Appendix

The current appendix summarizes the results of the systematic literature review performed in the context of the article "Quality Attributes Use in Architecture Design Decision Methods: Research and Practice" in the form of informative tables. In particular, it contains the list of selected papers along with the corresponding publication venue and year as well as the detailed evaluation of each of the selected approaches and tools with respect to the three research questions analyzed in the article.

Publication	Tool/Approach Name	Venue	Year
Kishi et al. [1]	-	APSEC	2001
Rosa et al. [2]	Parmenides	SAC	2001
Bachmann et al. [3]	-	STRAW	2003
Chung et al. [4]	Proteus	Computer Standards & Interfaces	2003
Svahnberg et al. [5]	UML4PF	IJSEKE	2003
Al-Naeem et al. [6]	ArchDesigner	ICSE	2005
Choi et al. [7]	AQUA	FMOODS	2006
Tibermacine et al. [8]	-	CBSE	2006
Babar and Gorton [9]	PAKME	SHARK	2007
Harrison and Avgeriou [10]	-	ECSA	2007
Zdun [11]	-	Software Practice & Experience	2007
Zimmermann et al. [12]	RADM	QoSA	2007
Babar and Capilla [13]	PAKME	MARK	2008
Cui et al. [14]	-	WICSA	2008
Makki et al. [15]	ADD+	ECSA	2008
Zimmermann et al. [16]	ArchPad	WICSA	2008
de Boer et al. [17]	AURES	WICSA/ECSA	2009
Bode and Riebisch [18]	-	ECSA	2010
Xu et al. [19]	-	UIC-ATC '10	2010
Alebrahim et al. [20]	-	APSEC	2011
Kassab et al. [21]	-	SERA	2011
Dermeval et al. [22]	STREAM-ADD	COMPSAC	2012
van Heesch et al. [23]	-	WICSA/ECSA	2012
Shen et al. [24]	QuOnt	COMPSAC	2012
Lytra et al. [25]	ADvISE	ECSA	2013
Nowak and Pautasso [26]	Software Architecture Warehouse	ECSA	2013
Ameller and Franch [27]	ArchiTech (Quark Method)	CLEI electronic journal	2014
Lopes Silva et al. [28]	-	SAC	2015
Lytra et al. [29]	CoCoADvISE	SESoS	2015
Saadatmand and Tahvili [30]	-	ITNG	2015
Me et al. [31]	-	QRASA	2016
Monteserin et al. [32]	DesignBots	PAMS	2017
Carrillo and Capilla [33]	-	ECSA	2018
Malakuti et al. [34]	-	ECSA	2018
Sedaghatbaf and Abdollahi Azgomi [35]	SQMETool	Software and Systems Modeling	2018
Schneider et al. [36]	Extension of Palladio and PerOpteryx	ECSA	2018

Table 1: **List of papers included in the systematic literature review.** The selected papers are listed along with the publication year and venue as well as the name of the method or tool used for architecture decision making and documentation. The majority of the publications (28) are from the years 2007–2018 which gives an indication of the increasing interest of the software architecture community in ADDs in recent years. In most of the cases, the proposals are accompanied by tools for software architects. More specifically, we analyzed 29 tools for decision making, 2 for decision documentation, and 5 for both tasks (see also 2).

