W Eric Wong

List of Publications by Year in descending order

Source: https://exaly.com/author-pdf/2884193/publications.pdf

Version: 2024-02-01

113 papers	3,556 citations	257450 24 h-index	168389 53 g-index
122 all docs	122 docs citations	122 times ranked	1267 citing authors

#	Article	IF	Citations
1	A Survey on Software Fault Localization. IEEE Transactions on Software Engineering, 2016, 42, 707-740.	5.6	636
2	The DStar Method for Effective Software Fault Localization. IEEE Transactions on Reliability, 2014, 63, 290-308.	4.6	287
3	A family of code coverage-based heuristics for effective fault localization. Journal of Systems and Software, 2010, 83, 188-208.	4.5	178
4	Using Mutation to Automatically Suggest Fixes for Faulty Programs. , 2010, , .		163
5	Effective Fault Localization using Code Coverage. Proceedings - IEEE Computer Society's International Computer Software and Applications Conference, 2007, , .	0.0	149
6	BP NEURAL NETWORK-BASED EFFECTIVE FAULT LOCALIZATION. International Journal of Software Engineering and Knowledge Engineering, 2009, 19, 573-597.	0.8	145
7	Effective Software Fault Localization Using an RBF Neural Network. IEEE Transactions on Reliability, 2012, 61, 149-169.	4.6	142
8	Effect of test set minimization on fault detection effectiveness. Software - Practice and Experience, 1998, 28, 347-369.	3.6	125
9	Metamorphic slice: An application in spectrum-based fault localization. Information and Software Technology, 2013, 55, 866-879.	4.4	86
10	Towards Better Fault Localization: A Crosstab-Based Statistical Approach. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 2012, 42, 378-396.	2.9	85
11	Be more familiar with our enemies and pave the way forward: A review of the roles bugs played in software failures. Journal of Systems and Software, 2017, 133, 68-94.	4.5	72
12	Effective program debugging based on execution slices and inter-block data dependency. Journal of Systems and Software, 2006, 79, 891-903.	4.5	66
13	Software Fault Localization Using DStar (D*). , 2012, , .		64
14	Recent Catastrophic Accidents: Investigating How Software was Responsible. , 2010, , .		54
15	TIES WITHIN FAULT LOCALIZATION RANKINGS: EXPOSING AND ADDRESSING THE PROBLEM. International Journal of Software Engineering and Knowledge Engineering, 2011, 21, 803-827.	0.8	54
16	MSeerâ€”An Advanced Technique for Locating Multiple Bugs in Parallel. IEEE Transactions on Software Engineering, 2019, 45, 301-318.	5.6	51
17	Fault detection effectiveness of mutation and data flow testing. Software Quality Journal, 1995, 4, 69-83.	2.2	50
18	Code Coverage of Adaptive Random Testing. IEEE Transactions on Reliability, 2013, 62, 226-237.	4.6	49

