp; xx1)))) &lt;&gt; &quot;EXTEND&quot; Then If UCase(Left(Sheet4.Range(&quot;C&quot; &amp; xx1), Len(Sheet4.Range(&quot;C&quot; &amp; xx1)))) = &quot;GENERALIZATION&quot; Then Sheet4.Range(&quot;F&quot; &amp; xx1) = &quot;Right Generalization&quot;</td>
</tr>
<tr>
<td>B is the second classifier</td>
<td></td>
</tr>
<tr>
<td>If A = “Use case” and B = “Use case” then check if the Relation between A and B = &quot;Generalization&quot; then print “Right Generalization”</td>
<td></td>
</tr>
</tbody>
</table>

3.4 Conclusion

By checking each element and the interaction in the use case diagram to be valid as it should be in the RS, the Correctness will be achieved, based on the definition in Chapter 2 in section 2.2.1, after doing the checking for all the diagram elements to be correct, the Completeness will be achieved, based on the definition in Chapter 2 in section 2.2.2.
CHAPTER FOUR

EXPERIMENT AND DISCUSSIONS

4.1 Experiment
The experiments show the effectiveness of the Correctness and Completeness in Requirements Specification approach, in a way that it can extract the use case diagram elements, to match them with the formal description of the use cases scenarios, that makes it possible for anyone to use the approach, as well as the requirements, as formal (RS).

4.2 Case studies:
The approach will experiment two use case diagrams were chosen from UML\(^{(7)}\) (UML, 2017) as case studies, the first is Hospital Reception and the second is Online Shopping.

4.2.1 Case study 1: Hospital Reception
The Hospital Reception is presented in the use case diagram representing the RS, from the UML site which is Level 1 (Structured Document) in the Approach.
The applied approach starts by transforming the use case diagram into XML format, opening the saved Rational Rose use case diagram from Visual Paradigm and exporting it to MS excel sheets format, so that each element has its ID to be identified, mentioning that the relation between the classifiers connecting their IDs, the XML is stored in excel sheet presenting the major items (columns) such as Element Stereotype, Element ID, Element Name, Relation Stereotype, Element Stereotype, (From) ID and (To) ID, referring to Appendix A.

\(^{(7)}\) https://www.uml-diagrams.org/use-case-diagrams.html
Each element in the diagram is transformed to details in the excel sheet, for Use cases and Actors; their IDs, names, stereotypes. For the Relations; they are presented by the IDs of the classifiers where they are connected from and to.

There are many details in the stored excel, but the items are selected and filtered as shown in figure 4.2:
In Figure 4.3 is the result of filtering the form in Figure 4.2 to be more understandable by removing the out of boundary data and collecting the elements of use case diagram as follows, which implements Level2 (Dynamic Language):

- **(Element ID)** is the identifier for each element.
- **(Element Stereotype)** is the UML type of each element.
- **((From) ID)** is the id of the source element that the relation is connected with.
- **((To) ID)** is the destination element that the relation is connected with.
- **(Classifier Name)** is the Actor or the Use case name in the use case diagram.
Figure 4.3: Filtered form for the in-boundary items

Showing the final result of acquiring C&C in figure 4.4, through applying the Approach, according to the rules in 3.3 which leads to achieving Level 3.
### 4.2.1.1. Experiment: wrong case Hospital Reception:

Some changes were made on the Hospital Reception use case diagram (colored in red) to falsify them, yet to ensure applying the UML rules and achieving the C&C.
Some of the modifications were made by IBM Rational Rose software and some were changed manually from the excel sheet, because some relations were unaccepted by Rational Rose such as Association between actors.

The right use case diagram is not a basis for the false one, which means that if any wrong case of use case diagram was evaluated through applying the C&C approach, it will detect the errors.