Publication	Appearance of QAs	Relationships ADDs-QAs	Evaluation of ADDs with QAs and QAs using ADDS
Kishi et al.[1]	Decision-Making	Explicit	Fully used for evaluation
Rosa et al.[2]	Decision-Making	Not explicit	Not supported
Bachmann et al.[3]	Decision-Making	Explicit	Fully used for evaluation
Chung et al.[4]	Decision-Making	Explicit	Fully used for evaluation
Svahnberg et al.[5]	Decision-Making	Not explicit	Partially used
Al-Naeem et al.[6]	Decision-Making	Explicit	Fully used for evaluation
Choi et al.[7]	Both	Explicit	Fully used for evaluation
Tibermacine et al.[8]	Documentation	Explicit	Fully used for evaluation
Babar and Gorton[9]	Both	Supported but not explicit	Partially used
Harrison and Avgeriou[10]	Decision-Making	Supported but not explicit	Fully used for evaluation
Zdun[11]	Decision-Making	Explicit	Fully used for evaluation
Zimmermann et al.[12]	Decision-Making	Explicit	Partially used
Babar and Capilla[13]	Decision-Making	Explicit	Fully used for evaluation
Cui et al.[14]	Decision-Making	Explicit	Fully used for evaluation
Makki et al.[15]	Decision-Making	Explicit	Fully used for evaluation
Zimmermann et al.[16]	Decision-Making	Explicit	Not supported
de Boer et al.[17]	Decision-Making	Explicit	Fully used for evaluation
Bode and Riebisch[18]	Decision-Making	Explicit	Fully used for evaluation
Xu et al.[19]	Decision-Making	Explicit	Not supported
Alebrahim et al.[20]	Decision-Making	Not explicit	Not supported
Kassab et al.[21]	Decision-Making	Not explicit	Partially used
Dermeval et al.[22]	Both	Explicit	Partially used
van Heesch et al.[23]	Documentation	Explicit	Fully used for evaluation
Shen et al.[24]	Decision-Making	Explicit	Not supported
Lytra et al.[25]	Both	Supported but not explicit	Not supported
Nowak and Pautasso[26]	Decision-Making	Supported but not explicit	Not supported
Ameller and Franch[27]	Decision-Making	Explicit	Fully used for evaluation
Lopes Silva et al.[28]	Decision-Making	Explicit	Fully used for evaluation
Lytra et al.[29]	Both	Explicit	Fully used for evaluation
Saadatmand and Tahvili[30]	Decision-Making	Not explicit	Partially used
Me et al.[31]	Decision-Making	Explicit	Fully used for evaluation
Monteserin et al.[32]	Decision-Making	Explicit	Fully used for evaluation
Carrillo and Capilla [33]	Decision-Making	Not explicit	Not supported
Malakuti et al. [34]	Decision-Making	Explicit	Fully used for evaluation
Sedaghatbaf and Abdollahi Azgomi [35]	Decision-Making	Explicit	Fully used for evaluation
Schneider et al. [36]	Decision-Making	Explicit	Fully used for evaluation

Table 2: **Evaluation of selected approaches with respect to RQ1.** The following aspects are being studied: 1) Appearance of QAs (in decision making, documentation or both processes; 2) Support for relationships between ADDs and QAs (i.e., whether they are described explicitly or not); 3) Support for evaluation of ADDs using QAs and vice versa with possible values “not supported”, “not indicated”, “partially used”, “captured but not used”, and “fully used for evaluation”.

Publication	QA Uncertainty	QA Interdependencies	QA Trade-offs
Kishi et al.[1]	■	□	■
Rosa et al.[2]	□	■	□
Bachmann et al.[3]	□	□	□
Chung et al.[4]	□	■	■
Svahnberg et al.[5]	□	■	■
Al-Naeem et al.[6]	□	■	■
Choi et al.[7]	□	□	□
Tibermacine et al.[8]	□	□	□
Babar and Gorton[9]	□	□	■
Harrison and Avgeriou[10]	□	□	□
Zdun[11]	■	□	■
Zimmermann et al.[12]	□	□	■
Babar and Capilla[13]	□	■	■
Cui et al.[14]	□	□	■
Makki et al.[15]	□	■	■
Zimmermann et al.[16]	□	□	■
de Boer et al.[17]	□	■	■
Bode and Riebisch[18]	■	■	□
Xu et al.[19]	□	□	□
Alebrahim et al.[20]	□	□	□
Kassab et al.[21]	■	□	□
Dermeval et al.[22]	□	□	■
van Heesch et al.[23]	■	□	□
Shen et al.[24]	□	■	■
Lytra et al.[25]	□	□	□
Nowak and Pautasso[26]	□	□	■
Ameller and Franch[27]	■	□	■
Lopes Silva et al.[28]	□	■	■
Lytra et al.[29]	■	■	□
Saadatmand and Tahvili[30]	■	■	■
Me et al.[31]	□	□	□
Monteserin et al.[32]	□	□	■
Carrillo and Capilla [33]	□	□	□
Malakuti et al. [34]	□	□	□
Sedaghatbaf and Abdollahi Azgomi [35]	■	■	■
Schneider et al. [36]	■	■	■