#	Article	IF	Citations
19	Model-based mutation testingâ€”Approach and case studies. Science of Computer Programming, 2016, 120, 25-48.	1.9	49
20	Insights on Fault Interference for Programs with Multiple Bugs. , 2009, , .		45
21	A Grouping-Based Strategy to Improve the Effectiveness of Fault Localization Techniques. , 2010, , .		45
22	Combining mutation and fault localization for automated program debugging. Journal of Systems and Software, 2014, 90, 45-60.	4.5	44
23	Genetic Algorithm-based Test Generation for Software Product Line with the Integration of Fault Localization Techniques. Empirical Software Engineering, 2018, 23, 1-51.	3.9	41
24	Using an RBF Neural Network to Locate Program Bugs. , 2008, , .		32
25	Basic Operations for Generating Behavioral Mutants. , 2006, , .		29
26	Spectrum-Based Fault Localization: Testing Oracles are No Longer Mandatory. , 2011, , .		29
27	A bibliometric assessment of software engineering scholars and institutions (2010â€“2017). Journal of Systems and Software, 2019, 147, 246-261.	4.5	29
28	Applying design metrics to predict fault-proneness: a case study on a large-scale software system. Software - Practice and Experience, 2000, 30, 1587-1608.	3.6	28
29	Integrating Safety Analysis With Functional Modeling. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2011, 41, 610-624.	2.9	28
30	An assessment of systems and software engineering scholars and institutions (2003â€“2007 and) Tj ETQq0 0 0 rgBT /Overlock 10 Tf 50	4.5	28
31	Successes, challenges, and rethinking â€“ an industrial investigation on crowdsourced mobile application testing. Empirical Software Engineering, 2019, 24, 537-561.	3.9	26
32	Smart debugging software architectural design in SDL. Journal of Systems and Software, 2005, 76, 15-28.	4.5	25
33	Quality-Oriented Hybrid Path Planning Based on A* and Q-Learning for Unmanned Aerial Vehicle. IEEE Access, 2022, 10, 7664-7674.	4.2	25
34	Enhancing software reliability estimates using modified adaptive testing. Information and Software Technology, 2013, 55, 288-300.	4.4	23
35	Experience report: How do techniques, programs, and tests impact automated program repair?. , 2015, , .		23
36	Using Tri-Relation Networks for Effective Software Fault-Proneness Prediction. IEEE Access, 2019, 7, 63066-63080.	4.2	21

#	Article	IF	Citations
37	Bridging the Gap between Fault Trees and UML State Machine Diagrams for Safety Analysis. , 2010, , .		20
38	A consensusâ€based strategy to improve the quality of fault localization. Software - Practice and Experience, 2013, 43, 989-1011.	3.6	20
39	How does combinatorial testing perform in the real world: an empirical study. Empirical Software Engineering, 2020, 25, 2661-2693.	3.9	20
40	Software Fault Localization. , 2010, , 1147-1156.		20
41	TESTING ASPECT-ORIENTED PROGRAMS WITH UML DESIGN MODELS. International Journal of Software Engineering and Knowledge Engineering, 2008, 18, 413-437.	0.8	17
42	An assessment of systems and software engineering scholars and institutions (2002â€"2006). Journal of Systems and Software, 2009, 82, 1370-1373.	4.5	17
43	Improve the Effectiveness of Test Case Generation on EFSM via Automatic Path Feasibility Analysis. , 2011, , .		17
44	Static and dynamic distance metrics for feature-based code analysis. Journal of Systems and Software, 2005, 74, 283-295.	4.5	16
45	Software monitoring through formal specification animation. Innovations in Systems and Software Engineering, 2009, 5, 231-241.	2.1	16
46	Teaching software testing: Experiences, lessons learned and the path forward. , 2011, , .		16
47	Effective software fault localization using predicted execution results. Software Quality Journal, 2017, 25, 131-169.	2.2	16
48	A bibliometric assessment of software engineering themes, scholars and institutions (2013â€"2020). Journal of Systems and Software, 2021, 180, 111029.	4.5	15
49	An assessment of systems and software engineering scholars and institutions (2001â€"2005). Journal of Systems and Software, 2008, 81, 1059-1062.	4.5	14
50	Source code-based software risk assessing. , 2005, , .		13
51	REACHABILITY GRAPH-BASED TEST SEQUENCE GENERATION FOR CONCURRENT PROGRAMS. International Journal of Software Engineering and Knowledge Engineering, 2008, 18, 803-822.	0.8	13
52	GUI Software Fault Localization Using N-gram Analysis. , 2011, , .		13
53	WAS: A weighted attribute-based strategy for cluster test selection. Journal of Systems and Software, 2014, 98, 44-58.	4.5	13
54	AUTOMATED TEST CODE GENERATION FROM CLASS STATE MODELS. International Journal of Software Engineering and Knowledge Engineering, 2009, 19, 599-623.	0.8	12

