Daniel M German

List of Publications by Year
in descending order

67	3,082	394421	434195
papers	citations	19 h-index	31 g-index

67	67	67	1301
all docs	docs citations	times ranked	citing authors

#	Article	IF	Citations
1	The promises and perils of mining GitHub. , 2014, , .		452
2	The promises and perils of mining git. , 2009, , .		220
3	Do developers update their library dependencies?. Empirical Software Engineering, 2018, 23, 384-417.	3.9	189
4	Open source software peer review practices. , 2008, , .		178
5	An in-depth study of the promises and perils of mining GitHub. Empirical Software Engineering, 2016, 21, 2035-2071.	3.9	170
6	What do large commits tell us?. , 2008, , .		144
7	The GNOME project: a case study of open source, global software development. Software Process Improvement and Practice, 2003, 8, 201-215.	1.1	115
8	On the use of visualization to support awareness of human activities in software development. , 2005, , .		106
9	Macro-level software evolution: a case study of a large software compilation. Empirical Software Engineering, 2009, 14, 262-285.	3.9	79
10	License integration patterns: Addressing license mismatches in component-based development. , 2009, , .		72
11	Will my patch make it? And how fast? Case study on the Linux kernel. , 2013, , .		70
12	Code siblings: Technical and legal implications of copying code between applications. , 2009, , .		69
13	A sentence-matching method for automatic license identification of source code files. , 2010, , .		67
14	The past, present, and future of software evolution. , 2008, , .		66
15	Software bertillonage. , 2011, , .		64
16	The Evolution of the R Software Ecosystem. , 2013, , .		60
17	An exploratory study of the evolution of software licensing. , 2010, , .		57
18	An empirical study of fine-grained software modifications. Empirical Software Engineering, 2007, 11, 369-393.	3.9	53

#	Article	IF	Citations
19	Understanding and Auditing the Licensing of Open Source Software Distributions. , 2010, , .		49
20	Open Source-Style Collaborative Development Practices in Commercial Projects Using GitHub. , 2015, , .		49
21	Software Bertillonage. Empirical Software Engineering, 2013, 18, 1195-1237.	3.9	39
22	Using software trails to reconstruct the evolution of software. Journal of Software: Evolution and Process, 2004, 16, 367-384.	1.1	38
23	Visualizing the Evolution of Systems and Their Library Dependencies. , 2014, , .		35
24	How bugs are born: a model to identify how bugs are introduced in software components. Empirical Software Engineering, 2020, 25, 1294-1340.	3.9	33
25	Trusting a library: A study of the latency to adopt the latest Maven release. , 2015, , .		32
26	The impact of context metrics on just-in-time defect prediction. Empirical Software Engineering, 2020, 25, 890-939.	3.9	30
27	Change impact graphs: Determining the impact of prior codechanges. Information and Software Technology, 2009, 51, 1394-1408.	4.4	28
28	When and why developers adopt and change software licenses. , 2015, , .		26
29	VISUALIZING THE EVOLUTION OF SOFTWARE USING SOFTCHANGE. International Journal of Software Engineering and Knowledge Engineering, 2006, 16, 5-21.	0.8	25
30	A Model to Understand the Building and Running Inter-Dependencies of Software. Reverse Engineering (WCRE), Working Conference on, 2007, , .	0.0	25
31	License usage and changes: a large-scale study on gitHub. Empirical Software Engineering, 2017, 22, 1537-1577.	3.9	25
32	On the prediction of the evolution of libre software projects. Conference on Software Maintenance, Proceedings of the, 2007, , .	0.0	24
33	Identifying licensing of jar archives using a code-search approach. , 2010, , .		24
34	An empirical study of security warnings from static application security testing tools. Journal of Systems and Software, 2019, 158, 110427.	4.5	24
35	An empirical study of unspecified dependencies in make-based build systems. Empirical Software Engineering, 2017, 22, 3117-3148.	3.9	23
36	Management of community contributions. Empirical Software Engineering, 2015, 20, 252-289.	3.9	22