Table 3: **Evaluation of selected approaches with respect to RQ2.** blueThe QA-related challenges that are addressed in existing tools and methods for architecture decision making and documentation are the following: 1) QA Uncertainty: Uncertainty is caused by vague, incomplete, or imprecise information about QAs of design solutions and requirements. An approach that supports dealing with uncertainty provides means for expressing and/or resolving QA uncertainty; 2) QA Interdependencies: QAs may have positive or negative impact on other QAs. Apart from that, prioritization of QAs is often considered in architecture decision making; 3) QA Trade-offs: Making ADDs is essentially the result of making trade-offs between competing requirements and stakeholders' concerns. Full boxes indicate support while empty boxes lack of support.

Publication	Automation Level	Method used for Trade-offs
Kishi et al.[1]	Semi-automatic	Architectural design technique consisting of multiple steps considering multiple quality attributes
Chung et al.[4]	Manual	A trade-off analysis leads to the selection among the competing design patterns, hence among the alternative architectures
Svahnberg et al.[5]	Automatic	Analytic Hierarchy Process is used to prioritize software architecture structures with respect to a quality attribute
Al-Naeem et al.[6]	Semi-automatic	Analytic Hierarchy Process is used to calculate value scores for design alternatives considering their impact on QAs and the stakeholders' preferences
Babar and Gorton[9]	Manual	Trade-offs are achieved by reusing pattern-based AK in elicited scenarios
Zdun[11]	Manual	Visual structures similar to QOC structures are used
Zimmermann et al.[12]	Manual	See RADM
Babar and Capilla[13]	Manual	Trade-offs are supported with the aid of utility trees
Cui et al.[14]	Manual	Architects select from synthesized architecture solutions according to their pros and cons with respect to FRs and NFRs
Makki et al.[15]	Semi-automatic	The process of stakeholders' preference elicitation and architecture decision making are formalized as a multi-attribute decision problem
Zimmermann et al.[16]	Manual	Reusable decision models are mainly based on patterns, and patterns are considered to provide trade-offs between QAs. In addition, trade-offs are supported by SWOT analysis tables and QOC diagrams
de Boer et al.[17]	Semi-automatic	Partial ordering is used to calculate scores for QAs based on QA prioritization and QA dependencies
Dermeval et al.[22]	Manual	QA trade-offs are made by considering the fulfillment and the priorities of softgoals and NFRs with regard to the design options
Shen et al.[24]	Automatic	A SAT solver is used for resolving trade-offs
Nowak and Pautasso[26]	Manual	Trade-offs are made in a group discussion after voting on the advantages/disadvantages of alternative design solutions
Ameller and Franch[27]	Semi-automatic	Trade-offs are made by solving a constraint satisfaction problem based on constraints posed by required qualities and QA priorities
Lopes Silva et al.[28]	Semi-automatic	QA trade-offs are made with the aid of an Expert System
Saadatmand and Tahvili[30]	Automatic	TOPSIS, a fuzzy optimization method for multi-criteria decision analysis is used
Monteserin et al.[32]	Manual	Trade-offs are made after a systematic exploration of the design alternatives
Sedaghatbaf and Abdollahi Azgomi [35]	Automatic	Calculation of trade-offs is considered as MCDM problem (use of TOPSIS techique)
Schneider et al. [36]	Automatic	Calculation of Pareto-optimal results

Table 4: **List of approaches supporting QA trade-offs.** Three values are available for describing the automation level (manual, semi-automatic, automatic); for each approach the corresponding method used for performing QA trade-offs is summarized. From the 21 approaches under study which support QA trade-offs only 5 provide automatic support while the majority (10) describe a manual process for making trade-offs.

