TRACEABILITY MANAGEMENT IN A DEVOPS ENVIRONMENT WITH CONTINUOUS INTEGRATION

Iresha Dilhani Rubasinghe

(178020N)

Degree of Master of Philosophy

Department of Computer Science and Engineering

University Of Moratuwa
Sri Lanka

April 2019

TRACEABILITY MANAGEMENT IN A DEVOPS ENVIRONMENT WITH CONTINUOUS INTEGRATION

Iresha Dilhani Rubasinghe

(178020N)

Degree of Master of Philosophy

Department of Computer Science and Engineering

University Of Moratuwa
Sri Lanka

April 2019

Declaration

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: ………………………… Date: ……………......

I. D. Rubasinghe

The above candidate has carried out research for the MPhil thesis under my supervision.

Name of the supervisor: Dr. D. A. Meedeniya

Signature of the supervisor: ……………………….. Date: ……………......

Name of the supervisor: Dr. G. I. U. S. Perera

Signature of the supervisor: ……………………….. Date: ……………......

Abstract

Software artefacts traceability is an important factor during the process of software development to analyse changes occur in software components. Traceability improves the quality attributes of software systems such that strengthens the testability, maintainability, reusability and helps for the system acceptance by providing consistent system documentation to the users. Meanwhile, the concept DevOps motivates towards the reduction of the gap between development and operations requiring considerable organizational changes. In a DevOps environment, significant software artefact changes are expectable rapidly where continuous integration is essential. Continuous integration is a cornerstone practice in DevOps that frequently merges developer working copies into a single shared branch. There is a requirement of determining and analysing the resulted impact of the traceability in order to make accurate change acceptance decisions during software development. Therefore, the core research problem addressed is determining a methodology for change detection and impact analysis together with software artefact synchronization to preserve consistency across all artefacts in a DevOps environment. A rule-based methodology is followed with visualization and analysis techniques applied on a proof-of-work traceability management prototype tool: SAT-Analyser 2.0. The evaluation results and industry-level user study results have shown the significant usefulness and suitability of the approach to a DevOps environment as well as to any software development process model.

Keywords:

Consistency management, Continuous integration, DevOps, Change impact analysis, Traceability management

Acknowledgements

I would like to offer my heartfelt gratitude to supervisors, Dr. D. A. Meedeniya and Dr. G. U. I. S. Perera for the immense guidance provided throughout the research project. Their valuable feedback and extremely kind advices motivated me to complete the research work as well as moulded my academic competencies. I ever appreciate their knowledge sharing and flexibility that made the research work an interesting and pleasant experience that would not have been possible without them.

I would also like to thank my project advisory panel members; Prof. N. D. Kodikara at University of Colombo School of Computing, Dr. A. C. De Silva at Department of Electronic and Telecommunication Engineering, University of Moratuwa, Dr. M. J. Walpola and Dr. K. Gunasekara at Department of Computer Science and Engineering, University of Moratuwa for the thoughtful comments and suggestions that helped me to achieve my research goals.

I highly appreciate the financial support received from Senate Research Council Grant, University of Moratuwa to complete this work.

I am thankful to my family and colleagues at the Department of Computer Science and Engineering, University of Moratuwa for the unfailing support.

Table of Contents

Declaration i

Acknowledgements iii

Table of Contents iv

List of Figures vii

List of Tables xi

List of Appendices xii

List of Abbreviations xiii

Section 1 1

1 Introduction 1

1.1 Overview of the research 1
1.2 Motivation for the research 2
1.3 Problem statement 3
1.4 Research statement 4
1.5 Research objectives 4
1.6 Research outcome 5

