THE IMPACT OF USE CASE FORMAT ON THE UNDERSTANDING OF SYSTEM REQUIREMENTS BY NOVICE AND EXPERIENCED USERS

BALSAM A. MUSTAFA

A thesis submitted in fulfilment of the requirements for the award of the degree of Doctor of Philosophy (Computer Science)

Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia

JULY 2009
To My Father with Love
ACKNOWLEDGEMENT

Alhamdulillah! Finally I have finished my thesis! However, the completion of my thesis would not have been possible without the assistance of many people who gave their support in different ways. I would like to thank my supervisors: Prof. Safaai Bin Deris and Prof. Erik Arisholm, for your suggestions, continued encouragement, and patience to guide me through my research. I have learned a lot from you. I would like to thank Prof. Tajudin from the Faculty of Education at the University of Technology Malaysia for discussing with me the theories of Cognitive Psychology related to my research at the early days of my study.

A very special thank to Prof John Sweller from the University of New South Wales in Australia, he never disappointed me and was generous with me by his time and experience. My thanks and appreciation to Dr. Andrew Burton-Jones from the University of British Colombia in Canada for his continued encouragement and invaluable suggestions and discussions during this work, his responses to my emails at the difficult times were of great help.

I would like to express my gratitude to my family, my father and sisters. I have always needed to work hard to achieve my goals in life and they have always been there for me as an unwavering support. I dedicate this work to my father, to honor his love, prayers, and support during years.
ABSTRACT

The effective analysis and specification of requirements is critical in software development. Faults in the requirements may later have significant impact on the quality of the software system. Ineffective communication between users and developers is a major cause of failures of software projects. Use case model is a powerful and widely recognized tool for elicitation and specification of functional software requirements in object oriented methodology. It has been advocated as a way to negotiate and communicate requirements between system analysts and stakeholders. However, issues concerning the format, level of details and the communication capability of use cases are still unclear and debatable. This study uses theories from cognitive psychology on human understanding to derive hypotheses on the effect of the format of use case model on user understanding. In this study, comprehension of the functional requirements are compared between experienced and novice users. Particularly, the effect of differences in use case format on novice and experienced users performance in both familiar and unfamiliar domains were explored and if combining the textual description of a use case with diagrams of different levels of detail improves their understanding. Two controlled experiments were conducted; one to assess the performance of novice users, the other to assess more experienced users. The results of both experiments provide evidence that support the propositions that individuals who view text with use case diagram (simple or detailed) will develop higher level of understanding of the system requirements in less time when compared to individuals who view a text only model. The results of both experiments provide no evidence that support the propositions of the benefit of the simple diagram for improving novice users understanding, and the detailed diagram for aiding experienced users when combined with the text description. It is also found that neither the observed level of prior domain knowledge nor the observed level of analysis method knowledge has a significant effect on the level of “understanding” that users developed regarding a system requirements. Finally, our analysis shows no considerable differences in performance in the experiments tasks between novice and experienced users, which mean that the effect of experience on users understanding is still an open issue and needs further research in the future.
ABSTRAK

# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECLARATION</td>
<td>iii</td>
<td></td>
</tr>
<tr>
<td>DEDICATION</td>
<td>iv</td>
<td></td>
</tr>
<tr>
<td>ACKNOWLEDGMENT</td>
<td>v</td>
<td></td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>vi</td>
<td></td>
</tr>
<tr>
<td>ABSTRAK</td>
<td>vi</td>
<td></td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>vii</td>
<td></td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xiii</td>
<td></td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xvi</td>
<td></td>
</tr>
<tr>
<td>LIST OF APPENDICES</td>
<td>xviii</td>
<td></td>
</tr>
<tr>
<td><strong>1</strong> INTRODUCTION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>Root Causes of Software Faults</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Why do Software Projects Fail?</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>Information Requirements Determination</td>
<td>5</td>
</tr>
<tr>
<td>1.4</td>
<td>Conceptual Modeling</td>
<td>6</td>
</tr>
<tr>
<td>1.5</td>
<td>Human Factors in Information Systems Development</td>
<td>7</td>
</tr>
<tr>
<td>1.6</td>
<td>Motivation for the Research</td>
<td>8</td>
</tr>
<tr>
<td>1.7</td>
<td>Research Objectives</td>
<td>9</td>
</tr>
<tr>
<td>1.8</td>
<td>The Research Scope</td>
<td>10</td>
</tr>
<tr>
<td>1.9</td>
<td>Research Implications</td>
<td>11</td>
</tr>
<tr>
<td>1.9.1</td>
<td>Theoretical Perspective</td>
<td>11</td>
</tr>
<tr>
<td>1.9.2</td>
<td>Practical Perspective</td>
<td>12</td>
</tr>
</tbody>
</table>
## 2 EMPIRICAL RESEARCH IN SYSTEMS ANALYSIS METHODS

