

VU Research Portal

Supporting Architecture Evolution by Mining Software Repositories

Vanya, A.

2012

document version

Publisher's PDF, also known as Version of record

[Link to publication in VU Research Portal](#)

citation for published version (APA)

Vanya, A. (2012). *Supporting Architecture Evolution by Mining Software Repositories*. [PhD-Thesis - Research and graduation internal, Vrije Universiteit Amsterdam].

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:

vuresearchportal.ub@vu.nl

Contents

Acknowledgments	ix
1 Introduction	1
1.1 Software Evolution	1
1.2 Mining Software Repositories	3
1.3 The Darwin Project	5
1.4 The Software System Analyzed	6
1.5 Problem Statement	8
1.6 Research Questions	10
1.7 Research Methods and Methodology	11
1.8 Outline of the Thesis	13
1.9 List of Publications	13
2 Approximating Change Sets at Philips Healthcare	15
2.1 Introduction	15
2.2 Related Work	17
2.2.1 Change Set Definitions	17
2.2.2 Change Set Approximation	18
2.2.3 Using Transactions	18
2.3 Case-Study Environment	19
2.4 Approximating Change Sets	21
2.5 Evaluation of Change Sets	22
2.5.1 Cross-checking with developers	22
2.5.2 Postlists	26
2.6 Discussion	29
2.7 Threats to Validity	29
2.8 Conclusion and consequences	30
3 Assessing Software System Decompositions with Evolutionary Clusters	31
3.1 Introduction	31
3.2 Related Work	32
3.2.1 History-Related Research	32
3.2.2 Recovering Software Components	33
3.3 Approach Overview	34
3.4 From Change Sets to evolutionary clusters	35
3.4.1 Change Sets and Their Approximations	35
3.4.2 The Similarity Metric	37
3.4.3 Hierarchy of evolutionary clusters	37
3.4.4 Evolutionary clusters	39

CONTENTS

3.4.5	The Filtered Hierarchy	39
3.4.6	Identifying Hot Spots	39
3.4.7	Evaluation of Hot Spots	40
3.5	Case Study	40
3.5.1	Our Approach at Work	40
3.5.2	The Analyzed Hot Spots	43
3.6	Discussion and Future Directions	45
3.7	Conclusion	47
4	Multidimensional Characterization of Evolutionary Clusters	49
4.1	Introduction	49
4.2	Study Environment	52
4.3	Need For Characterization	52
4.4	Properties of Evolutionary Clusters	53
4.4.1	Property 1: Cluster Size	55
4.4.2	Property 2: Borders Crossed	55
4.4.3	Property 3: Support Distribution	55
4.4.4	Property 4: Jaccard Similarity Distribution	56
4.4.5	Property 5: First Co-changes	56
4.4.6	Property 6: Last Co-changes	57
4.4.7	Property 7: Co-change Tendencies	57
4.4.8	Property 8: Static Relationships	58
4.4.9	Discussion	59
4.5	Evolution Anti-Scenarios	59
4.6	Property Justification	60
4.6.1	Property 1: Cluster Size	60
4.6.2	Property 2: Borders Crossed	61
4.6.3	Property 3: Support Distribution	61
4.6.4	Property 4: Jaccard similarity Distribution	62
4.6.5	Property 5: First Co-changes	62
4.6.6	Property 6: Last Co-changes	62
4.6.7	Property 7: Co-change Tendencies	63
4.6.8	Property 8: Static Relationships	63
4.7	Querying Evolutionary Clusters	63
4.7.1	Case 1	64
4.7.2	Case 2	66
4.7.3	Case 3	67
4.7.4	Comparing Selection Approaches	70
4.7.5	Discussion	73
4.8	Generalization	73
4.9	Threats to Validity	75
4.10	Related Work	77