The wrong case will pass through the same Approach procedure of the right one before (figure 4.1), and by applying the approach, the results were as shown in figure 4.6:
According to the UML rules in section 3.3.1:

I. Rule 1 - table 3.1- Actor to Actor: the relation between two Actors as Association (communicate as named in the approach) is wrong, it should be only (Generalization), and in the wrong case study the wrong relations result is:

Figure 4.6: Error results after changing elements of (Hospital Reception)
I. Rule 1 - table 3.1- Actor to Actor: the relation between two Actors as an association (communicate as named in the approach), is only (Generalization) and in the wrong case study the wrong relation result is:

II. Rule 2 - table 3.2- Actor to Use case: the relation between the Actor and Use case is a usual association (communicate), and in the wrong case the relation in red shows the following error result:

Figure 4.7: Sample of applying rule 1 (Hospital Reception)

Figure 4.8: Sample of applying rule 2 (Hospital Reception)
4.2.1 Case study 2: Online Shopping

As shown in figure 4.9 from the UML site\(^8\) (UML, 2017) showing the use case diagram of the (Online Shopping), which is Level 1 in the Approach in section 1.7.1 representing the RS.

The same process as in (4.2.1 case study 1) the diagram is stored in excel after transforming it to XML, presenting the major items (columns) such as (Element Stereotype, Element ID, Element Name, Relation Stereotype, Element Stereotype, (From) ID, (To) ID), referring to Appendix B.

Figure 4.9: Online shopping use case diagram

Resource: (UML, 2017)

\(^8\) https://www.uml-diagrams.org/use-case-diagrams.html
As a previous case study named (Hospital Reception), the approach will apply on the Online Shopping use case diagram as follows:

After selecting the in-boundary columns from the stored excel sheet in Appendix B, the result in figure 4.8, is showing the classifiers names and stereotypes by their IDs and the relations names and stereotypes by their IDs connecting the classifiers.

![Excel Sheet](image)

**Figure 4.10:** Extracting in-boundary items to acquire use case diagram elements

The figure 4.11 shows the filtered results of form in figure 4.10 so it is simple to understand, by re-structuring the inner data, and acquiring the use case diagram elements covering by that, Level 2. Where the:

- **(Element ID)** is the identifier for each element.
- **(Element Stereotype)** is the UML type of each element.
- **((From) ID)** is the id of the source element that the relation is connected with.
- **((To) ID)** is the destination element that the relation is connected with.
- **(Classifier Name)** is the Actor or the Use case name in the use case diagram.
Figure 4.11: Filtered form for the in-boundary items

The element name with its stereotype is connected by its identified ID to the ID of another classifier as in the item (From (ID)) to the (To (ID)). Which must be compared to (Element ID) item, in other word the (From (ID)) item is compared to the (Element ID) item to select the source classifier row then (To (ID)) is compared to the (Element ID) to select the destination classifier row, then stating them in one row, as shown in figure 4.12:
The final result of acquiring C&C in figure 4.12, through applying the Approach, according to the rules in 3.3, which leads to achieving Level 3.

Figure 4.12: The result of applying C&C (Online Shopping)
4.2.1.2. Experiment: wrong case Online Shopping:
Changes were made on the Online Shopping use case diagram (colored in red) to falsify them, yet to ensure applying the UML rules and achieving the C&C.

Figure 4.13: Changing some elements of Online Shopping case study (in red color)

The wrong case will pass through the same Approach procedure of the right use case in figure 4.9, as aforementioned that some relations were modified by Rational Rose and some are modified manually, and by applying the approach the results were as shown next:
Figure 4.14: Error results after changing elements of (Online Shopping)
According to the UML rules in section 3.1.1:

Rule 1 - table 3.1- Actor to Actor: the relation between Actor and Actor as an association (communicate as named in the approach) is not permitted, it is only (Generalization), and in the wrong case study the wrong relations result:

Rule 2 - table 3.2- Actor to Use case: the relation between the Actor and Use case is a usual association named (communicate), and in the wrong case the generalization relation between the Actor and Use case in red, resulted in the following error:

Figure 4.14: Sample of applying rule 1 (Online Shopping)

Figure 4.15: Sample of applying rule 2 (Online Shopping)
Table 4.1 concludes the forbidden relations between the use case diagram elements, which are proved earlier. The resulted wrong relations were refused according to the used software rules, or to the IBM and UML official sites standard rules, which means that Correct Relation, with Correct Actor and Correct Use case, leads to Complete use case diagram involving Correct and Complete RS.