2.1 The Domain Description Challenge 14
2.2 Creating and Interpreting a representation Model 15
2.3 Why Empirical Evaluation? 16
2.4 Empirical Evaluation of Modeling Methods 16
2.5 Previous Empirical Comparisons 17
   2.5.1 Subjects Used in Measurements 21
   2.5.2 Type of Measurements 22
   2.5.3 Measuring Techniques’ Performance 24
2.6 Previous Empirical Studies on Experts and Novices 25
2.7 Empirical Foundations 27
2.8 Empirical Components of The Study 31
2.9 Focus on Model Interpretation 32
2.10 Summary 33

## 3 THE UNIFIED MODELING LANGUAGE AND USE CASES

3.1 Object Orientation and UML 34
   3.1.1 Research on UML 36
3.2 Use Case Technique 37
   3.2.1 History and Definitions 37
   3.2.2 Functional and Non Functional Requirements 40
   3.2.3 The Basic Constructs of Use Case Modeling 40
   3.2.4 Use Case Format 41
   3.2.5 Use Case Diagram Constructs 44
   3.2.6 Motivation for Use Cases 48
   3.2.7 Use Cases and Structured Methods 49
   3.2.8 The Content of Use Case 50
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.9</td>
<td>The Style of Use Cases</td>
<td>51</td>
</tr>
<tr>
<td>3.2.10</td>
<td>Use Cases as a Communication Mechanism</td>
<td>53</td>
</tr>
<tr>
<td>3.3</td>
<td>Summary</td>
<td>54</td>
</tr>
<tr>
<td>4</td>
<td><strong>THEORETICAL BACKGROUND AND HYPOTHESES</strong></td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Theoretical Foundation from Cognitive Psychology</td>
<td>55</td>
</tr>
<tr>
<td>4.2</td>
<td>Human Cognitive System and Comprehension</td>
<td>56</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Theory of Cognitive Informatics (CI)</td>
<td>57</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Long Term Memory, Short Term Memory, and Schema Acquisition</td>
<td>57</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Schema Theory (What is a Schema?)</td>
<td>58</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Cognition with Text and Diagram</td>
<td>60</td>
</tr>
<tr>
<td>4.2.5</td>
<td>Functions of Diagram in Learning</td>
<td>61</td>
</tr>
<tr>
<td>4.2.6</td>
<td>The Conjoint Retention Effect</td>
<td>62</td>
</tr>
<tr>
<td>4.2.7</td>
<td>Dual Coding Theory and Multimedia</td>
<td>63</td>
</tr>
<tr>
<td>4.2.8</td>
<td>Active Learning with Multimedia</td>
<td>64</td>
</tr>
<tr>
<td>4.2.9</td>
<td>Which Kind of Diagram, Simple or Complex?</td>
<td>65</td>
</tr>
<tr>
<td>4.3</td>
<td>Cognitive Load Theory</td>
<td>67</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Cognitive Load Sources</td>
<td>67</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Managing Cognitive Load</td>
<td>69</td>
</tr>
<tr>
<td>4.3.3</td>
<td>External Management of Cognitive Load</td>
<td>69</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Element Interactivity Related to this Study</td>
<td>71</td>
</tr>
<tr>
<td>4.3.5</td>
<td>Techniques to Reduce Element Interactivity</td>
<td>71</td>
</tr>
<tr>
<td>4.3.6</td>
<td>Effects of Individual’s Prior Knowledge</td>
<td>74</td>
</tr>
<tr>
<td>4.4</td>
<td>Summary</td>
<td>76</td>
</tr>
<tr>
<td>5</td>
<td><strong>METHOD</strong></td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Study 1: Novice Users</td>
<td>77</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Method</td>
<td>78</td>
</tr>
</tbody>
</table>
5.2 Study 2: Experienced User