Section 2 6

2 Literature review 6

2.1 Overview 6
2.1.1 Software artefacts in SDLC 6
2.1.2 Traceability 6
2.1.3 Software artefact traceability in a DevOps environment 9
2.2 Data pre-processing 10
2.3 Information retrieval methods 11
2.4 Traceability management 11
2.4.1 Evaluation of traceability support techniques 11
2.5 Change detection 13
2.5.1 Change detection techniques 14
2.6 Change impact analysis 15
2.6.1 Change impact analysis of heterogeneous software artefacts 16
2.6.2 Change impact analysis categories 17
2.6.3 Change impact estimation and analysis techniques 19
2.6.4 Change impact analysis related frameworks and models 20
2.7 Consistency checking and management 25
2.8 Change propagation in DevOps 26
2.8.1 Change propagation techniques 26
2.9 Continuous integration 27
2.9.1 DevOps practices 28
2.9.2 DevOps tools 30
2.9.3 DevOps related project management tools 32

2.10 Analysis of related work 34
2.11 Visualization of traceability links 36
2.12 Tool support for tractability management and continuous integration 38
2.13 Evaluation techniques of traceability management 41
2.13.1 Quality measures 41
2.13.2 Network analysis 43
2.13.3 Traceability testing techniques 44
2.13.4 Supported testing tools 44
2.14 Discussion 45
2.14.1 Limitations in current practices 46
2.14.2 Future challenges and research directions 46
Section 3........ 47
3 Research methodology 47
3.1 System design 47
3.1.1 System overview 47
3.1.2 Research model 48
3.1.3 System architecture 52
3.1.4 Abstract system workflow 54
3.1.5 Detailed system workflow 55
3.1.6 System class structure 56
3.2 Traceability establishment 57
3.2.1 Data pre-processing of SAT-Analyser tool 57
3.2.2 Input to XML conversion 62
3.2.3 Traceability generation 63
3.3 Traceability visualization 66
3.3.1 Default SAT-Analyser informative traceability visualization 68
3.3.2 Analytical traceability visualization 71
3.3.3 Interactive traceability visualization 72
3.4 Impact analysis and change propagation 73
3.4.1 Identification of strengths of artefacts and relationships 73
3.4.2 Impact analysis process: workflow 78
3.4.3 Impact analysis process: pseudo code and implementation details 78
3.4.4 Impact analysis process: user modifiability 81
3.4.5 Change propagation of the impact 82
3.4.6 Notification approach 89
3.5 Traceability management 90
3.5.1 Change detection 91
3.5.2 Consistency management 98
3.5.3 Continuous integration 101
3.5.4 Multi-user supportability 109
3.6 Extended SAT-Analyser evaluation 110
3.6.1 Implementation of the accuracy analysis module 110
3.6.2 Implementation of the network analysis module 112
3.7 Tool performance analysis 113
3.8 Conclusion 115
Section 4........ 116

4 Evaluation.........116
4.1 Datasets and materials.........116
4.1.1 Pre-defined categorization of change types.........118
4.1.2 Evaluation environment specification.........119
4.2 Experimental results: case study 1 (POS system).........119
4.2.1 Evaluation of traceability establishment component.........121
4.2.2 Evaluation of continuous integration process.........123
4.2.3 Performance analysis.........125
4.3 Experimental results: case study 2 (Tour management system).........127
4.3.1 Evaluation of traceability establishment component.........129
4.3.2 Evaluation of continuous integration process.........130
4.3.3 Performance analysis.........132
4.4 Experimental results: case study 3 (MyDrive multimedia library).........133
4.4.1 Evaluation of traceability establishment component.........135
4.4.2 Evaluation of continuous integration process.........136
4.4.3 Performance analysis.........137
4.5 Experimental results: case study 4 (Disease management system).........139
4.5.1 Evaluation of traceability establishment component.........141
4.5.2 Evaluation of continuous integration process.........142
4.5.3 Performance analysis.........144
4.6 Experimental results: case study 5 (E-School management system).........145
4.6.1 Evaluation of traceability establishment component.........147
4.6.2 Evaluation of continuous integration process.........148
4.6.3 Performance analysis.........149
4.7 SAT-Analyser performance analysis.........151
4.7.1 Traceability establishment performance.........151
4.7.2 Accuracy evaluation of change detection component.........152
4.7.3 Accuracy evaluation of impact analysis component.........152
4.8 Usability of the extended SAT-Analyser tool.........153
Section 5.........156
5 Discussion.........156
5.1 Feature analysis of SAT-Analyser in practice.........156
5.2 Analysis of the usability study based evaluation.........160
5.3 Mapping of the objectives and the methodology.........161
5.4 Limitations.........161
5.5 Future work.........162
5.6 Conclusion.........163
References.........165