The right rules were proved earlier in figures 4.4 and 4.12

Table 4.1: Case studies rules discussion table

<table>
<thead>
<tr>
<th>#</th>
<th>Rule number</th>
<th>Case study #1 (Hospital Reception)</th>
<th>Case study #2 (Online Shopping)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I.</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>2</td>
<td>II.</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

The evaluation of the approach was achieved through the programmed tool based on the standard rules, applying them on the two standard case studies that showed the approach validation.
CHAPTER FIVE
CONCLUSIONS and FUTURE WORKS

5.1 Conclusions
This study addressed two main challenges; the first challenge was Correctness and Completeness that was covered in Requirements specification scope, the second challenge was Consistency that appeared among correctness for each requirement presented in UML especially among actors and use cases.

The Approach consisted of three levels; the first level was the Structured Document, showed the Requirements Specification, represented by use case diagram entailing the scenarios formal description, the second level was the Dynamic Language, which described transforming the use case diagram as dynamic to textual XML as static using specific software, then extracted use case diagram elements; Actor, Use case and Relation, and the third level was the Completeness, that checked the implemented standard rules, compared to the formal description of the use cases scenarios of the use case diagram in the first level, if not matched, the Requirements Specification or the use case diagram must then be modified, if matched for each requirement addressing Consistency among them, Completeness will be achieved.

The approach was implemented via VBA on MS-excel, on another hand, the supported tools that the studies used were XML due to IBM Rational Rose and Visual Paradigm.

The experiments used two case studies which are:

1- Hospital Reception (UML, 2017).
2- Online Shopping (UML, 2017).

The contribution of the study was to establish Correctness and Completeness with Consistency among. Concerning the RS, to minimize the customer modifications, to achieve quality.

This research addressed Correctness, Completeness and Consistency in RS through UML use case diagram, and the tool based on the standard rules, it was proved that C&C was improved in RS scope.
5.2 Answering the Research Questions:

1- How to understand requirements C&C challenges in RS?

Objective 1, to investigate C&C in RS using historical review inner to Approaches, Tools, Frameworks and Techniques. Covered in chapter two.

Find out the suitable elements that cover the problem of the research.

2- How to address and control requirements challenges in the RS?

Objective 2, to introduce a conceptual approach, covering RS challenges from function aspect, using C&C criteria. Covered in chapter three.

Resulted in introducing and discussing the standard rules between the use case diagram elements (Actor, Use case and Relation) and programming them through a tool to evaluate the approach.

3- Are requirements modifications able to be automated in the RS?

Objective 4, to develop the approach using an automated approach “Correctness and Completeness in Requirements Specification” that applied the C&C. Covered in chapter four.

The result was applying the approach and the tool on two case studies.

4- How does the approach conduct?

The answer is in chapter four, experimenting the approach on two standard case studies from UML site, and making changes to falsify them to evaluate the approach.

To evaluate the approach based on measurements of C&C through quality. Covered in chapter four.

Resulted in improving C&C criteria in RS, affecting the quality positively.
5.3 Achieved Objectives
This research included four objectives:

The first objective was achieved in chapter two named BACKGROUND AND RELATED WORK, which included two main sections; definition and previous work (Approaches, Tools, Frameworks and Techniques).

The second objective was achieved in chapter three named METHODOLOGY, and included two main sections: the first section was the Proposed Correctness and Completeness in Requirements Specification Approach, and the second was Map-Rules.