5.2.1 Method

5.3 The Tool of Analysis

5.4 Summary

6 THE IMPACT OF USE CASE FORMAT ON NOVICE END USER UNDERSTANDING

6.1 Preliminary Statistics of Experiment 1

6.1.1 Scale Variables: Knowledge of Domain, Method, and Ease of Use

6.1.2 The Relationship between Process and effort Measures

6.2 Preliminary Statistical Procedure for Experiment 1

6.2.1 Demographics

6.2.2 Testing for an Ordering Effect

6.2.3 Correlation between Dependent Variables

6.2.4 Correlation between Dependent Variables and Covariates

6.2.5 Normality of Dependent Variables Distribution

6.3 Analysis and Results

6.3.1 Why Use MANOVA?

6.3.2 Assessing Assumptions in MANCOVA

6.3.3 Procedure for Analyzing MANCOVA Results

6.3.4 Detailed Analysis

6.3.5 Analyzing Differences between Groups in MANCOVA

6.4 Summary
THE IMPACT OF EXPERIENCE ON UNDERSTANDING USE CASE MODELS
BY EXPERIENCED USERS

7 7.1 Preliminary Statistical Procedure for Experiment 2 147

7.1.1 Demographics 148

7.1.2 Testing for an Ordering Effect 150

7.1.3 Correlations between Dependent Variables 152

7.1.4 Correlations between Dependent Variables
And Covariates 154

7.1.5 Normality of Dependent Variables Distributions 155

7.2 Detailed Analysis 160

7.3 The Effect of Experience 173

7.4 Summary 178

8 DISCUSSION

8.1 Instruments Used in the Experiments 179

8.1.1 Procedures of Developing Research Instruments 180

8.1.2 The Requirements Verification Test 181

8.1.3 More Procedures 183

8.2 Instruments Validity 184

8.2.1 Construct Validity 184

8.2.2 Statistical Conclusion Validity 186

8.2.3 Internal Validity 187

8.3 Reliability 189

8.4 Comparing with Recent Research 189

9 CONCLUSIONS

9.1 Results of the Studies 192
9.2 Limitations of the Research 193
9.3 Contribution and Future Research 195

References 198

Appendices 211-267
<table>
<thead>
<tr>
<th>Table No.</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Factors Affecting System Development Failure</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Summary of Previous Empirical Comparisons</td>
<td>18</td>
</tr>
<tr>
<td>2.2</td>
<td>Summary of Empirical Studies on Expert and Novice Performance</td>
<td>26</td>
</tr>
<tr>
<td>5.1</td>
<td>3x2 Factorial Design</td>
<td>80</td>
</tr>
<tr>
<td>5.2</td>
<td>Ease of Use Post-test Questions for Experiment 1 and 2</td>
<td>88</td>
</tr>
<tr>
<td>6.1</td>
<td>Explanation of Variables Used in the Study</td>
<td>103</td>
</tr>
<tr>
<td>6.2</td>
<td>Reliability Analysis for Domain Knowledge Scale</td>
<td>105</td>
</tr>
<tr>
<td>6.3</td>
<td>Reliability Analysis for Scale Variable: Knowledge of Analysis Methods</td>
<td>107</td>
</tr>
<tr>
<td>6.4</td>
<td>Reliability Analysis of the Scale Variable Ease of Use</td>
<td>108</td>
</tr>
<tr>
<td>6.5</td>
<td>Descriptive Statistics for Demographic Variables (84 participants)</td>
<td>110</td>
</tr>
<tr>
<td>6.6</td>
<td>Comparison of Demographic Means across Treatment Groups</td>
<td></td>
</tr>
<tr>
<td>6.7</td>
<td>Test for Case Order Effects on Means of Dependent Variables (ANOVA)</td>
<td>113</td>
</tr>
<tr>
<td>6.8</td>
<td>Correlation Coefficients for Product Dependent Variable</td>
<td>114</td>
</tr>
<tr>
<td>6.9</td>
<td>Correlation Coefficients for Effort Dependent Variables</td>
<td>115</td>
</tr>
<tr>
<td>6.10</td>
<td>Correlations between dependent variables and Covariates</td>
<td>117</td>
</tr>
<tr>
<td>6.11</td>
<td>Normality measures of Dependent Variables</td>
<td>119</td>
</tr>
</tbody>
</table>
6.12 Kolmogorov-Smirnov z Test across Three Treatments
6.13 Summary of preliminary Results

