Forrest Shull

List of Publications by Year
in descending order

46 papers	2,008 citations	567281 15 h-index	477307 29 g-index
47 all docs	47 docs citations	47 times ranked	1139 citing authors

#	Article	IF	Citations
1	The empirical investigation of Perspective-Based Reading. Empirical Software Engineering, 1996, 1, 133-164.	3.9	360
2	Local versus Global Lessons for Defect Prediction and Effort Estimation. IEEE Transactions on Software Engineering, 2013, 39, 822-834.	5.6	202
3	Identification and management of technical debt: A systematic mapping study. Information and Software Technology, 2016, 70, 100-121.	4.4	200
4	Detecting defects in object-oriented designs. , 1999, , .		139
5	Investigating the impact of design debt on software quality. , 2011, , .		114
6	Understanding the High-Performance-Computing Community: A Software Engineer's Perspective. IEEE Software, 2008, 25, 29-36.	1.8	112
7	Knowledge-Sharing Issues in Experimental Software Engineering. Empirical Software Engineering, 2004, 9, 111-137.	3.9	93
8	A checklist for integrating student empirical studies with research and teaching goals. Empirical Software Engineering, 2010, 15, 35-59.	3.9	81
9	An empirical methodology for introducing software processes. , 2001, , .		77
10	Comparing four approaches for technical debt identification. Software Quality Journal, 2014, 22, 403-426.	2.2	72
11	Detecting defects in object-oriented designs. ACM SIGPLAN Notices, 1999, 34, 47-56.	0.2	69
12	Building empirical support for automated code smell detection. , 2010, , .		57
13	Defect categorization. , 2008, , .		53
14	Perspective-Based Reading: A Replicated Experiment Focused on Individual Reviewer Effectiveness. Empirical Software Engineering, 2006, 11, 119-142.	3.9	52
15	An empirical methodology for introducing software processes. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2001, 26, 288-296.	0.7	41
16	Organizing the technical debt landscape. , 2012, , .		29
17	A Framework for Software Engineering Experimental Replications. , 2008, , .		26
18	Exploring language support for immutability. , 2016, , .		24

#	Article	IF	Citations
19	Inspecting the History of Inspections: An Example of Evidence-Based Technology Diffusion. IEEE Software, 2008, 25, 88-90.	1.8	21
20	A Practical Approach for Quality-Driven Inspections. IEEE Software, 2007, 24, 79-86.	1.8	17
21	Are delayed issues harder to resolve? Revisiting cost-to-fix of defects throughout the lifecycle. Empirical Software Engineering, 2017, 22, 1903-1935.	3.9	17
22	An evolutionary testbed for software technology evaluation. Innovations in Systems and Software Engineering, 2005, 1, 3-11.	2.1	16
23	Chapter 5 An Environment for Conducting Families of Software Engineering Experiments. Advances in Computers, 2008, 74, 175-200.	1.6	15
24	Perfectionists in a World of Finite Resources. IEEE Software, 2011, 28, 4-6.	1.8	13
25	Understanding automated and human-based technical debt identification approaches-a two-phase study. Journal of the Brazilian Computer Society, 2019, 25, .	1.3	13
26	Tool supported detection and judgment of nonconformance in process execution. , 2009, , .		10
27	Assuring the Future? A Look at Validating Climate Model Software. IEEE Software, 2011, 28, 4-8.	1.8	10
28	Building Theories from Multiple Evidence Sources. , 2008, , 337-364.		10
29	Experimenting with software testbeds for evaluating new technologies. Empirical Software Engineering, 2007, 12, 417-444.	3.9	9
30	Simulating families of studies to build confidence in defect hypotheses. Information and Software Technology, 2005, 47, 1019-1032.	4.4	7
31	Can observational techniques help novices overcome the software inspection learning curve? An empirical investigation. Empirical Software Engineering, 2006, 11, 523-539.	3.9	7
32	The Future of Software Engineering. IEEE Software, 2016, 33, 32-35.	1.8	6
33	Crowd Sourcing the Creation of Personae Non Gratae for Requirements-Phase Threat Modeling. , 2017, , .		6
34	Fully employing software inspections data. Innovations in Systems and Software Engineering, 2012, 8, 243-254.	2.1	5
35	Using Experiments to Build a Body of Knowledge. Lecture Notes in Computer Science, 2000, , 265-282.	1.3	5
36	Evaluating the effectiveness of systems and software engineering methods, processes and tools for use in defense programs. , 2009, , .		4

#	Article	IF	Citations
37	Generating testable hypotheses from tacit knowledge for high productivity computing. , 2005, , .		3
38	Sharing Your Story. IEEE Software, 2013, 30, 4-7.	1.8	3
39	Progression, Regression, or Stasis?. IEEE Software, 2014, 31, 4-8.	1.8	2
40	3.2.2 Enhancing the System Development Process Performance: a Valueâ€Based Approach. Incose International Symposium, 2012, 22, 401-415.	0.6	1
41	Driving Innovation and Removing Barriers: Forrest Shull Takes the Helm as 2021 IEEE Computer Society President. Computer, 2021, 54, 4-9.	1.1	1
42	Evolving Defect â€œFolkloreâ€: A Cross-Study Analysis of Software Defect Behavior. Lecture Notes in Computer Science, 2006, , 1-9.	1.3	1
43	Comparing transitive to non-transitive object immutability. , 2015, , .		0
44	The Computational Research and Engineering Acquisition Tools and Environments (CREATE) Program, Part 2. Computing in Science and Engineering, 2016, 18, 7-9.	1.2	0
45	75 Years Young: Reflecting on a Milestone Year for the Society. Computer, 2021, 54, 13-17.	1.1	0
46	Succeeding Together. Computer, 2022, 55, 12-17.	1.1	0