The third and fourth objectives were applied in chapter four named EXPERIMENT AND DISCUSSIONS, which included two sections: Experiment section covered the third objective, by the implemented Map-Rules that was picked from IBM knowledge center, the fourth objective is shown in the section: Case Studies that included two sub-sections, the first sub-section was Hospital Reception and the second was Online Shopping.

5.4 Limitations and Future work

1) Unable to verify Extend relationship through sub-rules.
2) Unable to adjust this approach on different methodology and technique such as together J.
3) Unable to fully enhance the automation of Correctness and Completeness to the rest of the software engineering life cycle like Design, Implementation and Test, to achieve Quality.
4) Any one of the above limitations could be a future work research area.
REFERENCES


*Design research in information systems, 9-22*.

modeler.doc/topics/crelsmc_udc.html

ISO/IEC. (2007). Quality Measurement Division 2502n
ISO/IEC. (2016). Quality Requirements Division 2503n (pp. 1).

Engineering -- Life Cycle Processes --Requirements Engineering P29148/D1, 
(March) 2017 pp. 1-95).

Jahanshani, A. A., Hajizadeh, G. M. A., Mirdhamadi, S. A., Nawaser, K., & Khaksar, S. 
M. S. (2014). Study the effects of customer service and product quality on 
customer satisfaction and loyalty.

Kalinowski, M., Felderer, M., Conte, T, Spínola, R., Prikladnicki, R., Winkler, D., . . . 
Wagner, S. (2016). Preventing incomplete/hidden requirements: reflections on 
survey data from Austria and Brazil. Paper presented at the International 
Conference on Software Quality.

Kamalrudin, M., & Sidek, S. (2015). A review on software requirements validation and 
consistency management. *International Journal of Software Engineering and Its 
Applications, 9*(10), 39-58.


Larsson, J., & Borg, M. (2014). *Revisiting the challenges in aligning RE and V&V: 
Experiences from the public sector*. Paper presented at the Requirements 
Engineering and Testing (RET), 2014 IEEE 1st International Workshop on.

(SQUARE) methodology* (Vol. 30): ACM.

Problems and Strategies for Web Applications. *Pakistan Journal of Engineering, 
Technology & Science, 6*(1).

Ambiguity in Requirements Tool for Automatic Discovery of Ambiguity in 
Requirements.

Cataloging-in-Publication Data.

requirements are specified in industrial practice. *Information and Software 
Technology* 1236-1224, (7)55.

on software requirements engineering for real time projects based on customer 
requirements. *Int’l J of Advanced Research in Computer and Communication 
Engineering, 3*(1).


### Figure A: An output based on XML exported to excel sheet (Hospital Reception)
Figure B: An output based on XML exported to excel sheet (Online Shopping)
Appendix C

Sub Norm()
Dim counter, i, j, r, col As Integer

Sheet1.Activate
Sheet1.Cells(1, 1).Select
counter = 1

Sheets("sheet2").Cells.ClearContents
Sheets("sheet3").Cells.ClearContents
Sheets("sheet4").Cells.ClearContents

For i = 3 To 200
    If Cells(i, 1).Value <> "" Then
        Else
        counter = counter + 1
    '************************* Columns needed ********************************
    Sheet2.Range("a" & counter) = Sheet1.Range("a" & i - 1)
    Sheet2.Range("b" & counter) = Sheet1.Range("b" & i)
    Sheet2.Range("d" & counter) = Sheet1.Range("d" & i)
    Sheet2.Range("e" & counter) = Sheet1.Range("e" & i)
    Sheet2.Range("f" & counter) = Sheet1.Range("f" & i)
    Sheet2.Range("g" & counter) = Sheet1.Range("g" & i)
    Sheet2.Range("h" & counter) = Sheet1.Range("h" & i)
    Sheet2.Range("l" & counter) = Sheet1.Range("l" & i)
    End If
Next

