

# Pekka Abrahamsson

## List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/4483385/publications.pdf>

Version: 2024-02-01

155  
papers

4,510  
citations

182225

30  
h-index

175968

55  
g-index

163  
all docs

163  
docs citations

163  
times ranked

2339  
citing authors

#	ARTICLE	IF	CITATIONS
1	How Do Software Companies Deal with Artificial Intelligence Ethics? A Gap Analysis. , 2022, , .		5
2	Using Blockchain in Digitalizing Enterprise Legacy Systems: An Experience Report. Lecture Notes in Business Information Processing, 2021, , 70-85.	0.8	4
3	How to Write Ethical User Stories? Impacts of the ECCOLA Method. Lecture Notes in Business Information Processing, 2021, , 36-52.	0.8	8
4	The entrepreneurial logic of startup software development: A study of 40 software startups. Empirical Software Engineering, 2021, 26, 1.	3.0	11
5	ECCOLA â€” A method for implementing ethically aligned AI systems. Journal of Systems and Software, 2021, 182, 111067.	3.3	31
6	The Startup Scratch Book â€” Opening the Black Box of Startup Education. Lecture Notes in Business Information Processing, 2021, , 193-200.	0.8	0
7	Blockchain Governance: A Dynamic View. Lecture Notes in Business Information Processing, 2021, , 66-80.	0.8	2
8	Internal Software Startups â€” A Multiple Case Study on Practices, Methods, and Success Factors. , 2020, , .		2
9	Business Model Canvas Should Pay More Attention to the Software Startup Team. , 2020, , .		2
10	ECCOLA - a Method for Implementing Ethically Aligned AI Systems. , 2020, , .		10
11	The Current State of Industrial Practice in Artificial Intelligence Ethics. IEEE Software, 2020, 37, 50-57.	2.1	48
12	Software Startup Education: Gamifying Growth Hacking. , 2020, , 269-277.		2
13	Startup Metrics That Tech Entrepreneurs Need to Know. , 2020, , 111-127.		3
14	Software Startup ESSENCE: How Should Software Startups Work?. , 2020, , 97-109.		1
15	â€œThis is Just a Prototypeâ€” How Ethics Are Ignored in Software Startup-Like Environments. Lecture Notes in Business Information Processing, 2020, , 195-210.	0.8	23
16	Software Startup Practices â€” Software Development in Startups Through the Lens of the Essence Theory of Software Engineering. Lecture Notes in Computer Science, 2020, , 402-418.	1.0	3
17	Ethically Aligned Design: An Empirical Evaluation of the RESOLVEDD-Strategy in Software and Systems Development Context. , 2019, , .		7
18	A Tool-Based Approach for Essentializing Software Engineering Practices. , 2019, , .		1

#	ARTICLE	IF	CITATIONS
19	Implementing Ethics in AI: Initial Results of an Industrial Multiple Case Study. Lecture Notes in Computer Science, 2019, , 331-338.	1.0	14
20	Containers in Software Development: A Systematic Mapping Study. Lecture Notes in Computer Science, 2019, , 176-191.	1.0	7
21	What happens when software developers are (un)happy. Journal of Systems and Software, 2018, 140, 32-47.	3.3	124
22	Lean Internal Startups for Software Product Innovation in Large Companies: Enablers and Inhibitors. Journal of Systems and Software, 2018, 135, 69-87.	3.3	47
23	Innovation Initiatives in Large Software Companies: A Systematic Mapping Study. Information and Software Technology, 2018, 95, 1-14.	3.0	16
24	The Essence Theory of Software Engineering – Large-Scale Classroom Experiences from 450+ Software Engineering BSc Students. Lecture Notes in Computer Science, 2018, , 123-138.	1.0	5
25	A preliminary study of agility in business and production. , 2018, , .		6
26	Factors and actors leading to the adoption of a JavaScript framework. Empirical Software Engineering, 2018, 23, 3503-3534.	3.0	20
27	Female Leadership in Software Projects – A Preliminary Result on Leadership Style and Project Context Factors. Studies in Computational Intelligence, 2018, , 149-163.	0.7	4
28	Are Software Startups Applying Agile Practices? The State of the Practice from a Large Survey. Lecture Notes in Business Information Processing, 2017, , 167-183.	0.8	44
29	Start-Ups Must Be Ready to Pivot. IEEE Software, 2017, 34, 18-22.	2.1	26
30	On the Unhappiness of Software Developers. , 2017, , .		46
31	Security challenges in IoT development. , 2017, , .		23
32	Exploring the outsourcing relationship in software startups. , 2017, , .		6
33	Unhappy Developers: Bad for Themselves, Bad for Process, and Bad for Software Product. , 2017, , .		25
34	Consequences of Unhappiness while Developing Software. , 2017, , .		19
35	“Failures” to be celebrated: an analysis of major pivots of software startups. Empirical Software Engineering, 2017, 22, 2373-2408.	3.0	74
36	Future directions in Agile research: Alignment and divergence between research and practice. Journal of Software: Evolution and Process, 2017, 29, e1884.	1.2	6