#	Article	IF	Citations
37	How the R community creates and curates knowledge: an extended study of stack overflow and mailing lists. Empirical Software Engineering, 2018, 23, 953-986.	3.9	22
38	License Usage and Changes: A Large-Scale Study of Java Projects on GitHub. , 2015, , .		21
39	An empirical study of integration activities in distributions of open source software. Empirical Software Engineering, 2016, 21, 960-1001.	3.9	21
40	A Method to Detect License Inconsistencies in Large-Scale Open Source Projects. , 2015, , .		20
41	A study of the contributors of PostgreSQL. , 2006, , .		19
42	Using Software Distributions to Understand the Relationship among Free and Open Source Software Projects. , 2007, , .		19
43	How the R community creates and curates knowledge. , 2016, , .		19
44	The Debsources Dataset: two decades of free and open source software. Empirical Software Engineering, 2017, 22, 1405-1437.	3.9	17
45	Analysis of license inconsistency in large collections of open source projects. Empirical Software Engineering, 2017, 22, 1194-1222.	3.9	16
46	Visualizing Software Architecture Evolution Using Change-Sets. Reverse Engineering (WCRE), Working Conference on, 2007, , .	0.0	13
47	On the evolution of Lehman's Laws. Journal of Software: Evolution and Process, 2014, 26, 613-619.	1.6	13
48	What Is the Gist? Understanding the Use of Public Gists on GitHub. , 2015, , .		11
49	Who are Source Code Contributors and How do they Change?. , 2009, , .		10
50	Understanding the usage, impact, and adoption of non-OSI approved licenses. , 2018, , .		10
51	cregit: Token-level blame information in git version control repositories. Empirical Software Engineering, 2019, 24, 2725-2763.	3.9	10
52	Perspectives on bugs in the Debian bug tracking system. , 2010, , .		9
53	SCC++: Predicting the programming language of questions and snippets of Stack Overflow. Journal of Systems and Software, 2020, 162, 110505.	4.5	9
54	Software ingredients. , 2016, , .		7

#	Article	IF	Citations
55	Lawful software engineering. , 2010, , .		6
56	Analyzing the Relationship between the License of Packages and Their Files in Free and Open Source Software. IFIP Advances in Information and Communication Technology, 2014, , 51-60.	0.7	6
57	Remixing visualization to support collaboration in software maintenance. , 2008, , .		5
58	An analysis of open source software licensing questions in Stack Exchange sites. Journal of Systems and Software, 2022, 183, 111113.	4.5	4
59	Mergeâ€Tree: Visualizing the integration of commits into Linux. Journal of Software: Evolution and Process, 2018, 30, e1936.	1.6	3
60	Empirical Study on Dependency-related License Violation in the JavaScript Package Ecosystem. Journal of Information Processing, 2021, 29, 296-304.	0.4	3
61	Modification and developer metrics at the function level: Metrics for the study of the evolution of a software project. , 2012, , .		2
62	Merge-Tree: Visualizing the Integration of Commits into Linux. , 2016, , .		2
63	Improving scans of black and white photographs by recovering the print maker's artistic intent. Computers and Graphics, 2009, 33, 509-520.	2.5	1
64	How are Developers Treating License Inconsistency Issues? A Case Study on License Inconsistency Evolution in FOSS Projects. IFIP Advances in Information and Communication Technology, 2017, , 69-79.	0.7	1
65	REM: Visualizing the Ripple Effect on Dependencies Using Metrics of Health. , 2020, , .		1
66	1st workshop on maintenance and evolution of FLOSS (MEFLOSS). , 2008, , .		0
67	An Exploratory Study of Copyright Inconsistency in the Linux Kernel. IEICE Transactions on Information and Systems, 2021, E104.D, 254-263.	0.7	0