6.14 Summary of Research Hypotheses
6.15 Means and Std. dev. for Dependent Variables of ATM case
6.16 Means and Std. Dev. for Dependent Variables of HSS case
6.17 ANOVA Analysis of the Covariates across Three Treatments
6.18 MANCOVA Analysis for both Cases ATM and HSS
6.19 Tests Associated with MANCOVA Analysis
6.20 Post hoc Univariate Analysis for Dependent Variables of ATM Case
6.21 Post hoc Univariate Analysis for Dependent Variables of HSS Case
6.22 Pairwise Comparisons of both Cases (ATM&HSS)
6.23 Cohen’s d of Dependent Variables of both Cases
7.1 Descriptive Statistics for Experiment 2 (30 participants)
7.2 Comparison of Demographic Means across Treatment Groups

7.3 Test for Case Order Effects on Means of Dependent Variables (ANOVA)

7.4 Correlation Coefficients of Dependent Variables for Product
7.5 Correlation Coefficients of Dependent Variables for Effort
7.6 Correlations between Dependent Variables and Covariates
7.7 Normality Tests of Dependent Variables
7.8 Other Normality Tests of Dependent Variables
7.9 Summary of Preliminary Statistics Results
7.10 Means and Std. Dev. for Dependent Variables of ATM Case
7.11 Means and Std. Dev. for Dependent Variables of HSS Case
7.12 ANOVA Analysis for Covariates and Ease of Use across Three Treatments
7.13 MANCOVA Analysis for both Cases ATM and HSS 165
7.14 Tests Associated with MANCOVA Analysis 166
7.15 Post hoc Univariate Analysis for Dependent Variables of ATM Case 168
7.16 Post hoc Univariate Analysis for Dependent Variables of HSS Case 168
7.17 Pairwise Comparisons of both Cases (ATM & HSS) 170
7.18 Cohen’s d of dependent variables of both cases 173
7.19 Pairwise Comparison of the First Dependent Variable in ATM case 174
7.20 Pairwise Comparison of the Second Dependent Variable in ATM case 174
7.21 Pairwise Comparison of the Third Dependent Variable in ATM case 175
7.22 Pairwise Comparison of the Fourth Dependent Variable in ATM case 175
8.1 Multi-Trait, Multi Method Analysis 185
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure No.</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Chaos Report (2003)</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>CHAOS survey results 2004</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Components of Learning Process</td>
<td>28</td>
</tr>
<tr>
<td>2.2</td>
<td>A framework for a Cognitive Theory of Multimedia Learning</td>
<td>30</td>
</tr>
<tr>
<td>3.1</td>
<td>Use Case Text Style</td>
<td>42</td>
</tr>
<tr>
<td>3.2</td>
<td>Example Use Case Diagram</td>
<td>44</td>
</tr>
<tr>
<td>3.3</td>
<td>Extend Relationship</td>
<td>46</td>
</tr>
<tr>
<td>4.1</td>
<td>An information processing model</td>
<td>58</td>
</tr>
<tr>
<td>4.2</td>
<td>A cognitive theory of multimedia learning</td>
<td>64</td>
</tr>
<tr>
<td>5.1</td>
<td>Experiment Design Model</td>
<td>79</td>
</tr>
<tr>
<td>5.2</td>
<td>An Overview of Empirical Procedures</td>
<td>84</td>
</tr>
<tr>
<td>6.1</td>
<td>Histogram &amp; Q-Q Plots of the Dependent Variables of Treatment (1) ATM case</td>
<td>120</td>
</tr>
<tr>
<td>6.2</td>
<td>Histogram &amp; Q-Q Plots of the Dependent Variables of Treatment (1) HSS case</td>
<td>121</td>
</tr>
<tr>
<td>6.3</td>
<td>The Assessment Process</td>
<td>126</td>
</tr>
<tr>
<td>6.4</td>
<td>Means for Dependent Variables of ATM Case</td>
<td>135</td>
</tr>
<tr>
<td>7.1</td>
<td>Histogram &amp; Q-Q Plots of the Dependent Variables of Treatment (1) ATM case</td>
<td>157</td>
</tr>
<tr>
<td>7.2</td>
<td>Histogram &amp; Q-Q Plots of the Dependent Variables of Treatment (1) HSS case</td>
<td>158</td>
</tr>
<tr>
<td>7.3</td>
<td>Means Graphs for Dependent Variables of ATM</td>
<td></td>
</tr>
</tbody>
</table>
7.4 The Interaction Effect between Treatments and Experience
# LIST OF APPENDICES