List of Figures

Figure 2-1 : Summary of traceability classification....9
Figure 2-2 : The software evolution process (Sommerville, 2010)....13
Figure 2-3 : Change impact analysis process (Li, Sun, Leung, & Zhang, 2013)....15
Figure 2-4 : Change impact analysis categorization....19
Figure 2-5 : Chianti tool architecture (Ren et al., 2005)....21
Figure 2-6 : Continuous integration process (Farcic, 2016)....27
Figure 2-7 : DevOps overview ("QASource DevOps Experts," 2018)....29
Figure 2-8 : Jenkins workflow....30
Figure 2-9 : Docker workflow....31
Figure 3-1: Extended SAT-Analyser system overview....47
Figure 3-2 : SAT-Analyser tool research model....49
Figure 3-3 : Extended-SAT-Analyser system architecture....52
Figure 3-4 : Extended SAT-Analyser abstract workflow....54
Figure 3-5 : Extended SAT-Analyser detailed workflow....55
Figure 3-6 : Extended SAT-Analyser class structure....56
Figure 3-7 : Semantic network for words....64
Figure 3-8 : Pre-defined relationship XML model....65
Figure 3-9 : Traceability link generation component....66
Figure 3-10 : Visualization component....68
Figure 3-11 : Default SAT-Analyser Traceability visualization menu....69
Figure 3-12 : Default SAT-Analyser visualization full graph view....70
Figure 3-13 : Default SAT-Analyser visualization requirement artefact filtered view....70
Figure 3-14 : General analytical traceability graph....72
Figure 3-15 : Analytical traceability graph in traceability validation....72
Figure 3-16 : Interactive traceability graph view....73
Figure 3-17 : Node-edge direct connectivity....75
Figure 3-18 : Node-edge scenario 1....76
Figure 3-19 : Node-edge scenario 2....77
Figure 3-20 : Impact analysis component workflow....78
Figure 3-21 : Code snippet of weight and influential factor calculators....81
Figure 3-22 : SAT-Analyser generated CIA results....82
Figure 3-23 : User altered CIA results....82

Figure 3-24 : Change propagation workflow 83
Figure 3-25 : Code snippet of change propagation implementation 86
Figure 3-26 : Impact analysis results window 87
Figure 3-27 : Change propagated analytical traceability graph view 88
Figure 3-28 : Change propagated interactive traceability graph view 88
Figure 3-29 : Trello change propagation card instance 90
Figure 3-30 : Trello board with change propagation notification 90
Figure 3-31 : Design diagram: change detection component 91
Figure 3-32 : Diffmk based change types declaration 96
Figure 3-33 : Change detection menu item 96
Figure 3-34 : Insufficient CI versions for change detection 97
Figure 3-35 : Change detection results xml_CD directory 97
Figure 3-36 : Change detection results in outcome window 97
Figure 3-37 : Consistency management workflow 99
Figure 3-38 : Scheduler workflow 102
Figure 3-39 : Scheduler code snippet 104
Figure 3-40 : Continuous integration configuration menu item 105
Figure 3-41 : Continuous integration configuration window 105
Figure 3-42 : Continuous integration menu item 106
Figure 3-43 : Continuous artefact integration window 106
Figure 3-44 : Continuous artefact integration window with disabled forward option 106
Figure 3-45 : Code snippet of version control management 108
Figure 3-46 : CI version 1 of a project 108
Figure 3-47 : CI version 2 of a project 108
Figure 3-48 : Multi-user accessible SAT-Analyser web version 109
Figure 3-49 : SAT-Analyser traceability establishment accuracy analysis window 110
Figure 3-50 : SAT-Analyser CIA accuracy analysis window 111
Figure 3-51 : Network analysis centrality measures code snippet 112
Figure 3-52 : SAT-Analyser network analysis main window 113
Figure 3-53 : Network analysis artefact information view 113
Figure 3-54 : SAT-Analyser traceability establishment performance analysis window 114
Figure 3-55 : SAT-Analyser CIA performance analysis outcome window 114
Figure 4-1 : Project scale 118

