DEVELOPMENT OF A MODEL FOR CONFLICT RESOLUTION IN THE REQUIREMENTS ENGINEERING PROCESS OF SOFTWARE SYSTEMS

BY

Ishaya Peni GAMBO

B.Sc. Computer Science, Usmanu Danfodiyo University, Sokoto

M.Sc. Computer Science, Obafemi Awolowo University, Ile-Ife.

A THESIS PRESENTED TO THE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF TECHNOLOGY

OBAFEMI AWOLOWO UNIVERSITY

ILE-IFE, NIGERIA

IN PARTIAL FULFILMENT OF THE

REQUIREMENT FOR THE AWARD OF THE DEGREE OF

DOCTOR OF PHILOSOPHY (Ph.D.) IN COMPUTER SCIENCE

2016

CERTIFICATION

I, GAMBO Ishaya Peni with the registration number TP10/11/H/0358 in the Department of Computer Science and Engineering, Faculty of Technology, Obafemi Awolowo University, certify that this is an original research carried out by me under the supervision of:

______________________________ ________________

Professor H. A. Soriyan Date

Project Supervisor

______________________________ ________________

Dr. A. I. Oluwaranti Date

Chief Examiner

DEDICATION

This research work is dedicated to God Almighty who has made available the life, time and resources to start and complete this work successfully. Also, to my family for their love, patience, supports and prayers throughout the period of my studies and research.

ACKNOWLEDGEMENTS

I would first like to thank my motivating supervisor Prof. H. A Soriyan for all your help and support during the years leading up to the completion of this thesis. You had every right to give up on me, but you still stood by me. You have indeed done a lot for me. Your patience, objective criticism, guidance, and good advice are what made this possible. Thank you! Let me also acknowledge my indefatigable Head of Department, Dr. A. I. Oluwaranti for your good will, concern and significant role in getting this work done.

I would like to thank my beloved wife Mrs. Gambo Omobola for the spiritual, financial and moral support you gave all through the study. You are indeed God's gifts to me. You are a wonderful 'darling'! I will not forget to also acknowledge my two beautiful damsels: Silavatka Glory and Silaneska Grace. I equally want to acknowledge members of the Information Systems Research Group (ISRG): Dr. R.N. Ikono, Dr. R. G. Iroju, Miss T. O. Omodunbi, Mr. F. R. Famutimi, Mr. O. A. Afolabi, Mr. S. T. Yange, Mr. A. O. Olalekan, Mr. A. A. Adewunmi, Mr. A. Chris and others too numerous to mention on this page. Thanks for being there to support and criticise the work. Thanks a lot to Dr. B. I. Akhigbe for his great support.

I am not forgetting my lecturers who taught me at both M.Sc. and PhD level: Prof. E. R. Adagunodo, Prof. G. A. Aderounmu, Dr. O. A. Odejobi, Dr. B. S. Afolabi, Dr. A. O. Ajayi, Dr. E. A. Olajubu and Dr. P. A. Idowu. They all groomed me and made sure I measure up to the standard of my contemporaries in the international community. To all the other Lecturers in the Department, whose knowledge, contributions and positive criticisms have helped a lot; I deeply appreciates you all. Finally, I would like to thank my parents, brothers and sisters. You all supported me when things were hard and cheered me on when things went well.

AUTHORISATION TO COPY

OBAFEMI AWOLOWO UNIVERSITY, ILE-IFE, NIGERIA

HEZEKIAH OLUWASANMI LIBRARY

POSTGRADUATE THESIS

Author: Ishaya Peni **GAMBO**

Title: Development of a Model for Conflict Resolution in the Requirements Engineering Process of Software Systems

.

Degree: Doctor of Philosophy (Ph.D.) in Computer Science

Year: 2016

I, GAMBO Ishaya Peni, hereby authorise the Hezekiah Oluwasanmi Library to copy my thesis in part or in whole in response to request from individuals and/or organisations for the purpose of private study or research.