#	ARTICLE	IF	CITATIONS
37	What Influences the Speed of Prototyping? An Empirical Investigation of Twenty Software Startups. Lecture Notes in Business Information Processing, 2017, , 20-36.	0.8	21
38	Towards Understanding Startup Product Development as Effectual Entrepreneurial Behaviors. Lecture Notes in Computer Science, 2017, , 265-279.	1.0	8
39	Towards Understanding Startup Product Development as Effectual Entrepreneurial Behaviors. Lecture Notes in Business Information Processing, 2017, , 199-204.	0.8	1
40	Bringing the Cloud to Rural and Remote Areas via Cloudlets. , 2016, , .		8
41	The Perception of Technical Debt in the Embedded Systems Domain: An Industrial Case Study. , 2016, , .		22
42	Product Innovation through Internal Startup in Large Software Companies: A Case Study. , 2016, , .		2
43	How Do Software Startups Pivot? Empirical Results from a Multiple Case Study. Lecture Notes in Business Information Processing, 2016, , 169-176.	0.8	14
44	Key Challenges in Software Startups Across Life Cycle Stages. Lecture Notes in Business Information Processing, 2016, , 169-182.	0.8	31
45	Software Development in Startup Companies: The Greenfield Startup Model. IEEE Transactions on Software Engineering, 2016, 42, 585-604.	4.3	124
46	Minimum Viable Product or Multiple Facet Product? The Role of MVP in Software Startups. Lecture Notes in Business Information Processing, 2016, , 118-130.	0.8	45
47	Cloud computing implementation explained: A tale of two SMEs. , 2015, , .		1
48	Do feelings matter? On the correlation of affects and the self-assessed productivity in software engineering. Journal of Software: Evolution and Process, 2015, 27, 467-487.	1.2	56
49	In Need of Creative Mobile Service Ideas? Forget Adults and Ask Young Children. SAGE Open, 2015, 5, 215824401560171.	0.8	5
50	Key Challenges in Early-Stage Software Startups. Lecture Notes in Business Information Processing, 2015, , 52-63.	0.8	69
51	Lean startup. , 2015, , .		16
52	The Affect of Software Developers: Common Misconceptions and Measurements. , 2015, , .		14
53	Understanding the affect of developers: theoretical background and guidelines for psychoempirical software engineering. , 2015, , .		34
54	A survey study on major technical barriers affecting the decision to adopt cloud services. Journal of Systems and Software, 2015, 103, 167-181.	3.3	68