'********************* Delete empty rows *********************
Dim rr As Range, rows As Integer
Set rr = Sheet2.Range("A1:Z50")
rows = rr.rows.Count
For i = rows To 1 Step (-1)
    If WorksheetFunction.CountA(rr.rows(i)) = 0 Then rr.rows(i).Delete
Dim columns As Integer
Set rr = Sheet2.Range("A1:Z50")
columns = rr.columns.Count
For i = columns To 1 Step (-1)
    If WorksheetFunction.CountA(rr.columns(i)) = 0 Then rr.columns(i).Delete
Next

For i = LBound(Sheet2.Range("A1:A200").Value) To UBound(Sheet2.Range("A1:A200").Value)
    If UCase(Left(Sheet2.Range("D" & i), Len(Sheet2.Range("D" & i)))) = "<<COMMUNICATE>>" Or UCase(Left(Sheet2.Range("D" & i), Len(Sheet2.Range("D" & i)))) = "<<EXTEND>>" Or UCase(Left(Sheet2.Range("D" & i), Len(Sheet2.Range("D" & i)))) = "<<INCLUDE>>" Then
        Sheet2.Range("A" & i) = Sheet2.Range("D" & i)
        Sheet2.Range("D" & i) = ""
        Sheet2.Range("A" & i) = Mid(Sheet2.Range("A" & i), 3, Len(Sheet2.Range("A" & i)) - 4)
    Else
        Sheet2.Range("D" & i) = ""
    End If
    If IsNumeric(Sheet2.Range("B" & i)) Then
        Sheet3.Range("A" & i) = Sheet2.Range("B" & i)
    End If
    Sheet3.Range("B" & i) = Sheet2.Range("A" & i)
    Sheet3.Range("E" & i) = Sheet2.Range("C" & i)
    If UCase(Left(Sheet2.Range("A" & i), Len(Sheet2.Range("A" & i)))) = "EXTEND" Or UCase(Left(Sheet2.Range("A" & i), Len(Sheet2.Range("A" & i)))) = "INCLUDE" Then
        Sheet3.Range("C" & i) = Sheet2.Range("E" & i)
        Sheet3.Range("D" & i) = Sheet2.Range("F" & i)
    ElseIf UCase(Left(Sheet2.Range("A" & i), Len(Sheet2.Range("A" & i)))) = "GENERALIZATION" Then
        Sheet3.Range("C" & i) = Sheet2.Range("F" & i)
        Sheet3.Range("D" & i) = Sheet2.Range("G" & i)
    ElseIf IsNumeric(Sheet2.Range("F" & i)) Then
        Sheet3.Range("C" & i) = Sheet2.Range("F" & i)
        Sheet3.Range("D" & i) = Sheet2.Range("H" & i)
End If
Next