________________________________	____________________
Signature	**Date**

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ACRONYMS

AHP Analytic Hierarchy Process

AMI: Absolutely More Important

CASE: Computer aided Software Engineering

CBR: Cost Benefits Ranks

CMM: Capability Maturity Model

COCOMO : COnstructive COst MOdel

CORA: Conflict-Oriented Requirements Analysis

EBM: Evidence Based Medicine

EH: Extremely High

EI: Extremely Important

FMCDM: Fuzzy Multi Criteria Decision Making

FN: False Negative

FP: False Positives

GORA: Goal-Oriented Requirements Analysis

HOQ: House of Quality

ICT: Information Communication and Technology

IEEE: Institute of Electrical and Electronics Engineers

IHVN: Institute of Human Virology Pharmacy

INDEHELA: Informatics Development for Health in Africa

ISD: Information System Development

ISO: International Organization for Standardization

IT: Information Technology

KAOS: Knowledge Acquisition in Automated Specification

KcoC: Kendal Coefficient of Concordance

MINPHIS: Made in Nigeria Primary Healthcare Information System

MPARN: Multi-Criterion Preference Analysis for Systematic Requirements Negotiation

NFCs: Non-Functional Concerns

NFR: Non Functional Requirements

NFRF: Non-Functional Requirements Framework

NHIS: National Health Insurance Scheme

NLT: New Lanchester Theory

OAUTHC: Obafemi Awolowo University Teaching Hospitals Complex

PHC: Primary Healthcare

PhIS: Pharmacy Information System

PREView: *P*rocess and *R*equirements *E*ngineering *View*points)

QARCC: Quality Attribute Risk and Conflict Consultant

QFD: Quality Function Deployment

RCP: Requirements Change Process

rDscp: requirements Description

RE: Requirements Engineering

REP: Requirements Engineering Process

ReqCCR: Requirements Clustering for Conflict Resolution

rID: requirement Identity

RMFs: Requirements Measuring Factors

SCM: Software Configuration Management

SCOST: Software Cost Option Strategy Tool

SD: Software Development

SE: Software Engineering

SMI: Strongly More Important

SREM: Software Requirements Engineering Method

Stdev: Standard Deviation

TFNs: Triangular fuzzy numbers

TN: True Negatives

TP: True Positives

UML: Unified Modeling Language

VH: Very High

VSMI: Very Strongly More Important

WMI: Weakly More Important

WNDM: Weighted Normalized Decision Matrix

ABSTRACT

In this research a framework for conflict resolution was formulated, a model was designed, implemented and evaluated in the requirement engineering process of software development. This was with the view of identifying and resolving conflicts among multiple stakeholders' expectations that often arise during requirements analysis.

The research employed both qualitative and quantitative research approaches. Requirements were elicited from stakeholders using interview, case study, observation and secondary data. Ten (10) staff from the Pharmacy Department of Obafemi Awolowo University Teaching Hospitals Complex (OAUTHC) and five (5) software developers from software industry were the respondents. The requirements were analysed using the requirement filtering technique. Delphi method was used to resolve the inputs from the requirements. Clustering Algorithm was used to formulate the framework for conflict resolution. The model was specified in Unified Modeling Language, and the system was implemented using Java programming language. The system was tested for completeness and consistency using recall and precision as parameters.

The result showed the Kendal Coefficient of Concordance (KcoC) W was 0.000115598 $\approx$ 0.00. This indicated the existence of disagreement among the stakeholders' subjective views. With the clustering analysis, the result showed 5 clusters and their corresponding centroids and standard deviation. The analysis carried out indicated that cluster 3 is more reliable with the lowest standard deviation. This is because the data were clustered closely around the centroids. Cluster 3 had an average standard deviation of 0.61, while clusters 1, 2, 4 and 5 had high standard deviation of 0.95, 0.78, 0.86 and 1.31 respectively. Further analysis of the result revealed that 81.19% of all the attributes with the lowest standard deviation were in cluster 3, and 18.81% of the attributes with the lowest standard deviation were distributed in clusters 1, 2, 4, and 5. This also makes cluster 3 more effective. Based on the final resolution on cluster 3, a generated list of ordered requirements was produced. The output indicated the order of priorities finally assigned to each requirement. Seventy-seven (77) requirements were seen to be "very high", making it 76.24% of the entire sets of requirements, while twenty-four (24) requirements were finally seen to be "high", making it 23.76% of the entire sets of requirements. The model evaluation revealed that clusters 3 and 4 had 100% recall while clusters 1, 2 and 5 had 50%, 66.7% and 80% respectively. The model evaluation also indicated 100% precision in clusters 2, 3 and 5 respectively, while custers 1 and 4 had 66.7% and 78.95% respectively. The F-measure showed the harmonic mean of precision and recall to be 0.57, 0.80, 1.00, 0.88 and 0.89 for clusters 1, 2, 3, 4 and 5 respectively . Consequently, the F-Measure of cluster 3 with the value of 1.00 (100%) is most effective and reliable. This implies that all the instances originally belonging to cluster 3 were correctly clustered and all clustering results also belong to the original instances in cluster 3. The total sum of false negative (FN) in the model was 6.0 (14.29 %). This means the model is reliable with an accuracy level of 85.71%.

