Mika V Mantyla

List of Publications by Citations

Source: https://exaly.com/author-pdf/3756082/mika-v-mantyla-publications-by-citations.pdf
Version: 2024-04-17

This document has been generated based on the publications and citations recorded by exaly.com. For the latest version of this publication list, visit the link given above.
The third column is the impact factor (IF) of the journal, and the fourth column is the number of citations of the article.

74 papers	2,099 citations	23 h-index	44 g-index
78 ext. papers	2,837 ext. citations	2.7 avg, IF	5.71 L-index

#	Paper	IF	Citations
74	The evolution of sentiment analysis—A review of research topics, venues, and top cited papers. *Computer Science Review*, **2018**, 27, 16-32	8.3	222
73	Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology*, **2019**, 106, 101-121	3.4	163
72	Comparing and experimenting machine learning techniques for code smell detection. *Empirical Software Engineering*, **2016**, 21, 1143-1191	3.3	144
71	Subjective evaluation of software evolvability using code smells: An empirical study. *Empirical Software Engineering*, **2007**, 11, 395-431	3.3	91
70	Using metrics in Agile and Lean Software Development —A systematic literature review of industrial studies. *Information and Software Technology*, **2015**, 62, 143-163	3.4	89
69	What Types of Defects Are Really Discovered in Code Reviews?. *IEEE Transactions on Software Engineering*, **2009**, 35, 430-448	3.5	82
68	When and what to automate in software testing? A multi-vocal literature review. *Information and Software Technology*, **2016**, 76, 92-117	3.4	80
67	Code Smell Detection: Towards a Machine Learning-Based Approach **2013**,		75
66	The need for multivocal literature reviews in software engineering **2016**,		74
65	. *IEEE Software*, **2015**, 32, 64-72	1.5	70
64	Perceived causes of software project failures —An analysis of their relationships. *Information and Software Technology*, **2014**, 56, 623-643	3.4	70
63	A systematic literature review of literature reviews in software testing. *Information and Software Technology*, **2016**, 80, 195-216	3.4	61
62	The Role of the Tester's Knowledge in Exploratory Software Testing. *IEEE Transactions on Software Engineering*, **2013**, 39, 707-724	3.5	61
61	Mining valence, arousal, and dominance **2016**,		54
60	Citations, research topics and active countries in software engineering: A bibliometrics study. *Computer Science Review*, **2016**, 19, 56-77	8.3	50
59	How do testers do it? An exploratory study on manual testing practices **2009**,		50
58	On rapid releases and software testing: a case study and a semi-systematic literature review. *Empirical Software Engineering*, **2015**, 20, 1384-1425	3.3	44

57	A benchmark study on the effectiveness of search-based data selection and feature selection for cross project defect prediction. *Information and Software Technology*, **2018**, 95, 296-312	3.4	41
56	Analyzing an automotive testing process with evidence-based software engineering. *Information and Software Technology*, **2013**, 55, 1237-1259	3.4	31
55	Defect Detection Efficiency: Test Case Based vs. Exploratory Testing **2007**,		29
54	Are test cases needed? Replicated comparison between exploratory and test-case-based software testing. *Empirical Software Engineering*, **2014**, 19, 303-342	3.3	28
53	How is exploratory testing used? **2014**,		25
52	More testers The effect of crowd size and time restriction in software testing. *Information and Software Technology*, **2013**, 55, 986-1003	3.4	24
51	Who tested my software? Testing as an organizationally cross-cutting activity. *Software Quality Journal*, **2012**, 20, 145-172	1.2	23
50	Development and evaluation of a lightweight root cause analysis method (ARCA method) Field studies at four software companies. *Information and Software Technology*, **2011**, 53, 1045-1061	3.4	23
49	Benefits and limitations of automated software testing: Systematic literature review and practitioner survey **2012**,		21
48	The Effect of Team Exploratory Testing -- Experience Report from F-Secure **2016**,		21
47	On Rapid Releases and Software Testing **2013**,		19
46	Survey Reproduction of Defect Reporting in Industrial Software Development **2011**,		17
45	Drivers for software refactoring decisions **2006**,		17
44	Choosing the Right Test Automation Tool **2017**,		15
43	**2015**,		15
42	Measuring LDA topic stability from clusters of replicated runs **2018**,		15
41	Test prioritization in continuous integration environments. *Journal of Systems and Software*, **2018**, 146, 80-98	3.3	14
40	Time pressure in software engineering: A systematic review. *Information and Software Technology*, **2020**, 121, 106257	3.4	13