#	Article	IF	Citations
55	Validation of SDL specifications using EFSM-based test generation. Information and Software Technology, 2009, 51, 1505-1519.	4.4	12
56	Applying Combinatorial Testing in Industrial Settings. , 2016, , .		12
57	Reliability analysis of dynamic fault trees with spare gates using conditional binary decision diagrams. Journal of Systems and Software, 2020, 170, 110766.	4.5	12
58	Towards Scalable Robustness Testing. , 2010, , .		11
59	Does Adaptive Random Testing Deliver a Higher Confidence than Random Testing?. , 2008, , .		10
60	The impacts of techniques, programs and tests on automated program repair: An empirical study. Journal of Systems and Software, 2018, 137, 480-496.	4.5	10
61	Policychain: A Decentralized Authorization Service With Script-Driven Policy on Blockchain for Internet of Things. IEEE Internet of Things Journal, 2022, 9, 5391-5409.	8.7	10
62	Improving MC/DC and Fault Detection Strength Using Combinatorial Testing. , 2017, , .		9
63	Improving Search-Based Automatic Program Repair With Neural Machine Translation. IEEE Access, 2022, 10, 51167-51175.	4.2	9
64	On the equivalence of certain fault localization techniques. , 2011, , .		8
65	On the Consensus-Based Application of Fault Localization Techniques. , 2011, , .		7
66	On the estimation of adequate test set size using fault failure rates. Journal of Systems and Software, 2011, 84, 587-602.	4.5	7
67	A Systematic Approach for Integrating Fault Trees into System Statecharts. , 2008, , .		6
68	A SEGMENT BASED APPROACH FOR THE REDUCTION OF THE NUMBER OF TEST CASES FOR PERFORMANCE EVALUATION OF COMPONENTS. International Journal of Software Engineering and Knowledge Engineering, 2009, 19, 481-505.	0.8	6
69	CT-IoT: a combinatorial testing-based path selection framework for effective IoT testing. Empirical Software Engineering, 2022, 27, 1.	3.9	6
70	Safe Software: Does It Cost More to Develop?. , 2011, , .		5
71	Analysis of software specifications based on statistics of Markov chain. , 2013, , .		5
72	Effective Test Generation for Combinatorial Decision Coverage. , 2016, , .		5

#	Article	IF	Citations
73	MSeer. , 2018, , .		5
74	An Approach to Integrating SIP in Converged Multimodal/Multimedia Communication Services. Telecommunication Systems, 2005, 28, 387-405.	2.5	4
75	Application of a Statistical Methodology to Simplify Software Quality Metric Models Constructed Using Incomplete Data Samples. Proceedings International Conference on Quality Software, 2006, , .	0.0	4
76	A STATISTICAL METHODOLOGY TO SIMPLIFY SOFTWARE METRIC MODELS CONSTRUCTED USING INCOMPLETE DATA SAMPLES. International Journal of Software Engineering and Knowledge Engineering, 2007, 17, 689-707.	0.8	4
77	A Control-Theoretic Approach to QoS Adaptation in Data Stream Management Systems Design. Proceedings - IEEE Computer Society's International Computer Software and Applications Conference, 2007, , .	0.0	4
78	An improved method to simplify software metric models constructed with incomplete data samples. , 2010, , .		4
79	Validation of SDL-based architectural design models using communication-based coverage criteria. Information and Software Technology, 2012, 54, 1418-1431.	4.4	4
80	Mining Executable Specifications of Web Applications from Selenium IDE Tests. , 2012, , .		4
81	Optimizing Test Process Action Plans by Blending Testing Maturity Model and Design of Experiments. , 2008, , .		3
82	Flexible Aspect-Oriented Design Model Checking. , 2008, , .		3
83	Visualizing Multiple Program Executions to Assist Behavior Verification. , 2009, , .		3
84	Evaluating Software Safety Standards: A Systematic Review and Comparison. , 2014, , .		3
85	Improving Software Testing Education via Industry Sponsored Contests. , 2018, , .		3
86	MCDC-Star: A White-Box Based Automated Test Generation for High MC/DC Coverage. , 2018, , .		3
87	An Algebraic Binary Decision Diagram for Analysis of Dynamic Fault Tree. , 2018, , .		3
88	What Ruined Your Cake: Impacts of Code Modifications on Bug Distribution. IEEE Access, 2020, 8, 84020-84036.	4.2	3
89	Software Fault-Proneness Analysis based on Composite Developer-Module Networks. IEEE Access, 2021, 9, 155314-155334.	4.2	3
90	Simplifying Software Metric Models via Hierarchical LASSO with Incomplete Data Samples. , 2010, , .		2