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Pre-test questionnaire</td>
<td>211</td>
</tr>
<tr>
<td>B</td>
<td>Post-test questionnaire</td>
<td>212</td>
</tr>
<tr>
<td>C</td>
<td>Comprehension Test Models (ATM) case</td>
<td>213</td>
</tr>
<tr>
<td>D</td>
<td>Comprehension Test questions (ATM) case</td>
<td>217</td>
</tr>
<tr>
<td>E</td>
<td>Verification Test Models (ATM) case</td>
<td>219</td>
</tr>
<tr>
<td>F</td>
<td>Comprehension Test Models (HSS) case</td>
<td>225</td>
</tr>
<tr>
<td>G</td>
<td>Comprehension Test questions (HSS) case</td>
<td>230</td>
</tr>
<tr>
<td>H</td>
<td>Verification Test Models (HSS) case</td>
<td>233</td>
</tr>
<tr>
<td>I</td>
<td>Instructions</td>
<td>239</td>
</tr>
<tr>
<td>J</td>
<td>Experiments Interfaces</td>
<td>240</td>
</tr>
<tr>
<td>K</td>
<td>ANOVA and MANOVA Methods</td>
<td>247</td>
</tr>
<tr>
<td>L</td>
<td>Software Projects Failures (1992-2005)</td>
<td>252</td>
</tr>
<tr>
<td>M</td>
<td>Scoring of experiment1&amp;2</td>
<td>253</td>
</tr>
<tr>
<td>N</td>
<td>Sample data of Experiment1 including Demographic data</td>
<td>254</td>
</tr>
<tr>
<td>O</td>
<td>Experimental results</td>
<td>259</td>
</tr>
<tr>
<td>P</td>
<td>Publications</td>
<td>267</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Root Causes of Software Faults

Information technology (IT) is playing a crucial role in contemporary society. Information systems have become a vital component of successful businesses and organizations for the enormous advantages it has in easing the delivery of information. Moreover, information systems have been used by many organizations as strategic resources to attain or retain competitive advantages, as illustrated by the pervasive adoption of electronic commerce and enterprise systems (Siau & Xin, 2008). The increase production of software and information systems may indicates that system development projects are often successful, and that system development methods have matured enough to provide relatively low risk and high return business opportunities. However, according to industry studies, failure in software projects is common in many organizations. Studies pointed out that the errors which appear at the early stage of the software development can affect the reliability, cost, and safety of a system. Conclusions from these studies points to requirements specification as one major source of errors in software development and that the “highest density of major defects found through the
The report of the ‘Consultancy Standish Group International Inc.’ (2003 Chaos Report) a globally respected source of independent primary research and analysis of IT project performance, with an analysis of 13,522 IT projects revealed that 66% of all IT projects failed either because of over budgeting, over time, or failed to meet 20% or more of the business requirements for the system, 15% of the projects failed completely and were canceled prior to completion (Figure 1.1). The average cost overrun was 43% and 82% of the challenged projects delivered with time overrun. However, these numbers are a significant improvement over the previous survey conducted in 1995.

![Figure 1.1 Chaos Report (2003)](image)

Similar statistics were found in 2004 “CHAOS Report” (Johnson, 2006) when it was estimated that only 29% of software projects in large enterprises succeeded (i.e.
produced acceptable results that were delivered almost on-time and on-budget (Figure 1.2), 53% were “challenged” (significantly over budget and schedule), and 18% failed to deliver any substantial result. The projects that were in trouble have an average budget overrun of 56%. This represents a serious and chronic risk-control problem.

**Survey Results 2004**

Resolution of Projects

![Survey Results 2004](image)

**Figure 1.2** CHAOS Survey Results 2004

Other studies as the one conducted by “Consultant Capers Jones” in Marlborough, Mass., US stated:

"Large software systems...have one of the highest failure rates of any manufactured object in human history" (Ross, 2005).
Most IT experts agree that such failure occur more than they should. The failures are all over the world, they happen in every country, to large and small companies, in commercial, non profit and governmental organization, (Ross, 2005) (Appendix L). Charette (2005) indicates that the business and social costs of these failures run well into billions of dollars a year. Moreover he argued that small-medium size businesses spends about 4 to 5 percent of revenue on information technology while those that are highly IT dependent such as financial and telecommunication companies spending more than 10 percent on it. In organizations IT is now one of the largest corporate expenses after employee costs. Most of that money goes to fund new software projects in order to create better future for the organization and its customers.

