Arie van Deursen

List of Publications by Year in descending order

Source: https://exaly.com/author-pdf/9503399/publications.pdf

Version: 2024-02-01

177

papers

6,486

citations

257450

24

h-index

149698

56

g-index

182

all docs

182

docs citations

182

times ranked

2886

citing authors

#	Article	IF	Citations
1	â€œProject smellsâ€•- Experiences in Analysing the Software Quality of ML Projects with mllint. , 2022, , .		0
2	A Theoretical and Empirical Analysis of Program Spectra Diagnosability. IEEE Transactions on Software Engineering, 2021, 47, 412-431.	5.6	4
3	Log-based software monitoring: a systematic mapping study. PeerJ Computer Science, 2021, 7, e489.	4.5	12
4	The Prevalence of Code Smells in Machine Learning projects. , 2021, , .		12
5	Search-Based Crash Reproduction and Its Impact on Debugging. IEEE Transactions on Software Engineering, 2020, 46, 1294-1317.	5.6	30
6	The Adoption of JavaScript Linters in Practice: A Case Study on ESLint. IEEE Transactions on Software Engineering, 2020, 46, 863-891.	5.6	34
7	A benchmark-based evaluation of search-based crash reproduction. Empirical Software Engineering, 2020, 25, 96-138.	3.9	22
8	Searchâ€•based crash reproduction using behavioural model seeding. Software Testing Verification and Reliability, 2020, 30, e1733.	2.0	16
9	Good things come in threes. , 2020, , .		6
10	Generating highly-structured input data by combining search-based testing and grammar-based fuzzing. , 2020, , .		9
11	Botsing, a search-based crash reproduction framework for Java. , 2020, , .		11
12	Crash reproduction using helper objectives. , 2020, , .		2
13	Effective and efficient API misuse detection via exception propagation and search-based testing. , 2019, , .		18
14	Perceived Relevance of Automatic Code Inspection in End-User Development. , 2019, , .		0
15	The Delta Maintainability Model: Measuring Maintainability of Fine-Grained Code Changes. , 2019, , .		12
16	Factors Affecting Cloud Infra-Service Development Lead Times: A Case Study at ING. , 2019, , .		0
17	Lessons learned from developing mbeddr: a case study in language engineering with MPS. Software and Systems Modeling, 2019, 18, 585-630.	2.7	20
18	The effects of change decomposition on code reviewâ€”a controlled experiment. PeerJ Computer Science, 2019, 5, e193.	4.5	4

#	Article	IF	Citations
19	Code smells for Model-View-Controller architectures. Empirical Software Engineering, 2018, 23, 2121-2157.	3.9	47
20	FEVER: An approach to analyze feature-oriented changes and artefact co-evolution in highly configurable systems. Empirical Software Engineering, 2018, 23, 905-952.	3.9	20
21	On the Effectiveness of Automatically Inferred Invariants in Detecting Regression Faults in Spreadsheets. , 2018, , .		0
22	Single-objective Versus Multi-objectivized Optimization for Evolutionary Crash Reproduction. Lecture Notes in Computer Science, 2018, , 325-340.	1.3	13
23	Hierarchical abstraction of execution traces for program comprehension. , 2018, , .		14
24	Analysing the Linux kernel feature model changes using FMDiff. Software and Systems Modeling, 2017, 16, 55-76.	2.7	16
25	Semantic versioning and impact of breaking changes in the Maven repository. Journal of Systems and Software, 2017, 129, 140-158.	4.5	58
26	The effects of perceived value and stakeholder satisfaction on software project impact. Information and Software Technology, 2017, 89, 19-36.	4.4	13
27	Effort and Cost in Software Engineering. , 2017, , .		8
28	Spreadsheet testing in practice. , 2017, , .		12
29	A Guided Genetic Algorithm for Automated Crash Reproduction. , 2017, , .		36
30	A Test-Suite Diagnosability Metric for Spectrum-Based Fault Localization Approaches. , 2017, , .		50
31	A Collaborative Approach to Teaching Software Architecture. , 2017, , .		24
32	Zero-Downtime SQL Database Schema Evolution for Continuous Deployment. , 2017, , .		9
33	Exception handling bug hazards in Android. Empirical Software Engineering, 2017, 22, 1264-1304.	3.9	29
34	Why and how JavaScript developers use linters. , 2017, , .		26
35	Software engineering without borders. , 2017, , .		0
36	Revisiting the Practical Use of Automated Software Fault Localization Techniques. , 2017, , .		13

