

Table of Contents

I	Prelude	1
1	Introduction	3
1.1	Problem Statement	4
1.1.1	Clone Rates	4
1.1.2	Clone Evolution	5
1.1.3	Relevant Clones	5
1.2	Terminology	6
1.2.1	Token	6
1.2.2	Code Fragment	7
1.2.3	Clone Pair	7
1.2.4	Clone Class	8
1.2.5	Clone Type	8
1.3	Contributions	11
1.3.1	Clone Rates	11
1.3.2	Clone Evolution	12
1.3.3	Clone Removals	12
1.3.4	Usability of Clone Data	12
1.4	Project Context	13
1.5	Related Publications	13
1.6	Thesis Outline	14
2	Related Research	17
2.1	Causes	17
2.1.1	Development Strategy	18
2.1.2	Maintenance Benefits	19
2.1.3	Overcoming Underlying Limitations	20

2.1.4	Cloning By Accident	22
2.2	Effects	22
2.2.1	Co-Change and Inconsistent Change	23
2.2.2	Code Size and Code Comprehension	26
2.3	Clone Rate and Locality	27
2.4	Detection	31
2.4.1	Text-based	31
2.4.2	Token-based	33
2.4.3	AST-based	35
2.4.4	PDG-based	36
2.4.5	Metric-based	36
2.4.6	Other techniques	37
2.4.7	Beyond Source Code	38
2.4.8	Applications	39
2.5	Clone Mapping	40
2.6	Change Patterns	44
2.6.1	Clone Fragment Patterns	44
2.6.2	Clone Class Patterns	46
2.7	Clone Evolution Analysis	50
2.7.1	Clone Rate	51
2.7.2	Changing Clones	52
2.7.3	Clones and Defects	53
2.7.4	Impact on Maintainability	54
2.8	Management	57
2.8.1	Monitoring	57
2.8.2	Visualization	58
2.8.3	Removal	60
2.9	Human-based Studies	64
2.10	Code Search	67
2.11	Summary	69

II Studies 71

3 Software Clone Rates 73

3.1	Study Setup	74
3.1.1	Source Code Corpora	74

3.1.2	Clone Detection	76
3.1.3	Validity Procedure	80
3.1.4	Quantitative Analysis	84
3.1.5	Qualitative Analysis	95
3.1.6	Threats to Validity	99
3.2	Summary	100
4	Evolution of Software Clones	103
4.1	Repository Mining	104
4.2	Clone Detection	104
4.3	Fragment Mapping	105
4.3.1	Creating Buckets	106
4.3.2	Updating Fragments	106
4.3.3	Mapping Fragments	106
4.3.4	Detecting Late Propagation	108
4.4	Study Setup	110
4.5	Findings	111
4.5.1	Clone Rate	111
4.5.2	Fragment's Lifetime	114
4.5.3	Clone Type Changes	116
4.5.4	Change Consistency	116
4.5.5	Late Propagation	117
4.6	Threats to Validity	120
4.7	Summary	120
5	Clone Removals and their Return on Investment	123
5.1	Identification of Clone Refactorings	124
5.1.1	Clone Detection	125
5.1.2	Filtering Clone Fragments	126
5.1.3	Manual Inspection	127
5.1.4	Categorization and grouping	128
5.2	Tracking Refactored Code Fragments	132
5.3	Study Setup	133
5.4	Refactoring Frequency	137
5.5	Refactoring Types	138
5.6	Ranking Clones	142
5.6.1	Clone Types	142

5.6.2	Clone Metrics	146
5.7	Return on Investment	151
5.7.1	Refactoring Types	151
5.7.2	Committer Know-how	152
5.7.3	Quantitative Analysis	154
5.7.4	Qualitative Analysis	156
5.8	Threats to Validity	158
5.9	Summary	159
6	Developers Fixing Cloned Bugs	161
6.1	Experimental Design	162
6.1.1	Hypotheses and Variables	162
6.1.2	Design	163
6.1.3	Subjects	164
6.1.4	Objects	165
6.1.5	Instrumentation	169
6.2	Experiment Execution	171
6.3	Evaluation	173
6.3.1	Descriptive Statistics	173
6.3.2	Hypothesis Testing	176
6.4	Discussion	177
6.5	Threats to Validity	179
6.5.1	Construct Validity	179
6.5.2	Internal Validity	179
6.5.3	External Validity	179
6.6	Summary	180
7	Approximative Code Search	183
7.1	Search Algorithm	185
7.1.1	Partitioning	185
7.1.2	Searching	185
7.1.3	Checking	187
7.2	Implementation	188
7.2.1	Preprocessing	189
7.2.2	First Version Analysis	189
7.2.3	Postprocessing	191
7.3	Complexity	191

7.4	Study Setup	192
7.5	Performance	193
7.6	Incomplete Defect Correction	197
7.7	Discussion	200
7.8	Threats to Validity	201
	7.8.1 Internal Validity	201
	7.8.2 External Validity	202
7.9	Summary	202
 III Finale		 203
 8 Conclusion		 205
8.1	Clone Rates	205
8.2	Clone Evolution	205
8.3	Relevant Clones	206
8.4	Future Work	207
	8.4.1 Closed-source Projects	207
	8.4.2 Human-based Studies	208
	8.4.3 IDE Integration	208
 Appendix		 208
 A Experiment		 209
A.1	Handout	209
A.2	Pre-study Questionnaire	209
A.3	Post-study Questionnaire	209
A.4	Introduction	214
	A.4.1 Frozen Bubble	214
	A.4.2 Pacman	214
A.5	Bug Report	214
	A.5.1 Frozen Bubble	214
	A.5.2 Pacman	215