39	Time pressure: a controlled experiment of test case development and requirements review **2014**,		13
38	Advances in Using Agile and Lean Processes for Software Development. *Advances in Computers*, **2019**, 113, 135-224	2.9	12
37	Characterizing industry-academia collaborations in software engineering: evidence from 101 projects. *Empirical Software Engineering*, **2019**, 24, 2540-2602	3.3	12
36	Issues and Tactics when Adopting Pair Programming: A Longitudinal Case Study **2007**,		12
35	Prioritizing manual test cases in rapid release environments. *Software Testing Verification and Reliability*, **2017**, 27, e1609	0.9	11
34	A replicated study on duplicate detection **2014**,		10
33	Lightweight Elicitation and Analysis of Software Product Quality Goals: A Multiple Industrial Case Study **2009**,		9
32	Testing highly complex system of systems **2012**,		9
31	Industry-academia collaborations in software engineering **2017**,		8
30	Using experience sampling to link software repositories with emotions and work well-being **2018**,		8
29	Test Better by Exploring: Harnessing Human Skills and Knowledge. *IEEE Software*, **2016**, 33, 90-96	1.5	7
28	A tool supporting root cause analysis for synchronous retrospectives in distributed software teams. *Information and Software Technology*, **2014**, 56, 408-437	3.4	7
27	Citation and Topic Analysis of the ESEM Papers **2015**,		7
26	Characteristics of high performing testers **2010**,		7
25	What are Problem Causes of Software Projects? Data of Root Cause Analysis at Four Software Companies **2011**,		7
24	Benefitting from the Grey Literature in Software Engineering Research **2020**, 385-413		7
23	TestAWARE **2017**, 1, 1-29		6
22	Daily questionnaire to assess self-reported well-being during a software development project **2018** ,		6

21	Why are industrial agile teams using metrics and how do they use them? **2014**,		6
20	A SYSTEMATIC MAPPING STUDY OF EMPIRICAL STUDIES ON THE USE OF PAIR PROGRAMMING IN INDUSTRY. *International Journal of Software Engineering and Knowledge Engineering*, **2013**, 23, 1221-1267	1	6
19	How are software defects found? The role of implicit defect detection, individual responsibility, documents, and knowledge. *Information and Software Technology*, **2014**, 56, 1597-1612	3.4	5
18	Software Deployment Activities and Challenges - A Case Study of Four Software Product Companies **2011**,		5
17	Gamification of Software Testing - An MLR. *Lecture Notes in Computer Science*, **2016**, 611-614	0.9	5
16	A Self-assessment Instrument for Assessing Test Automation Maturity **2019**,		4
15	Supporting Regression Test Scoping with Visual Analytics **2014**,		4
14	Build Waiting Time in Continuous Integration -- An Initial Interdisciplinary Literature Review **2015**,		4
13	Diagrams or structural lists in software project retrospectives An experimental comparison. *Journal of Systems and Software*, **2015**, 103, 17-35	3.3	4
12	How to validate mobile crowdsourcing design? leveraging data integration in prototype testing **2016**,		4
11	Prevalence, Contents and Automatic Detection of KL-SATD **2020**,		3
10	Empirical software evolvability - code smells and human evaluations **2010**,		3
9	Test Automation Process Improvement in a DevOps Team: Experience Report **2020**,		3
8	Practitioner Evaluations on Software Testing Tools **2019**,		2
7	How many individuals to use in a QA task with fixed total effort? **2012**,		2
6	Predicting technical debt from commit contents: reproduction and extension with automated feature selection. *Software Quality Journal*, **2020**, 28, 1551-1579	1.2	2
5	Individual differences limit predicting well-being and productivity using software repositories: a longitudinal industrial study. *Empirical Software Engineering*, **2021**, 26, 1	3.3	2
4	Test Case Prioritization Using Test Similarities. *Lecture Notes in Computer Science*, **2018**, 243-259	0.9	1

3	Test automation maturity improves product quality Quantitative study of open source projects using continuous integration. *Journal of Systems and Software*, **2022**, 188, 111259	3.3	0
2	What Do We Know About Time Pressure in Software Development?. *IEEE Software*, **2021**, 38, 32-38	1.5	0
1	Applying Surveys and Interviews in Software Test Tool Evaluation. *Lecture Notes in Computer Science*, **2019**, 20-36	0.9	