#	Article	IF	Citations
37	An Experience Report on Applying Passive Learning in a Large-Scale Payment Company. , 2017, , .		8
38	Visualizing code and coverage changes for code review. , 2016, , .		12
39	SATT: Tailoring Code Metric Thresholds for Different Software Architectures. , 2016, , .		25
40	An exploratory study on the effects of perceived value and stakeholder satisfaction on software projects. , 2016, , .		7
41	An exploratory study on functional size measurement based on code. , 2016, , .		7
42	Success factors in managing legacy system evolution. , 2016, , .		8
43	Evolutionary testing for crash reproduction. , 2016, , .		7
44	Software that Meets Its Intent. Lecture Notes in Computer Science, 2016, , 609-625.	1.3	3
45	Continuous Deployment and Schema Evolution in SQL Databases. , 2015, , .		4
46	Beyond Page Objects: Testing Web Applications with State Objects. Queue, 2015, 13, 20-37.	1.1	4
47	Work Practices and Challenges in Pull-Based Development: The Integrator's Perspective. , 2015, , .		146
48	Pricing via Functional Size - A Case Study of a Company's Portfolio of 77 Outsourced Projects. , 2015, , .		5
49	Unveiling Exception Handling Bug Hazards in Android Based on GitHub and Google Code Issues. , 2015, , .		35
50	Tracking known security vulnerabilities in proprietary software systems. , 2015, , .		45
51	Supporting Developers' Coordination in the IDE. , 2015, , .		19
52	Detecting and refactoring code smells in spreadsheet formulas. Empirical Software Engineering, 2015, 20, 549-575.	3.9	47
53	Crawl-based analysis of web applications: Prospects and challenges. Science of Computer Programming, 2015, 97, 173-180.	1.9	14
54	Testing web applications with state objects. Communications of the ACM, 2015, 58, 36-43.	4.5	11

#	Article	IF	Citations
55	Using C language extensions for developing embedded software: a case study. , 2015, , .		16
56	Extracting feature model changes from the Linux kernel using FMDiff. , 2014, , .		21
57	Special issue on program comprehension. Empirical Software Engineering, 2014, 19, 1259-1260.	3.9	0
58	Quantifying the Encapsulation of Implemented Software Architectures. , 2014, , .		5
59	Semantic Versioning versus Breaking Changes: A Study of the Maven Repository. , 2014, , .		92
60	Towards a catalog format for software metrics. , 2014, , .		10
61	How to build a good practice software project portfolio?. , 2014, , .		14
62	Software engineering for the web: the state of the practice. , 2014, , .		11
63	What your plug-in test suites really test: an integration perspective on test suite understanding. Empirical Software Engineering, 2013, 18, 859-900.	3.9	2
64	Introduction to the special issue on mining software repositories. Empirical Software Engineering, 2013, 18, 1043-1046.	3.9	5
65	Evaluating usefulness of software metrics: An industrial experience report. , 2013, , .		17
66	Data clone detection and visualization in spreadsheets. , 2013, , .		26
67	Understanding Ajax applications by connecting client and server-side execution traces. Empirical Software Engineering, 2013, 18, 181-218.	3.9	22
68	Software metrics: Pitfalls and best practices. , 2013, , .		0
69	Automated Detection of Test Fixture Strategies and Smells. , 2013, , .		58
70	Strategies for avoiding text fixture smells during software evolution. , 2013, , .		25
71	Testing principles, current practices, and effects of change localization. , 2013, , .		2
72	The Maven repository dataset of metrics, changes, and dependencies. , 2013, , .		43

#	Article	IF	Citations
73	Communication in open source software development mailing lists. , 2013, , .		77
74	Realizing service migration in industryâ€"lessons learned. Journal of Software: Evolution and Process, 2013, 25, 639-661.	1.6	11
75	Fixing the &amp;#x2018;Out of sight out of mind&amp;#x2019; problem one year of mood-based microblogging in a distributed software team. , 2013, , .		6
76	Getting what you measure. Communications of the ACM, 2012, 55, 54-59.	4.5	51
77	Detecting code smells in spreadsheet formulas. , 2012, , .		65
78	Measuring software library stability through historical version analysis. , 2012, , .		81
79	Crawling Ajax-Based Web Applications through Dynamic Analysis of User Interface State Changes. ACM Transactions on the Web, 2012, 6, 1-30.	2.5	199
80	Test confessions: A study of testing practices for plug-in systems. , 2012, , .		29
81	Detecting and visualizing inter-worksheet smells in spreadsheets. , 2012, , .		63
82	Invariant-Based Automatic Testing of Modern Web Applications. IEEE Transactions on Software Engineering, 2012, 38, 35-53.	5.6	113
83	Quantifying the Analyzability of Software Architectures. , 2011, , .		38
84	Dependency profiles for software architecture evaluations. , 2011, , .		15
85	Performance trade-offs in client-side service delegation. , 2011, , .		0
86	Collective Code Bookmarks for Program Comprehension. , 2011, , .		19
87	A Controlled Experiment for Program Comprehension through Trace Visualization. IEEE Transactions on Software Engineering, 2011, 37, 341-355.	5.6	101
88	Studying the co-evolution of production and test code in open source and industrial developer test processes through repository mining. Empirical Software Engineering, 2011, 16, 325-364.	3.9	122
89	Empirical research in software architecture: opportunities, challenges, and approaches. Empirical Software Engineering, 2011, 16, 539-543.	3.9	6
90	Second international workshop on web 2.0 for software engineering (Web2SE 2011). , 2011, , .		0