1.2 Why Do Software Projects Fail?

Researchers over the past twenty years have studied the factors that could increase the high failure rate. A sample of these factors include the lack of early project management (Lyythinen et al., 1996), underestimation of the cost and effort associated with a project (Jorgensen, 2006), and the lack of structured development techniques (methodology) for developing a system design (Coad & Yordon, 1991). One of the most common factors repeatedly attributed to system failure, is the lack of accurate communication in the early system development process (Holtzblatt & Beyer, 1995; Damian et al., 2006). Chaos survey (1995; 2003) helps highlight the historic problem of bridging the communication gap between users who understand business, and system experts who understand technology. The Standish Group Report (“Chaos1995”) provides useful list of factors that practitioners view as important in contributing to system development failure (Table 1.1):

In this table, many of the top ten factors (1,2,3,7,8) are concerned with communication in the early phase of the system development process. This phase is commonly referred to as “System Analysis”. The survey results suggest that when considering factors for
project failure, practitioners recognized the importance of communication in early phases of planning.

**Table 1.1:** Factors Affecting System Development Failure “Chaos” 1995

<table>
<thead>
<tr>
<th>Rank</th>
<th>Project Failure Factor</th>
<th>% Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lack of User Input</td>
<td>12.8</td>
</tr>
<tr>
<td>2</td>
<td>Incomplete Requirements / Specifications</td>
<td>12.3</td>
</tr>
<tr>
<td>3</td>
<td>Changing Requirements and Specifications</td>
<td>11.8</td>
</tr>
<tr>
<td>4</td>
<td>Lack of Executive support</td>
<td>7.5</td>
</tr>
<tr>
<td>5</td>
<td>Technological incompetence</td>
<td>7.0</td>
</tr>
<tr>
<td>6</td>
<td>Lack of Resources</td>
<td>6.4</td>
</tr>
<tr>
<td>7</td>
<td>Unrealistic expectations</td>
<td>5.9</td>
</tr>
<tr>
<td>8</td>
<td>Unclear Objectives</td>
<td>5.3</td>
</tr>
<tr>
<td>9</td>
<td>Unrealistic Time Frames</td>
<td>4.3</td>
</tr>
<tr>
<td>10</td>
<td>New Technology</td>
<td>3.7</td>
</tr>
</tbody>
</table>

1.3 **Information Requirements Determination**

It has been recognized that determining correct and complete information requirements is essential for designing an information system. As mentioned in section 1.2, many information systems failures can be attributed to a lack of clear and specific information requirements. Information requirements determination, also known as requirements definition, requirements gathering, requirements elicitation, and requirements engineering, is concerned about figuring what to build (Holtzblatt & Beyer, 1995). It includes any activity undertaken by systems developers and analysts to specify the required functions in the proposed system (Browne & Ramesh, 2002). Four tasks to be performed in requirements determination have been identified (Siau & Xin, 2008).
- **Requirements specification**: to understand the organizational situation that the system under consideration aims to improve and describe the needs and constraints of the system under development.

- **Requirements negotiation**: to establish an agreement on the requirements of the system among the various stakeholders involved in the process.

- **Requirements representation**: to develop a mapping of real-world needs onto a requirements model.

- **Requirements validation**: to ensure that the derived specification corresponds to the original stakeholder's needs and conforms to the internal and/or external constraints set by the enterprise and its environment.

During the development of software project systems analysts gather information about users’ needs and the requirements expected in the proposed system and sometimes negotiate the requirements when they are unrealistic or confliction with others. This is done through direct discussions with system stakeholders and observations.

### 1.4 Conceptual Modeling

In the early stages of information systems development, analysts use conceptual modeling to build a representation of the domain understudy. High quality conceptual modeling work is important because it facilitates early detection and correction of system development errors (Wand & Weber, 2002).

The *American National Standards Institute* ANSI/SPARC definition of a conceptual data model is any model that is independent of the underlying hardware and software. Definitions of researchers in the area (Topi & Ramesh, 2002; Batra & Davis, 1992) did not also refer to any implementation details. Conceptual modeling is directly related to two tasks in requirements engineering- requirements representation and requirements validation. In requirements representation, conceptual models are created to map real-
world needs. During requirements validation, users of the system verify whether their needs have been correctly specified by viewing the conceptual models (Siau & Xin, 2008).