#	ARTICLE	IF	CITATIONS
55	Performance Alignment Work: How software developers experience the continuous adaptation of team performance in Lean and Agile environments. <i>Information and Software Technology</i> , 2015, 64, 132-147.	3.0	56
56	Hunter-gatherer cycle: a conceptual model of the evolution of software startups. , 2015, , .		36
57	A Simultaneous, Multidisciplinary Development and Design Journey â€œ Reflections on Prototyping. <i>Lecture Notes in Computer Science</i> , 2015, , 409-416.	1.0	15
58	Bridging Tangible and Virtual Interaction: Rapid Prototyping of a Gaming Idea. <i>Lecture Notes in Computer Science</i> , 2015, , 523-528.	1.0	1
59	What Can Software Startups Learn from the Artistic Design Flow? Experiences, Reflections and Future Avenues. <i>Lecture Notes in Computer Science</i> , 2015, , 584-599.	1.0	1
60	Software Product Size Measurement Methods: A Systematic Mapping Study. , 2014, , .		5
61	Software Developers, Moods, Emotions, and Performance. <i>IEEE Software</i> , 2014, 31, 24-27.	2.1	53
62	Why Early-Stage Software Startups Fail: A Behavioral Framework. <i>Lecture Notes in Business Information Processing</i> , 2014, , 27-41.	0.8	86
63	How do software developers experience team performance in lean and agile environments?. , 2014, , .		12
64	A framework for systematic analysis of open access journals and its application in software engineering and information systems. <i>Scientometrics</i> , 2014, 101, 1627-1656.	1.6	12
65	Software development in startup companies: A systematic mapping study. <i>Information and Software Technology</i> , 2014, 56, 1200-1218.	3.0	292
66	Microblogging in Open Source Software Development: The Case of Drupal and Twitter. <i>IEEE Software</i> , 2014, 31, 72-80.	2.1	23
67	Feature Usage as a Value Indicator for Decision Making. , 2014, , .		5
68	Software Business. Towards Continuous Value Delivery. <i>Lecture Notes in Business Information Processing</i> , 2014, , .	0.8	3
69	What Do We Know about Software Development in Startups?. <i>IEEE Software</i> , 2014, 31, 28-32.	2.1	127
70	Automated Feature Identification in Web Applications. <i>Lecture Notes in Business Information Processing</i> , 2014, , 100-114.	0.8	4
71	Happy software developers solve problems better: psychological measurements in empirical software engineering. <i>PeerJ</i> , 2014, 2, e289.	0.9	115
72	Self-organized Learning in Software Factory: Experiences and Lessons Learned. <i>Lecture Notes in Business Information Processing</i> , 2014, , 126-142.	0.8	4

#	ARTICLE	IF	CITATIONS
73	Affordable and Energy-Efficient Cloud Computing Clusters: The Bolzano Raspberry Pi Cloud Cluster Experiment. , 2013, , .		62
74	Feature Usage Diagram for Feature Reduction. Lecture Notes in Business Information Processing, 2013, , 223-237.	0.8	3
75	Are Happy Developers More Productive?. Lecture Notes in Computer Science, 2013, , 50-64.	1.0	46
76	Foundations and Technological Landscape of Cloud Computing. , 2013, 2013, 1-31.		19
77	Definitions of Agile Software Development and Agility. Communications in Computer and Information Science, 2013, , 247-258.	0.4	27
78	Making Sense Out of a Jungle of JavaScript Frameworks. Lecture Notes in Computer Science, 2013, , 334-337.	1.0	14
79	A Web-based modeling tool for the SEMAT Essence theory of software engineering. Journal of Open Research Software, 2013, 1, e4.	2.7	12
80	Agile Project â€“ An Oxymoron? Proposing an Unproject Leadership Model for Complex Space. Lecture Notes in Business Information Processing, 2013, , 194-209.	0.8	0
81	Exploring How Feature Usage Relates to Customer Perceived Value: A Case Study in a Startup Company. Lecture Notes in Business Information Processing, 2013, , 166-177.	0.8	11
82	Towards a Conceptual Framework for Assessing the Benefits of Cloud Computing. Lecture Notes in Business Information Processing, 2013, , 141-152.	0.8	1
83	Lean Software Development. IEEE Software, 2012, 29, 22-25.	2.1	45
84	Does cloud computing deliver the promised benefits for IT industry?. , 2012, , .		13
85	Strengths and barriers behind the successful agile deploymentâ€”insights from the three software intensive companies in Finland. Empirical Software Engineering, 2012, 17, 675-702.	3.0	65
86	Making the leap to a software platform strategy: Issues and challenges. Information and Software Technology, 2012, 54, 968-984.	3.0	32
87	Predicting Development Effort from User Stories. , 2011, , .		39
88	Analyzing tool usage to understand to what extent experts change their activities when mentoring. , 2011, , .		0
89	On the Impact of Kanban on Software Project Work: An Empirical Case Study Investigation. , 2011, , .		27
90	Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. Information and Software Technology, 2011, 53, 276-290.	3.0	165