#	Article	IF	Citations
91	Supporting professional spreadsheet users by generating leveled dataflow diagrams. , 2011, , .		66
92	Workshop report from Web2SE 2011. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2011, 36, 24-29.	0.7	1
93	Regression Testing Ajax Applications: Coping with Dynamism. , 2010, , .		45
94	Combining micro-blogging and IDE interactions to support developers in their quests. , 2010, , .		23
95	A pragmatic perspective on software visualization. , 2010, , .		1
96	Evaluation of online testing for services. , 2010, , .		22
97	Adinda. , 2010, , .		20
98	The impact of social media on software engineering practices and tools. , 2010, , .		147
99	A Lightweight Sanity Check for Implemented Architectures. IEEE Software, 2010, 27, 44-50.	1.8	11
100	Web2SE. , 2010, , .		2
101	A Cognitive Model for Software Architecture Complexity. , 2010, , .		3
102	Adopting and Evaluating Service Oriented Architecture in Industry. , 2010, , .		9
103	Understanding Plug-in Test Suites from an Extensibility Perspective. , 2010, , .		4
104	Connecting Traces: Understanding Client-Server Interactions in Ajax Applications. , 2010, , .		19
105	Research Issues in the Automated Testing of Ajax Applications. Lecture Notes in Computer Science, 2010, , 16-28.	1.3	5
106	Automatically Extracting Class Diagrams from Spreadsheets. Lecture Notes in Computer Science, 2010, , 52-75.	1.3	37
107	Workshop report from Web2SE. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2010, 35, 45-50.	0.7	3
108	Trace visualization for program comprehension: A controlled experiment. , 2009, , .		32

#	Article	IF	Citations
109	What your IDE could do once you understand your code. , 2009, , .		1
110	Automated security testing of web widget interactions. , 2009, , .		27
111	An integrated crosscutting concern migration strategy and its semi-automated application to JHotDraw. Automated Software Engineering, 2009, 16, 323-356.	2.9	16
112	Splitting a large software repository for easing future software evolutionâ€"an industrial experience report. Journal of Software: Evolution and Process, 2009, 21, 113-141.	1.1	21
113	Criteria for the evaluation of implemented architectures. , 2009, , .		6
114	Managing code clones using dynamic change tracking and resolution. , 2009, , .		35
115	Invariant-based automatic testing of AJAX user interfaces. , 2009, , .		138
116	A Systematic Survey of Program Comprehension through Dynamic Analysis. IEEE Transactions on Software Engineering, 2009, 35, 684-702.	5.6	305
117	Domain-Specific Languages in Practice: A User Study on the Success Factors. Lecture Notes in Computer Science, 2009, , 423-437.	1.3	49
118	Model-driven migration of supervisory machine control architectures. Journal of Systems and Software, 2008, 81, 517-535.	4.5	9
119	A component- and push-based architectural style for ajax applications. Journal of Systems and Software, 2008, 81, 2194-2209.	4.5	51
120	An industrial case study in reconstructing requirements views. Empirical Software Engineering, 2008, 13, 727-760.	3.9	27
121	Execution trace analysis through massive sequence and circular bundle views. Journal of Systems and Software, 2008, 81, 2252-2268.	4.5	69
122	Mining Software Repositories to Study Co-Evolution of Production &amp; Test Code. , 2008, , .		106
123	Crawling AJAX by Inferring User Interface State Changes. , 2008, , .		119
124	Using Cluster Analysis to Improve the Design of Component Interfaces. , 2008, , .		8
125	Splitting a Large Software Archive for Easing Future Software Evolution - An Industrial Experience Report using Formal Concept Analysis. , 2008, , .		4
126	Generating Version Convertors for Domain-Specific Languages. , 2008, , .		5