#	Article	IF	Citations
91	Involving Undergraduates in Research: Motivations and Challenges. , 2012, , .		2
92	Software engineering education via the use of corporate-sponsored projects: A panel discussion of the approaches, benefits, and challenges for industry-academic collaboration. , 2013, , .		2
93	Software Safety Standards: Evolution and Lessons Learned. , 2014, , .		2
94	Analysis of System Reliability for Cache Coherence Scheme in Multi-processor. , 2014, , .		2
95	Diagnosing SDN Network Problems by Using Spectrum-Based Fault Localization Techniques. , 2015, , .		2
96	Software-Testing Contests: Observations and Lessons Learned. Computer, 2019, 52, 61-69.	1.1	2
97	REDESIGNING LEGACY SYSTEMS INTO THE OBJECT-ORIENTED PARADIGM. International Journal of Software Engineering and Knowledge Engineering, 2004, 14, 255-276.	0.8	1
98	A Method Combining Review and Testing for Verifying Software Systems. , 2008, , .		1
99	A REVIEW APPROACH TO DETECTING VIOLATIONS OF CONSISTENCY BETWEEN SPECIFICATION AND PROGRAM STRUCTURES. International Journal of Software Engineering and Knowledge Engineering, 2008, 18, 1013-1042.	0.8	1
100	A Genetic Algorithm Based Approach for Event Synchronization Analysis in Real-Time Embedded Systems. , 2009, , .		1
101	Are Fault Failure Rates Good Estimators of Adequate Test Set Size?. , 2009, , .		1
102	Validation of SDL-Based Architectural Design Models: New Coverage Criteria. , 2011, , .		1
103	Undergraduates and research: Motivations, challenges, and the path forward. , 2013, , .		1
104	Spectrum-Base Fault Localization by Exploiting the Failure Path. , 2016, , .		1
105	Editorial: ICCCN 2001. Software - Practice and Experience, 2003, 33, 1299-1300.	3.6	0
106	Guest editorsâ€™ introduction to the special section on the software engineering track of the 22nd annual ACM symposium on applied computing (ACM SAC-SE 2007). Software Quality Journal, 2008, 16, 301-301.	2.2	0
107	PRIORITIZING COVERAGE-ORIENTED TESTING PROCESS â€” AN ADAPTIVE-LEARNING-BASED APPROACH AND CASE STUDY. Series on Software Engineering and Knowledge Engineering, 2011, , 1-21.	0.1	0
108	Program Debugging: Research and Practice. , 2014, , .		0

#	Article	IF	Citations
109	Message from the Chair of the Student Research Symposium. , 2015, , .		0
110	Safety, Security, and Reliability of Autonomous Vehicle Software. Computer, 2021, 54, 20-21.	1.1	0
111	NPI-Based Adaptive Software Rejuvenation Schedule under Random Censoring. , 2016, , .		0
112	TBEM: Testing-Based GPU-Memory Consumption Estimation for Deep Learning. IEEE Access, 2022, 10, 39674-39680.	4.2	0
113	A generalized ternary decision diagram for reliability analysis on fault-tolerant systems in different coverage models. Journal of Nuclear Science and Technology, 2022, 59, 1417-1435.	1.3	0