#	ARTICLE	IF	CITATIONS
91	Software product roadmapping in a volatile business environment. Journal of Systems and Software, 2011, 84, 958-975.	3.3	23
92	Operationalizing the Concept of Success in Software Engineering Projects. International Journal of Innovation in the Digital Economy, 2011, 2, 11-37.	0.2	2
93	The set-up of a software engineering research infrastructure of the 2010s. , 2010, , .		13
94	Agile Software Development Methods: A Comparative Review1. , 2010, , 31-59.		41
95	Building blocks for self-organizing software development teams a framework model and empirical pilot study. , 2010, , .		15
96	Agility and Architecture: Can They Coexist?. IEEE Software, 2010, 27, 16-22.	2.1	126
97	Exploring the Sources of Waste in Kanban Software Development Projects. , 2010, , .		21
98	Anticipating Success of a Business-Critical Software Project: A Comparative Case Study of Waterfall and Agile Approaches. Lecture Notes in Business Information Processing, 2010, , 187-192.	0.8	4
99	XP Workshop on Agile Product Line Engineering. Lecture Notes in Business Information Processing, 2009, , 215-216.	0.8	0
100	Agile Processes in Software Engineering and Extreme Programming. Lecture Notes in Business Information Processing, 2009, , .	0.8	3
101	“Lots done, more to do”: the current state of agile systems development research. European Journal of Information Systems, 2009, 18, 281-284.	5.5	180
102	Positioning Agility. Lecture Notes in Business Information Processing, 2009, , 206-208.	0.8	1
103	Message from Program Chair. , 2009, , .		0
104	Architecture-Centric Methods and Agile Approaches. Lecture Notes in Business Information Processing, 2009, , 232-233.	0.8	0
105	The impact of agile practices on communication in software development. Empirical Software Engineering, 2008, 13, 303-337.	3.0	228
106	Perspectives on Global Software Development: special issue on PROFES 2007. Software Process Improvement and Practice, 2008, 13, 213-215.	1.1	0
107	Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. IET Software, 2008, 2, 58.	1.5	145
108	Scrum in a Multiproject Environment: An Ethnographically-Inspired Case Study on the Adoption Challenges. , 2008, , .		20

#	ARTICLE	IF	CITATIONS
109	A Preliminary Roadmap for Empirical Research on Agile Software Development. , 2008, , .		31
110	Architecture-Centric Methods and Agile Approaches. Lecture Notes in Business Information Processing, 2008, , 242-243.	0.8	11
111	Does Test-Driven Development Improve the Program Code? Alarming Results from a Comparative Case Study. Lecture Notes in Computer Science, 2008, , 143-156.	1.0	19
112	A Case Study on the Impact of Refactoring on Quality and Productivity in an Agile Team. Lecture Notes in Computer Science, 2008, , 252-266.	1.0	67
113	Culture and Agile: Challenges and Synergies. Lecture Notes in Business Information Processing, 2008, , 251-255.	0.8	0
114	Communication in Distributed Agile Development: A Case Study. , 2007, , .		46
115	A Comparative Case Study on the Impact of Test-Driven Development on Program Design and Test Coverage. , 2007, , .		21
116	A Comparative Case Study on the Impact of Test-Driven Development on Program Design and Test Coverage. First International Symposium on Empirical Software Engineering and Measurement (ESEM) Tj ETQq0 0 0 ngBT /Overlock 10 T		
117	Effort Prediction in Iterative Software Development Processes -- Incremental Versus Global Prediction Models. , 2007, , .		31
118	A Critical Analysis of Empirical Research in Software Testing. , 2007, , .		36
119	Agile Software Development of Mobile Information Systems. Notes on Numerical Fluid Mechanics and Multidisciplinary Design, 2007, , 1-4.	0.2	10
120	An iterative improvement process for agile software development. Software Process Improvement and Practice, 2007, 12, 81-100.	1.1	44
121	Predicting Software Defect Density: A Case Study on Automated Static Code Analysis. , 2007, , 137-140.		5
122	Agile Software Development: Theoretical and Practical Outlook. Lecture Notes in Computer Science, 2007, , 410-411.	1.0	2
123	Agile Software Development Quality Assurance. , 2007, , .		15
124	Does Refactoring Improve Reusability?. Lecture Notes in Computer Science, 2006, , 287-297.	1.0	46
125	Guest editorial for the special section on distributed software development. Information and Software Technology, 2006, 48, 765-766.	3.0	6
126	Agile Software Development Methods: When and Why Do They Work?. , 2005, , 371-373.		1