#	Article	IF	Citations
127	On the Interplay Between Software Testing and Evolution and its Effect on Program Comprehension. , 2008, , 173-202.		35
128	Sort-based refactoring of crosscutting concerns to aspects. , 2008, , .		3
129	Identifying Crosscutting Concerns Using Fan-In Analysis. ACM Transactions on Software Engineering and Methodology, 2007, 17, 1-37.	6.0	68
130	Simple crosscutting concerns are not so simple. , 2007, , .		25
131	SoQueT: Query-Based Documentation of Crosscutting Concerns. Proceedings - International Conference on Software Engineering, 2007, , .	0.0	16
132	Documenting Typical Crosscutting Concerns. , 2007, , .		9
133	Workshop on Technology for supporting software engineers in globally distributed contexts. , 2007, , .		0
134	Migrating Multi-page Web Applications to Single-page AJAX Interfaces. , 2007, , .		69
135	Visualizing Testsuites to Aid in Software Understanding. , 2007, , .		45
136	Understanding Execution Traces Using Massive Sequence and Circular Bundle Views. , 2007, , .		94
137	Visualisation of Domain-Specific Modelling Languages Using UML. , 2007, , .		9
138	An Integrated Crosscutting Concern Migration Strategy and its Application to JHOTDRAW. , 2007, , .		17
139	An Architectural Style for Ajax. , 2007, , .		31
140	A Comparison of Push and Pull Techniques for AJAX. , 2007, , .		58
141	Model-Driven Consistency Checking of Behavioural Specifications. , 2007, , .		14
142	A common framework for aspect mining based on crosscutting concern sorts. , 2006, , .		22
143	Migrating supervisory control architectures using model transformations. , 2006, , .		1
144	Monitoring Requirements Coverage using Reconstructed Views: An Industrial Case Study. , 2006, , .		9

#	Article	IF	Citations
145	FINT: Tool Support for Aspect Mining. , 2006, , .		7
146	An empirical study into class testability. Journal of Systems and Software, 2006, 79, 1219-1232.	4.5	102
147	Documenting software systems using types. Science of Computer Programming, 2006, 60, 205-220.	1.9	10
148	Harvesting Software Systems for MDA-Based Reengineering. Lecture Notes in Computer Science, 2006, , 213-225.	1.3	25
149	Software reverse engineering. Journal of Systems and Software, 2005, 77, 209-211.	4.5	5
150	An approach to aspect refactoring based on crosscutting concern types. , 2005, , .		8
151	Reconstructing requirements coverage views from design and test using traceability recovery via LSI. , 2005, , .		29
152	On the use of clone detection for identifying crosscutting concern code. IEEE Transactions on Software Engineering, 2005, 31, 804-818.	5.6	102
153	Guest Editor's Introduction: 10th Working Conference on Reverse Engineering. IEEE Transactions on Software Engineering, 2005, 31, 97-98.	5.6	0
154	A classification of crosscutting concerns. , 2005, , .		25
155	Isolating idiomatic crosscutting concerns. , 2005, , .		20
156	Source model analysis using the JJTraveler visitor combinator framework. Software - Practice and Experience, 2004, 34, 1345-1379.	3.6	7
157	Domain-Specific Language Design Requires Feature Descriptions. Journal of Computing and Information Technology, 2002, 10, 1.	0.3	168
158	Software architecture recovery and modelling. ACM SIGAPP Applied Computing Review: A Publication of the Special Interest Group on Applied Computing, 2002, 10, 4-7.	0.9	13
159	An empirical study into COBOL type inferencing. Science of Computer Programming, 2001, 40, 189-211.	1.9	9
160	Customer involvement in extreme programming. Software Engineering Notes: an Informal Newsletter of the Special Interest Committee on Software Engineering / ACM, 2001, 26, 70-73.	0.7	22
161	Program plan recognition for year 2000 tools. Science of Computer Programming, 2000, 36, 303-324.	1.9	5
162	Domain-specific languages. ACM SIGPLAN Notices, 2000, 35, 26-36.	0.2	899

#	Article	IF	Citations
163	Research Issues in the Renovation of Legacy Systems. Lecture Notes in Computer Science, 1999, , 1-21.	1.3	26
164	Identifying objects using cluster and concept analysis. , 1999, , .		142
165	Little languages: little maintenance?. Journal of Software: Evolution and Process, 1998, 10, 75-92.	0.4	110
166	Origin tracking. Journal of Symbolic Computation, 1993, 15, 523-545.	0.8	74
167	The Reengineering Wiki. , 0, , .		16
168	Building program understanding tools using visitor combinators. , 0, , .		3
169	Experiences in teaching software evolution and program comprehension. , 0, , .		6
170	Source-based software risk assessment. , 0, , .		30
171	Symphony: view-driven software architecture reconstruction. , 0, , .		68
172	An evaluation of clone detection techniques for identifying crosscutting concerns. , 0, , .		51
173	Identifying aspects using fan-in analysis. , 0, , .		74
174	Predicting class testability using object-oriented metrics. , 0, , .		34
175	Crosscutting Concerns in J2EE Applications. , 0, , .		14
176	Evaluating an Embedded Software Reference Architecture &#8212; Industrial Experience Report &#8212;. , 0, , .		18
177	Migration of Supervisory Machine Control Architectures. , 0, , .		1