Giuliano Antoniol

List of Publications by Year
in descending order

24	1,215	1040056 9	1125743 13
papers	citations	h-index	g-index

24	24	24	725
all docs	docs citations	times ranked	citing authors

#	Article	IF	Citations
1	Feature Location Using Probabilistic Ranking of Methods Based on Execution Scenarios and Information Retrieval. IEEE Transactions on Software Engineering, 2007, 33, 420-432.	5.6	371
2	An exploratory study of the impact of antipatterns on class change- and fault-proneness. Empirical Software Engineering, 2012, 17, 243-275.	3.9	277
3	CERBERUS: Tracing Requirements to Source Code Using Information Retrieval, Dynamic Analysis, and Program Analysis. , 2008, , .		108
4	Linguistic antipatterns: what they are and how developers perceive them. Empirical Software Engineering, 2016, 21, 104-158.	3.9	85
5	EARMO: An Energy-Aware Refactoring Approach for Mobile Apps. IEEE Transactions on Software Engineering, 2018, 44, 1176-1206.	5.6	50
6	On the use of developersâ€™ context for automatic refactoring of software anti-patterns. Journal of Systems and Software, 2017, 128, 236-251.	4.5	34
7	Can Lexicon Bad Smells Improve Fault Prediction?. , 2012, , .		30
8	A Heuristic-Based Approach to Identify Concepts in Execution Traces. , 2010, , .		28
9	An empirical study on requirements traceability using eye-tracking. , 2012, , .		26
10	An empirical study on the importance of source code entities for requirements traceability. Empirical Software Engineering, 2015, 20, 442-478.	3.9	24
11	Requirements Traceability for Object Oriented Systems by Partitioning Source Code. , 2011, , .		23
12	Concept Location with Genetic Algorithms: A Comparison of Four Distributed Architectures. , 2010, , .		22
13	Trust-Based Requirements Traceability. , 2011, , .		22
14	A large-scale empirical study of code smells in JavaScript projects. Software Quality Journal, 2019, 27, 1271-1314.	2.2	18
15	Improving Bug Location Using Binary Class Relationships. , 2012, , .		16
16	Investigating the relation between lexical smells and change- and fault-proneness: an empirical study. Software Quality Journal, 2017, 25, 641-670.	2.2	15
17	MoMS: Multi-objective miniaturization of software. , 2011, , .		10
18	An experimental investigation on the effects of context on source code identifiers splitting and expansion. Empirical Software Engineering, 2014, 19, 1706-1753.	3.9	10

#	Article	IF	Citations
19	Efficient refactoring scheduling based on partial order reduction. Journal of Systems and Software, 2018, 145, 25-51.	4.5	10
20	A Fast Algorithm to Locate Concepts in Execution Traces. Lecture Notes in Computer Science, 2011, , 252-266.	1.3	9
21	Fragile base-class problem, problem?. Empirical Software Engineering, 2017, 22, 2612-2657.	3.9	8
22	A Study on the Relation between Antipatterns and the Cost of Class Unit Testing. , 2013, , .		7
23	SCAN: an approach to label and relate execution trace segments. Journal of Software: Evolution and Process, 2014, 26, 962-995.	1.6	7
24	SCAN: An Approach to Label and Relate Execution Trace Segments. , 2012, , .		5