#	ARTICLE	IF	CITATIONS
127	On establishing the essential components of a technology-dependent framework. , 2005, , .		3
128	A multiple case study on the impact of pair programming on product quality. , 2005, , .		69
129	A Case Study on Naked Objects in Agile Software Development. Lecture Notes in Computer Science, 2005, , 189-197.	1.0	1
130	Improving Business Agility Through Technical Solutions: A Case Study on Test-Driven Development in Mobile Software Development. , 2005, , 227-243.		14
131	Extreme Programming: Reassessing the Requirements Management Process for an Offsite Customer. Lecture Notes in Computer Science, 2004, , 12-22.	1.0	3
132	On-Site Customer in an XP Project: Empirical Results from a Case Study. Lecture Notes in Computer Science, 2004, , 1-11.	1.0	29
133	Usability Assessment of an Extreme Programming Project: Close Co-operation with the Customer Does Not Equal to Good Usability. Lecture Notes in Computer Science, 2004, , 393-407.	1.0	26
134	Empirical Evaluation of Agile Software Development: The Controlled Case Study Approach. Lecture Notes in Computer Science, 2004, , 408-423.	1.0	30
135	Extreme programming: a survey of empirical data from a controlled case study. , 2004, , .		38
136	Mobile-D. , 2004, , .		99
137	Achieving CMMI Level 2 with Enhanced Extreme Programming Approach. Lecture Notes in Computer Science, 2004, , 378-392.	1.0	18
138	Self-Adaptability of Agile Software Processes: A Case Study on Post-iteration Workshops. Lecture Notes in Computer Science, 2004, , 184-193.	1.0	17
139	Digging into the fundamentals of extreme programming building the theoretical base for agile methods. , 2003, , .		8
140	Extreme programming: first results from a controlled case study. , 2003, , .		28
141	Software Development under Stringent Hardware Constraints: Do Agile Methods Have a Chance?. Lecture Notes in Computer Science, 2003, , 73-79.	1.0	42
142	Experimental software engineering (STESE). , 2003, , .		1
143	Commitment Nets in Software Process Improvement. Annals of Software Engineering, 2002, 14, 407-438.	0.5	11
144	Personal Software Process: Classroom Experiences from Finland. Lecture Notes in Computer Science, 2002, , 175-185.	1.0	17

#	ARTICLE	IF	CITATIONS
145	Title is missing!. Software Quality Journal, 1999, 8, 63-76.	1.4	7
146	Is management commitment a necessity after all in software process improvement?. , 0, , .		7
147	The personal software process: experiences from Denmark. , 0, , .		6
148	The interaction between organizational subcultures and user-centered design-a case study of an implementation effort. , 0, , .		12
149	Commitment in software process improvement - in search of the process. , 0, , .		9
150	Integrating agile software development and software process improvement: a longitudinal case study. , 0, , .		14
151	Providing Test Quality Feedback Using Static Source Code and Automatic Test Suite Metrics. , 0, , .		19
152	Naked Objects versus Traditional Mobile Platform Development: A Comparative Case Study. , 0, , .		3
153	A Case Study on the Impact of Customer Communication on Defects in Agile Software Development.. , 0, , .		30
154	How do you feel, developer? An explanatory theory of the impact of affects on programming performance. PeerJ Computer Science, 0, 1, e18.	2.7	41
155	Operationalizing the Concept of Success in Software Engineering Projects. , 0, , 89-116.		0