In conclusion, the study opined that resolving conflicting views of stakeholders' in software development is crucial and vital in order to reduce the cost of software developmen

CHAPTER ONE

INTRODUCTION

1.1 Synopsis

In life, everybody makes one form of decision or the other. In doing this, requirements specification is inevitable. Therefore, Requirements Engineering Process (REP) of software system is needful for software systems to be successful. This is because the software industry is seen as one of the fastest growing and most profitable industry due to its profound impact at the societal level (Arif *et al.,* 2010; Alshazly *et al.,* 2014). Despite many failures with some of these systems that are well publicised and in use; the re-engineering of such systems at the requirement level has improved consistently over the past few decades. Of particular interest is the fact that the success of a software system depends on how well it fits the needs of its users and its environment (Parnas, 1998, Nuseibeh and Easterbrook, 2000).

Software requirements comprise the needs of the user(s). Thus every system needs requirement definition. In a software development project, requirements are stakeholders statements concerning the system under development that when implemented will satisfy user needs. These requirements are determined and agreed to by the users (who are the legitimate sources of requirements) before the software can be built. Requirements express the final capabilities of a system. It is also the non-functional properties of a system that meets the users' needs and are meant to perform a specific set of tasks (within a defined scope). In addition, errors in these requirements are the most expensive to fix when found during the development process of any system (Heeks, 2002 and 2006).

In the context of Information System Development (ISD) and Software Engineering (SE), many approaches to software project development process have been formulated. In spite of their differences, virtually all of them include the Requirements Engineering (RE) phase. This is because, requirement related problems, during the elicitation and analysis phases in software development projects, are perceived as the three top-most reasons for a project being challenged, and the two top-most reasons for system failure (Standish Group, 1995).

Requirements define the '*what*' of a software product. Specifically, it defines what the software must *do* to add value to its users. It also means what the software must *be* to add value to its users; and what *limitations* there are on the choices developers have made when implementing the software. Therefore, requirements are pervasive. They continuously affect both the development and maintenance phases of a system's life.

Requirements are key objectives in development. They need to be precise and unambiguous for software to be consistent in behaviour. In reality, requirements should be both complete (that is they should include descriptions of all that is required to function) and consistent. There should be no conflict(s) or contradictions in the descriptions of the system (Baxter and Sommerville, 2011). However, inconsistencies often arise when there are multiple viewpoints that are conflicting and are embodied in the requirement (Rodrigues *et al.,* 2004). This is a common occurrence during the elicitation and analysis phases of software development.

Human needs are insatiable, unpredictable and quite varied. As such, conflicts arise almost inevitably as users and other stakeholders frequently pursue mismatching goals (Boehm *et al.,* 2000). Consequently, conflict becomes an inevitable part of requirement(s) elicitation and analysis. A conflict resolution strategy that controls and monitors divergence in the behaviour of a software is therefore required. Such resolution strategy will employ scientific methods that can take on challenges from its roots. The strategy should involve theories and methods that can improve the understanding of conflict. It should also underpin a collective practice that is focus on conflict reduction and the enhancement of software development processes (Babbitt and Hampson, 2011).

The term conflict in the context of this work is the disagreement between two or more viewpoints on some decision or values that are proposed in a design (Pruitt, 2013). These disagreements are due to the differing needs that lead to incompatible preferences among the alternatives under consideration (Harrington *et al.,* 1995). It is an interaction of people who acknowledge same object, but different views and values that are capable of interfering with the object. In software development, requirements conflict is the activity that exists among users. Conflict resolution during requirement(s) gathering should be handled by stakeholders and the requirements engineers. With proper management, the activities carried out in the resolution of conflict can be harnessed to solve problems at the requirements engineering phase.

Conflicts in requirements occur on the basis of goals and desires during project development at the RE phase (Easterbrook *et al.,* 1994). The importance of handling conflicts in requirements is well-known in practice and has been acknowledged by the RE research community (van Lamsweerde *et al.,* 1998; Fuxman *et al.,* 2001).

RE is the process by which users' requirements are determined (Cheng and Atlee, 2009). In Hull *et al.,* (2011) RE is "the subset of systems engineering concerned with discovering, developing, tracing, negotiating, analysing, prioritising, qualifying, communicating and managing requirements that define the system at successive levels of abstraction". It covers all of the activities involved in discovering, documenting, and maintaining a set of requirements for a software system. This suggests that RE is not just a single phase of software development process. It is a complex process that comprises