4.11 Conclusion and Future Work	79
5 Resolving Unwanted Couplings Through Interactive Exploration of Co-evolving Software Entities	81
5.1 Introduction	81
5.2 Related Work	83
5.2.1 Visualizing co-changed entities	83
5.2.2 Interactions with visualizations	84
5.2.3 Reasons for interaction	85
5.2.4 Identifying reasons for co-changes	86
5.3 Experimental Design	87
5.3.1 Unwanted couplings to be analyzed	87
5.3.2 Interacting with co-changed entities	89
5.3.3 Analysis of unwanted couplings	93
5.4 Case Study	96
5.4.1 Case 1: Recalling reasons for co-changes	96
5.4.2 Case 2: Seeking solutions to unwanted couplings	98
5.4.3 Case 3: Co-change of C implementation and header files	99
5.5 Discussion	100
5.6 Threats to Validity	106
5.7 Conclusions	108
6 Conclusion	109
6.1 Introduction	109
6.2 Process Outline	110
6.3 Categories of lessons learned	113
6.4 Lessons learned	116
6.4.1 Knowing the process	116
6.4.2 Knowing the data	117
6.4.3 Frequent feedback loops	118
6.5 Related Work	121
6.5.1 Modularity for Software Evolvability	121
6.5.2 Critical Success Factors	121
6.6 Conclusion	122
6.7 Future Work	124
Samenvatting	127
SIKS Dissertation Series	131
Bibliography	136

List of Figures

1.1	Magnetic Resonance Imaging	6
1.2	Two decompositions of the same software system	9
2.1	An example process instance executed by a developer	19
2.2	Change sets approximated with different time interval. A cross denotes a file checked-in to the version repository by a developer D	21
2.3	Screenshot of the on-line survey to evaluate change sets	23
2.4	Number of check-ins made by most active developers at Philips Healthcare. We invited Developers 5–14 to evaluate the approximated change sets.	25
2.5	A sample postlist presenting a change set.	26
3.1	The evolutionary cluster approach	35
3.2	Change Sets	36
3.3	Change Set approximations	36
3.4	The complete dendrogram	42
3.5	Cohesion levels	43
3.6	Modification Ratios	44
3.7	Example Hot Spot	45
3.8	Approach generalization	47
4.1	Illustration of the evolutionary cluster characterized	54
4.2	Three scenarios for co-change tendency between two building blocks	58
4.3	An unwanted coupling visualized	69
5.1	A visualization of co-changed entities	83
5.2	iVIS main screen	88
5.3	Abstraction level increased	90
5.4	Support threshold	91
5.5	Internal relations	92
5.6	Re-arranging entities	94
5.7	Preparation of the visualization for the working session.	95
5.8	An example showing how adjusting threshold value revealed relevant co-changes.	97
5.9	An illustration of the unwanted coupling in Case 2.	98
5.10	Co-changes in Case 3	100
5.11	Unwanted couplings in ArgoUML	107
6.1	The artifacts generated and used by the process (C_i $i \in [1..N]$ denote components)	114
6.2	Impact relationships between categories	116

List of Tables

2.1	The Approximated Change Sets (ACS)	22
2.2	Precision estimated with help of developers	24
2.3	Precision estimated using postlists	28
2.4	Recall estimated using postlists	28
3.1	Jaccard similarities	38
3.2	Average linkage values (alv) for $\{C_1, \{C_2, C_3\}, C_4\}$	38
3.3	Top 10 Hot Spots	42
4.1	Support Distribution	56
4.2	Jaccard Similarity Distribution	56
4.3	First Co-change Distribution	57
4.4	Last Co-change Distribution	57
4.5	Co-change Tendency Distribution	58
4.6	Selection Criteria for Case 1	66
4.7	Selection Criteria for Case 2	67
4.8	Priority mapping for Case 1	72
4.9	Priority mapping for Case 2	72
4.10	Priority mapping for the ArgoUML case	75
5.1	Interactions supported	86
5.2	Overview of the working sessions	101
5.3	Sequences of interactions during the sessions discussed	102
6.1	Lessons Learned and Their Mapping to Process Steps	116