Publication	Size & Scope of Design Space	Evaluation Method
Kishi et al.[1]	6 design decisions	Case Study: on-board system for ITS systems
Bachmann et al.[3]	> 10 tactics	Garage Door Example
Chung et al.[4]	3 patterns and 6 tactics	Example: Home Appliance Controller
Svahnberg et al.[5]	3 alternative architectures	Case study with Swedish company
Al-Naeem et al.[6]	19 design decisions	Case study: Glass Box project
Choi et al.[7]	8 design decisions	Example: House Alarm System
Tibermacine et al.[8]	6 design decisions	Motivating example: Museum Access Control System – Tested tool with industrial project
Zdun[11]	Pattern language on distributed object middleware (<10 design patterns)	Example Case Study: Remoting patterns – asynchronous invocation patterns. Evaluation with 4 industrial case studies
Zimmermann et al.[12]	160 SOA decisions	Web services projects
Babar and Capilla[13]	Not indicated	Airborne Mission Systems
Cui et al.[14]	48 potential decisions	Case Study: Commanding Display System
Makki et al.[15]	Not indicated	Case Study: Garage Door Example
Zimmermann et al.[16]	300 SOA design decisions	Case Study from the finance industry
de Boer et al.[17]	11 design decisions	Example: fictional HRM system
Bode and Riebisch[18]	15 architectural patterns	Case Study: Collective Ordering System
Alebrahim et al.[20]	Not indicated	Case Study: Chat application
Kassab et al.[21]	4 architectural styles considered	Not indicated
Dermeval et al.[22]	2 alternatives (MVC vs Layers) for the architecture of a component	Motivating example: BTW system
van Heesch et al.[23]	4 decisions are reported with 7 alternatives in total	3 case studies
Shen et al.[24]	Few alternative decisions for adaptation (2 cases)	Train ticket booking system
Lytra et al.[25]	Design patterns for service-based platform integration (<10)	Case Study from the industry automation domain
Nowak and Pautasso[26]	100 design issues (5 alternatives each)	Focus group
Ameller and Franch[27]	1 design decision	Motivating example: DBMS selection
Lopes Silva et al.[28]	2 architectural styles	Learning Management System
Lytra et al.[29]	12 architectural decisions	Smart city ecosystems case study
Saadatmand and Tahvili[30]	12 feature alternatives	Motivating example: NFR model of a mobile phone
Monteserin et al.[32]	9 architectural tactics	Case study: Battlefield Control System
Carrillo and Capilla [33]	2 sample decision networks (101/70 and 75/50 nodes/relationships respectively. 61 ADDs in total.	Case study in service-based platform integration
Sedaghatbaf and Abdollahi Azgomi [35]	9 decision points considered for the architectural model in total	Case study on a building surveillance (BS) system
Schneider et al. [36]	22 and 9 Pareto-optimal architecture candidates for Case Study 1 and 2 respectively.	Case studies (Business Reporting System, Remote Diagnostic Solution)

Table 5: **Evaluation of selected approaches with respect to RQ3.** For each of the approaches the size and scope of the design space in terms of ADDs as well as the evaluation method used (e.g., case study) are indicated.

References

[1] T. Kishi, N. Noda, T. Katayama, Architectural Design for Evolution by Analyzing Requirements on Quality Attributes, in: 8th Asia-Pacific Software Engineering Conference (APSEC 2001), 4-7 December 2001, Macau, China, 111–118, 2001.