Figure 4-2 : POS system description 120
Figure 4-3 : POS system design diagram 120
Figure 4-4 : SAT-Analyser main artefact summary for POS system 120
Figure 4-5 : POS system Relations.xml instance 121
Figure 4-6 : Part of the traceability visualization graph - POS system 122
Figure 4-7 : Network analysis summary - POS system 122
Figure 4-8 : POS system change detection window 123
Figure 4-9 : POS system impact analysis window 124
Figure 4-10 : POS system change propagation instance 124
Figure 4-11 : CIA statistical analysis results: POS system 125
Figure 4-12 : CIA performance analysis results: POS system 126
Figure 4-13 : Tour management system description 127
Figure 4-14 : Tour management system design diagram 127
Figure 4-15 : SAT-Analyser main artefact summary for tour management system 128
Figure 4-16 : Tour management system Relations.xml instance 129
Figure 4-17 : Traceability visualization - tour management system 129
Figure 4-18 : Network analysis summary - tour management system 130
Figure 4-19 : Tour management system change detection window 131
Figure 4-20 : Tour management system impact analysis window 131
Figure 4-21 : Tour management system change propagation instance 131
Figure 4-22 : CIA statistical analysis results: tour management system 132
Figure 4-23 : CIA performance analysis results: tour management system 133
Figure 4-24 : MyDrive multimedia library system description 133
Figure 4-25 : MyDrive multimedia library system design diagram 134
Figure 4-26 : SAT-Analyser artefact summary for MyDrive multimedia library system 134
Figure 4-27 : Multimedia library system Relations.xml instance 135
Figure 4-28 : Traceability visualization - multimedia library system 135
Figure 4-29 : Network analysis summary - multimedia library system 135
Figure 4-30 : Multimedia library system change detection window 136
Figure 4-31 : Multimedia library system impact analysis window 137
Figure 4-32 : Multimedia library system change propagation instance 137
Figure 4-33 : CIA statistical analysis results: multimedia library system 138
Figure 4-34 : CIA performance analysis results: multimedia library system 138

Figure 4-35 : Disease management system description 139
Figure 4-36 : Disease management system design diagram 140
Figure 4-37 : SAT-Analyser main artefact summary for disease management system......... 140
Figure 4-38 : Disease management system Relations.xml instance 141
Figure 4-39 : Traceability visualization - disease management system 141
Figure 4-40 : Network analysis summary - disease management system 142
Figure 4-41 : Disease management system change detection window................................. 143
Figure 4-42 : Disease management system impact analysis window 143
Figure 4-43 : Disease management system change propagation instance 143
Figure 4-44 : CIA statistical analysis results - disease management system......................... 144
Figure 4-45 : CIA performance analysis results: disease management system 145
Figure 4-46 : E-School management system description 145
Figure 4-47 : E-School management system design diagram... 146
Figure 4-48 : SAT-Analyser main artefact summary for E-School management system 146
Figure 4-49 : Relations XML format of traceability establishment - E-School system 147
Figure 4-50 : Traceability visualization – E-School management system 147
Figure 4-51 : Network analysis summary – E-School management system 148
Figure 4-52 : E-School management system change detection window 148
Figure 4-53 : E-School management system impact analysis window................................. 149
Figure 4-54 : E-School management system change propagation instance........................... 149
Figure 4-55 : CIA statistical analysis results: E-School management system....................... 150
Figure 4-56 : CIA performance analysis results: E-School management system.................. 150
Figure 4-57 : SAT-Analyser traceability establishment performance 151
Figure 4-58 : SUS positive responses analysis 154
Figure 4-59 : SUS negative responses analysis 155
Figure 4-60 : SAT-Analyser usability tag cloud.. 155
Figure 5-1 : Research objectives-methodology-results mapping.. 161