Many conceptual modeling techniques have been devised in different methodologies. As the Unified Modeling Language (UML) widely adopted in modeling, many system analysts employ Use Case diagrams to model information requirements. Use cases are used to capture the user functional requirements. Each use case describes an element of the functionality of a system, which gives users a result of value. The sum of these use cases defines the total functionality of the system. The system is then designed and implemented to support the use cases (Siau & Xin, 2008).

### 1.5 Human Factors in IS Development

Researchers in the field of information systems have realized the need to pay attention to the human dimensions in systems development. In particular, when identifying requirements people must communicate effectively and share a common understanding of the work problems and the required solutions (Holtzblatt & Beyer, 1995). For example, when a system analyst represents a domain in a conceptual model, the model should be understandable by other developers, or end users. In this social process, human factors have significant implications. Human factors may include human's cognitive abilities, personality, knowledge, and motivation of the participants in the requirements determination process (Siau & Xin, 2008). Researchers in the area of conceptual data modeling pointed out the need for more research in human factors in information systems development. Topi and Ramesh (2002, P.11) stated that:

“*We need a better understanding of the psychological processes in data modeling and the ways the tools affect these processes. This will enable us to find a firm theoretical basis for human factors research on data modeling. Researchers in this area should be*
interested not only in the characteristics of the current models, but the reasons underlying the potential performance differences between various approaches to data modeling.”

1.6 Motivation for the Research

During the analysis phase of information systems development, system analysts and developers capture and represent system requirements using conceptual models such as entity relationship diagram, Data flow diagram, Class diagram, Use case diagram. Considering that the requirements specification process must support effective communication between stakeholders who do not share common background (Holtzblatt & Beyer, 1995) and the fact that the failure of system development projects are attributed to faulty or incomplete requirements, it is then extremely important for the analyst and the success of the system to ensure that the conceptual models developed in the early phases of the system development must support communication between users and developers in defining and documenting system requirements as accurately as possible. The models should allow users to verify whether the analyst’s understanding of the system requirements reflects the reality as perceived by the users (Parsons & Cole, 2005).

From this perspective, research to evaluate techniques that represent requirements should focus on their capacity to facilitate this verification. To improve the performance and selection of analysis technique this work argues for the empirical evaluation of (Use Cases), one of the Unified Modeling Language (UML) models. UML is the standard for specifying, visualizing, constructing, and documenting the components of software systems (Booch et al., 1999). The UML has been adopted as a standard for the object oriented modeling and has already found widespread popularity in various domains. However, UML diagrams are not widely rated in terms of understandability by end users and developers alike (Agarwal & Sinha, 2003). Use case models serve as basis for
deriving other UML conceptual models. Thus it is important to ensure the quality of these models. This research will investigate whether use case models are useful in the analysis process of a software system by enabling users to verify that their needs are documented. There are two research questions in this study:

1- Does the format of use case model influence the understanding and the patterns of performance, when individuals have to solve tasks on the basis of their previously acquired knowledge? And which use case, text only or text accompanied with diagram better support user understanding of the domain requirements?

2- How does the degree of detail in a Use Case diagram that accompanies text in a use case model affect user comprehension of the domain requirements?

1.7 Research Objectives

Use Cases is the technique used in UML for handling the functional requirements in a software development project and serve as a means of communication between different stakeholders in a project. The models developed as a result of requirements analysis are represented in diagrammatic form, supplemented with textual description for those parts that cannot be captured diagrammatically. In order to reduce the possibility for differences in understanding stem from using different formats for the use case models it would be useful to evaluate the extent of the different stakeholders’ understanding to the model and also to detect differences in interpretation. This study has three objectives:

1- To present an empirical method for comparing different formats of use case model, and investigate the effect of the format on understanding the model by novice end users.
2- To extend the research by investigating high knowledge users to gain a comprehensive view of how stakeholders of different degrees of knowledge in the modeling technique may understand the use case models for the purpose of identifying system requirements.

3- To find out the impact of experience on understanding different formats of use case model.