'------------------------------------------------------------------------------------------------------
Dim x, x1, xx1, y, y1, yy1, z, z1 As Integer
For x = LBound(Sheet3.Range("A1:A200").Value) To
UBound(Sheet3.Range("A1:A200").Value)
    For y = LBound(Sheet3.Range("A1:A200").Value) To
UBound(Sheet3.Range("A1:A200").Value)
        If Sheet3.Range("A" & x) = Sheet3.Range("C" & y) Then
            AAA:            If Sheet4.Cells(x, "D") = "" Then
                Sheet4.Range("D" & x) = Sheet3.Range("D" & y)
                Sheet4.Range("C" & x) = Sheet3.Range("B" & y)
                Sheet4.Range("K" & x) = Sheet3.Range("B" & y)
                If UCase(Left(Sheet3.Range("B" & x), Len(Sheet3.Range("B" & x)))) = "ACTOR"
Or UCase(Left(Sheet3.Range("B" & x), Len(Sheet3.Range("B" & x)))) = "USE CASE" Then
                    Sheet4.Range("A" & x) = Sheet3.Range("A" & x)
                    Sheet4.Range("B" & x) = Sheet3.Range("B" & x)
                    Sheet4.Range("J" & x) = Sheet3.Range("E" & x)
                End If
            Else
                Sheet4.Range("A" & x).EntireRow.Insert xlUp ' if a cell is not empty - insert new row
                GoTo AAA:
            End If
        End If
        If Sheet3.Range("A" & x) = Sheet3.Range("D" & y) Then
            BBB:            If Sheet4.Cells(x, "D") = "" Then
                Sheet4.Range("D" & x) = Sheet3.Range("C" & y)
                Sheet4.Range("C" & x) = Sheet3.Range("B" & y)
                Sheet4.Range("K" & x) = Sheet3.Range("B" & y)
                If UCase(Left(Sheet3.Range("B" & x), Len(Sheet3.Range("B" & x)))) = "ACTOR"
Or UCase(Left(Sheet3.Range("B" & x), Len(Sheet3.Range("B" & x)))) = "USE CASE" Then
                    Sheet4.Range("A" & x) = Sheet3.Range("A" & x)
                    Sheet4.Range("B" & x) = Sheet3.Range("B" & x)
                    Sheet4.Range("J" & x) = Sheet3.Range("E" & x)
                End If
            Else
                Sheet4.Range("A" & x).EntireRow.Insert xlUp ' if a cell is not empty - insert new row
                GoTo AAA:
            End If
        End If
    End If
Next
End If
Else
    Sheet4.Range("A" & x).EntireRow.Insert xlUp
    GoTo BBB:
End If
End If
Next
Next
'------------------------------- Delete Empty Rows -----------------------------
Dim IRow, iCntr As Long
For iCntr = IRow To 1 Step -1
    If Sheet4.Cells(iCntr, 1).Value = "" Then
        Sheet4.rows(iCntr).Delete
    End If
Next
'------------------------------- copy classifiers names to sheet4 ------------------
For x1 = LBound(Sheet3.Range("A1:A300").Value) To UBound(Sheet3.Range("A1:A300").Value)
    For y1 = LBound(Sheet3.Range("A1:A300").Value) To UBound(Sheet3.Range("A1:A300").Value)
        If Sheet3.Range("A" & x1) = Sheet4.Range("D" & y1) Then
            Sheet4.Range("E" & y1) = Sheet3.Range("B" & x1)
            Sheet4.Range("L" & y1) = Sheet3.Range("E" & x1)
        End If
    Next
Next
'------------------------------ Checking Use case diagram Elements Relations -----------------------------
    'A - A
    If UCase(Left(Sheet4.Range("B" & xx1), Len(Sheet4.Range("B" & xx1)))) = "ACTOR" And UCase(Left(Sheet4.Range("E" & xx1), Len(Sheet4.Range("E" & xx1)))) = "ACTOR" Then
If UCase(Left(Sheet4.Range("C" & xx1), Len(Sheet4.Range("C" & xx1)))) = "GENERALIZATION" Then
    Sheet4.Range("F" & xx1) = "Right Generalization"
Else
    Sheet4.Range("G" & xx1) = "Wrong R Generalization"
End If
End If

'A - U
If UCase(Left(Sheet4.Range("B" & xx1), Len(Sheet4.Range("B" & xx1)))) = "ACTOR" And UCase(Left(Sheet4.Range("E" & xx1), Len(Sheet4.Range("E" & xx1)))) = "USE CASE" Then
    If UCase(Left(Sheet4.Range("C" & xx1), Len(Sheet4.Range("C" & xx1)))) = "COMMUNICATE" Then
        Sheet4.Range("F" & xx1) = "Right Communicate"
    Else
        Sheet4.Range("G" & xx1) = "Wrong R Communicate"
    End If
End If
End If

If UCase(Left(Sheet4.Range("B" & xx1), Len(Sheet4.Range("B" & xx1)))) = "USE CASE" And UCase(Left(Sheet4.Range("E" & xx1), Len(Sheet4.Range("E" & xx1)))) = "ACTOR" Then
    If UCase(Left(Sheet4.Range("C" & xx1), Len(Sheet4.Range("C" & xx1)))) = "COMMUNICATE" Then
        Sheet4.Range("F" & xx1) = "Right Communicate"
    Else
        Sheet4.Range("G" & xx1) = "Wrong R Communicate"
    End If
End If
End If