List of Tables

Table 2.1 : Evaluation of software artefacts traceability management techniques.................12

Table 2.2 : Summary of change impact analysis techniques 20

Table 2.3 : Change impact analysis related work summary 23

Table 2.4 : Scope-based change impact analysis related work...25

Table 2.5 : Evaluation of related work on traceability management......................................35

Table 2.6 : Traceability visualization techniques 36

Table 2.7 : Evaluation of related work on traceability visualization techniques.....................37

Table 2.8 : Tool support for traceability management..38

Table 2.9 : Tool support for information retrieval..39

Table 2.10 : Tool support for traceability management..40

Table 3.1 : Analysis of existed SAT-Analyser 50

Table 3.2 : Analysis of the SAT-Analyser with DevOps extension 51

Table 4.1 : Dataset summary 116

Table 4.2 : Artefact categorization: POS system 121

Table 4.3 : Artefact categorization: tour management system 128

Table 4.4 : Artefact categorization: MyDrive multimedia library system............................134

Table 4.5 : Artefact categorization: disease management system 140

Table 4.6 : Artefact categorization: E-School management system 146

Table 4.7 : Change detection component accuracy evaluation...152

Table 4.8 : Change impact analysis component accuracy evaluation...................................152

Table 5.1 : Comparison between Jenkins and SAT-Analyser for CI....................................158

Table 5.2 : Industry level traceability management vs. SAT-Analyser tool..........................158

Table 5.3 : Existing traceability management tools vs. SAT-Analyser................................159

List of Appendices

Appendix A: Initial survey 173

Appendix B: User acceptance survey 178

Appendix C: Research tool configuration settings 180

Appendix D: SAT-Analyser 2.0 user guide overview 181

Appendix E: List of companies involved in the surveys/ interviews 186

Appendix F: Published papers 187

List of Abbreviations

AST	Abstract Syntax Tree
AIS	Actual Impact Set
ANTLR	ANother Tool for Language Recognition
AO	Aspect-Oriented
AR	Augmented Reality
AWS	Amazon Web Services
CIA	Change Impact Analysis
CD	Continuous Delivery
CI	Continuous Integration
CIP	Change Impact Prediction
CR	Change Request
DAG	Directed Acyclic Graph
DSL	Domain Specific Language
EDG	Entity Dependency Graph
EIS	Estimated Impact Set
GCT	Goal-Centric Traceability
GPS	Global Positioning System
HMC	Hidden Markov Chain
IR	Information Retrieval
IDE	Integrated Development Environment
IoT	Internet Of Things
IT	Information Technology
IQR	Interquartile Range
LSI	Latent Semantic Indexing
LTR	Likelihood To Recommend
MDD	Model Driven Development
MDE	Model Driven Engineering
ML	Machine Learning
NER	Named Entity Recognizer
NLP	Natural Language Processing
NPS	Net Promoter Score
PCA	Principle Component Analysis

PM	Project Management
POS	Part-Of-Speech, Point Of Sales
RCM	Requirement Change Management
RSSI	Really Simple Syndication
ROI	Return On Investment
SCM	Supply-Chain Management
SMS	Short Message Service
SVD	Singular Value Decomposition
SIG	Soft Goal Interdependency Graph
SDLC	Software Development Life Cycle
SRS	Software Requirement Specification
SUS	System Usability Scale
TF-IDF	Term Frequency–Inverse Document Frequency
TDD	Test Driven Development
UAT	User Acceptance Testing
UML	Unified Modeling Language
VM	Virtual Machine
VSM	Vector Space Model