To accomplish these objectives, a research instruments for the comparison of Use case formats was developed and two comparative studies were designed and implemented. The empirical instruments that used in lab experiments was developed from a combination of comprehension test, and verification test and finally a comparison between the performance of participants in the two studies was undertaken.

1.8 The Research Scope

In studying any research problem it is important to identify the research scope which is narrow enough to be effectively researched and yet touches an issue of significant potential impact. The researcher needs to focus on specific area for improvement. This research will focus on one of the techniques used within the unified modeling language (UML), the standard modeling language in the Object-Oriented methodology of information systems development. The UML has been widely accepted as the standard for object-oriented analysis and design (OOAD) (Kobryn, 1999). It has been an important part of the software development landscape since its introduction in 1997. UML models are used by professional developers to communicate their work to project stakeholders. For developers, eliciting high level goals early in the software development process is crucial. It focuses the developer on the problem domain and the needs of the stakeholders, rather than the implementation of the system. Despite the movement toward UML as a standard modeling language in practice, there appears very
few empirical researches on the effectiveness of various modeling techniques in UML (Dobing & Parsons, 2000).

“**USE CASE**” model is one of the key modeling techniques of UML which is utilized in the analysis phase for capturing and describing the functional requirements of a system. Use Case models can also be harnessed in communication between stakeholders in project development. It is therefore essential that use case models support the development process and promote understanding of the requirements among stakeholders. There is currently no detailed account of the cognitive processes involved in understanding software requirements. While there are several studies on how programmers understand programs (Burkhardt *et al.*, 2002) we have not been able to find any study on human understanding of use case models. This lack of research on Use Case models understandability means that the guidelines and practices on how use cases should be designed to provide a base for better understanding of the requirements, is highly subjective. We intend to investigate how understandability of Use Case models may depend on the format of the model and how different stakeholders understand use case models.

Two reasons provided for focusing on Use Case technique as an analysis tool, first, solving problems in the early stages of the development process that employ these models can reduce the cost and effort of fixing these problems later. Second, although considerable attention is devoted to object oriented development methods and the standard language (UML) in information systems field, the evaluation of existing methods is not keeping pace with the rapid growth of the systems development methods (Siau & Tan, 2005). The implication of that is:

1. By failing to evaluate currently used object oriented methods, organizations may not clearly comprehend the usefulness and effectiveness of these methods.

2. The lack of object oriented methods evaluation including (UML), impede practitioners and researchers trying to understand the strength and weaknesses of various methods. This understanding is a critical knowledge for improving existing methods or designing new ones.
1.9  Research Implications

1.91  Theoretical Perspective

- In view of the paucity of empirical research on the effectiveness of various techniques of UML, it is significant from an academic perspective to independently evaluate the capabilities and limitations of UML techniques. Such evaluation can contribute to the development of theoretical underpinnings of UML, and to an improvement in its modeling power and usability.

- This work takes a step towards empirical validation of the theoretical basis regarding the understandability of (UML) models. It presents an empirical methodology with control, which can hopefully be used to study the effects of other factors on understanding the models, and other dependent variables as well as studying the rest of the modeling diagrams in UML.

- As UML become complex with each version, it is useful to focus on the core diagrams of UML particularly in teaching object oriented and software development methods. This work is relevant to instructors in universities and software practitioners as well.

1.92  Practical Perspective

From a practical standpoint, this work aids to the appropriate application of UML in systems development projects. It has implications for both method developers who want to know the strength and weaknesses of various methods, as
well as for practitioners who want to use comparisons as a practical tool for selecting methods. From an analyst’s perspective, this work assists in strategies at the beginning of the analysis phase, when decisions are made regarding the format and level of abstraction to be actually used to create requirements models that meet the user’s needs. It would be useful if modelers could create models that convey accurate information with an easy-to-understand manner.

1.10 Organization of the Thesis

The remaining chapters are organized in the following manner. Chapter 2 outlines literature review of previous research in system analysis and design methods. Chapter 3 outlines the main topics related to use cases. Chapter 4 outlines the theoretical foundation for the proposed empirical method which led to develop the hypotheses that will form the basis for the two empirical studies in this research. Chapter 5 outlines the proposed design for both empirical studies and highlights the experimental procedures used in the study. Chapters 6 and 7 present the analysis of the empirical work, Chapter 8 presents general discussion and finally conclusions are made in chapter 9.
References


http://www.cs.colorado.edu/~kena/classes/6448/s01/homeworks/hw02.html .


