Earl T Barr

List of Publications by Year
in descending order

		840776	752698
47	3,439	11	20
papers	citations	h-index	g-index

47	47	47	1615
all docs	docs citations	times ranked	citing authors

#	ARTICLE	IF	CITATIONS
1	Aide-mÃ©moire: Improving a Projectâ€™s Collective Memory via Pull Requestâ€“Issue Links. ACM Transactions on Software Engineering and Methodology, 2023, 32, 1-36.	6.0	0
2	The Assessor's Dilemma: Improving Bug Repair via Empirical Game Theory. IEEE Transactions on Software Engineering, 2021, 47, 2143-2161.	5.6	7
3	Today Was a Good Day: The Daily Life of Software Developers. IEEE Transactions on Software Engineering, 2021, 47, 863-880.	5.6	39
4	Getting Ahead of the Arms Race: Hothousing the Coevolution of VirusTotal with a Packer. Entropy, 2021, 23, 395.	2.2	11
5	Artefact Relation Graphs for Unit Test Reuse Recommendation. , 2021, , .		4
6	Trident: Controlling Side Effects in Automated Program Repair. IEEE Transactions on Software Engineering, 2021, , 1-1.	5.6	2
7	Game-theoretic analysis of development practices: Challenges and opportunities. Journal of Systems and Software, 2020, 159, 110424.	4.5	13
8	Detecting Malware with Information Complexity. Entropy, 2020, 22, 575.	2.2	6
9	POSIT. , 2020, , .		5
10	A Survey of Machine Learning for Big Code and Naturalness. ACM Computing Surveys, 2019, 51, 1-37.	23.0	383
11	Approximate Oracles and Synergy in Software Energy Search Spaces. IEEE Transactions on Software Engineering, 2019, 45, 1150-1169.	5.6	19
12	The arms race: Adversarial search defeats entropy used to detect malware. Expert Systems With Applications, 2019, 118, 246-260.	7.6	27
13	Deep learning type inference. , 2018, , .		100
14	RefiNym: using names to refine types. , 2018, , .		17
15	Darwinian data structure selection. , 2018, , .		14
16	Making data-driven porting decisions with Tuscan. , 2018, , .		1
17	Mining Semantic Loop Idioms. IEEE Transactions on Software Engineering, 2018, 44, 651-668.	5.6	14
18	Understanding the syntactic rule usage in java. Journal of Systems and Software, 2017, 123, 160-172.	4.5	10

#	Article	IF	Citations
19	To Type or Not to Type: Quantifying Detectable Bugs in JavaScript. , 2017, , .		50
20	Optimising Darwinian Data Structures on Google Guava. Lecture Notes in Computer Science, 2017, , 161-167.	1.3	6
21	Time-travel debugging for JavaScript/Node.js. , 2016, , .		19
22	Casper: Automatic tracking of null dereferences to inception with causality traces. Journal of Systems and Software, 2016, 122, 52-62.	4.5	4
23	On the naturalness of software. Communications of the ACM, 2016, 59, 122-131.	4.5	158
24	Suggesting accurate method and class names. , 2015, , .		247
25	Is the cure worse than the disease? overfitting in automated program repair. , 2015, , .		203
26	The Oracle Problem in Software Testing: A Survey. IEEE Transactions on Software Engineering, 2015, 41, 507-525.	5.6	608
27	Automated software transplantation. , 2015, , .		90
28	Automated Transplantation of Call Graph and Layout Features into Kate. Lecture Notes in Computer Science, 2015, , 262-268.	1.3	17
29	The plastic surgery hypothesis. , 2014, , .		136
30	Tardis. , 2014, , .		16
31	Learning natural coding conventions. , 2014, , .		250
32	Capturing and Exploiting IDE Interactions. , 2014, , .		6
33	Comparing static bug finders and statistical prediction. , 2014, , .		76
34	Uncertainty, risk, and information value in software requirements and architecture. , 2014, , .		76
35	Tardis. ACM SIGPLAN Notices, 2014, 49, 67-82.	0.2	7
36	What effect does Distributed Version Control have on OSS project organization?. , 2013, , .		3

#	ARTICLE	IF	CITATIONS
37	Collecting a heap of shapes. , 2013, , .		4
38	Automatic detection of floating-point exceptions. , 2013, , .		72
39	Automatic detection of floating-point exceptions. ACM SIGPLAN Notices, 2013, 48, 549-560.	0.2	19
40	Reusing debugging knowledge via trace-based bug search. , 2012, , .		14
41	Liberating the programmer with prorogued programming. , 2012, , .		6
42	Reusing debugging knowledge via trace-based bug search. ACM SIGPLAN Notices, 2012, 47, 927-942.	0.2	1
43	On the naturalness of software. , 2012, , .		345
44	Cohesive and Isolated Development with Branches. Lecture Notes in Computer Science, 2012, , 316-331.	1.3	48
45	BQL. , 2011, , .		3
46	Has the bug really been fixed?. , 2010, , .		63
47	The promises and perils of mining git. , 2009, , .		220