'U - U
If UCase(Left(Sheet4.Range("B" & xx1), Len(Sheet4.Range("B" & xx1)))) = "USE CASE" And UCase(Left(Sheet4.Range("E" & xx1), Len(Sheet4.Range("E" & xx1)))) = "USE CASE" And UCase(Left(Sheet4.Range("C" & xx1), Len(Sheet4.Range("C" & xx1)))) <> "EXTEND" Then
    If UCase(Left(Sheet4.Range("C" & xx1), Len(Sheet4.Range("C" & xx1)))) = "COMMUNICATE" Then
        Sheet4.Range("F" & xx1) = "Right Communicate"
    Else
        Sheet4.Range("G" & xx1) = "Right Communicate"
    End If
End If
End If
If UCase(Left(Sheet4.Range("C" & xx1), Len(Sheet4.Range("C" & xx1)))) = "INCLUDE" Then
    Sheet4.Range("F" & xx1) = "Right Include"
Else
    If UCase(Left(Sheet4.Range("C" & xx1), Len(Sheet4.Range("C" & xx1)))) = "GENERALIZATION" Then
        Sheet4.Range("F" & xx1) = "Right Generalization"
    Else
        Sheet4.Range("G" & xx1) = "Wrong R Relation"
    End If
End If
End If
End If
End If

'Extend Relation ******************************

If UCase(Left(Sheet4.Range("C" & xx1), Len(Sheet4.Range("C" & xx1)))) = "EXTEND" Then
    If UCase(Left(Sheet4.Range("B" & xx1), Len(Sheet4.Range("B" & xx1)))) = "USE CASE" And UCase(Left(Sheet4.Range("E" & xx1), Len(Sheet4.Range("E" & xx1)))) = "USE CASE" Then
        Sheet4.Range("F" & xx1) = "Right Extend"
            If UCase(Left(Sheet4.Range("B" & xx1), Len(Sheet4.Range("B" & xx1)))) = "USE CASE" And UCase(Left(Sheet4.Range("E" & yy1), Len(Sheet4.Range("E" & yy1)))) = "ACTOR" And UCase(Left(Sheet4.Range("C" & yy1), Len(Sheet4.Range("C" & yy1)))) = "COMMUNICATE" Then
                If UCase(Left(Sheet4.Range("B" & yy1), Len(Sheet4.Range("B" & yy1)))) = "ACTOR" And UCase(Left(Sheet4.Range("E" & xx1), Len(Sheet4.Range("E" & xx1)))) = "USE CASE" And UCase(Left(Sheet4.Range("C" & yy1), Len(Sheet4.Range("C" & yy1)))) = "COMMUNICATE" Then
                    End If
                End If
            End If
        Next
    End If
End If
Next

'-------------------------------- Delete Empty Rows --------------------------------
For iCntr = IRow To 1 Step -1
    If Sheet4.Cells(iCntr, 1).Value = \\
        "" Then
        Sheet4.rows(iCntr).Delete
    End If
Next
Sheet4.Activate
End Sub
المنهج الآلي للتحقق من صحة واكتمال متطلبات البرمجيات

يزن الكسابرة

د. عايش الحروب

ملخص

هي عملية تطوير البرمجيات ينبغي بعين الاعتبار وهو لبنة أساسية لتحقيق البرمجيات وجودتها. إنها متغيرة، بحيث السيطرة عليها المهام.

يشير صحتها واكتمالها، ومعالجتها تنفيذ تحقيق

د. عايش الحروب

ويتناول تحقيق

وبناءً على المنهج

(تطبيق ( ( قياسية لتقليم نهج. يتم التقييم

والذي يوجه تحسين معايير

المنهج التخطيطي.