[2] N. S. Rosa, G. R. Ribeiro-Justo, P. R. F. Cunha, A framework for building non-functional software architectures, in: Proceedings of the 2001 ACM Symposium on Applied Computing (SAC), March 11-14, 2001, Las Vegas, NV, USA, 141–147, 2001.

[3] F. Bachmann, L. Bass, M. Klein, Moving from Quality Attribute Requirements to Architectural Decisions, in: ICSE 2003 - Proceedings of 2nd International Software Requirements to Architectures Workshop, STRAW 2003, May 9, 2003, Portland, Oregon, USA, 122–129, 2003.

[4] L. Chung, K. Cooper, A. Yi, Developing adaptable software architectures using design patterns: an NFR approach, Computer Standards & Interfaces 25 (3) (2003) 253–260.

[5] M. Svahnberg, C. Wohlin, L. Lundberg, M. Mattsson, A Quality-Driven Decision-Support Method for Identifying Software Architecture Candidates, International Journal of Software Engineering and Knowledge Engineering 13 (5) (2003) 547–573.

[6] T. Al-Naeem, I. Gorton, M. A. Babar, F. A. Rabhi, B. Benatallah, A quality-driven systematic approach for architecting distributed software applications, in: 27th International Conference on Software Engineering (ICSE 2005), 15-21 May 2005, St. Louis, Missouri, USA, 244–253, 2005.

[7] H. Choi, K. Yeom, Y. Choi, M. Moon, An Approach to Quality Achievement at the Architectural Level: AQUA, in: 8th IFIP WG 6.1 International Conference on Formal Methods for Open Object-Based Distributed Systems, FMOODS'06, Springer-Verlag, Berlin, Heidelberg, 20–32, 2006.

[8] C. Tibermacine, R. Fleurquin, S. Sadou, On-Demand Quality-oriented Assistance in Component-based Software Evolution, in: 9th International Conference on Component-Based Software Engineering, CBSE'06, Springer-Verlag, Berlin, Heidelberg, 294–309, 2006.

[9] M. A. Babar, I. Gorton, A Tool for Managing Software Architecture Knowledge, in: Proceedings of the Second Workshop on SHAring and Reusing architectural Knowledge Architecture, Rationale, and Design Intent, SHARK-ADI'07, IEEE CS, Washington, DC, USA, 11–, 2007.

[10] N. B. Harrison, P. Avgeriou, Leveraging Architecture Patterns to Satisfy Quality Attributes, in: First European Conference on Software Architecture (ECSA), Springer, 263–270, 2007.

[11] U. Zdun, Systematic Pattern Selection Using Pattern Language Grammars and Design Space Analysis, Software: Practice & Experience 37 (9) (2007) 983–1016.

[12] O. Zimmermann, T. Gschwind, J. Küster, F. Leymann, N. Schuster, Reusable Architectural Decision Models for Enterprise Application Development, in: 3rd International Conference on Quality of Software Architectures (QoSA), Springer, 15–32, 2007.

[13] M. A. Babar, R. Capilla, Capturing and Using Quality Attributes Knowledge in Software Architecture Evaluation Process, in: First International Workshop on Managing Requirements Knowledge, IEEE CS, 53–62, 2008.

[14] X. Cui, Y. Sun, H. Mei, Towards Automated Solution Synthesis and Rationale Capture in Decision-Centric Architecture Design, in: Proceedings of the Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA), IEEE CS, Washington, DC, USA, 221–230, 2008.

[15] M. Makki, E. Bagheri, A. A. Ghorbani, Automating Architecture Trade-Off Decision Making through a Complex Multi-attribute Decision Process, in: 2nd European Conference on Software Architecture (ECSA), Lecture Notes in Computer Science, Springer, 264–272, 2008.

[16] O. Zimmermann, U. Zdun, T. Gschwind, F. Leymann, Combining Pattern Languages and Reusable Architectural Decision Models into a Comprehensive and Comprehensible Design Method, in: Proceedings 7th Work. IEEE/IFIP Conference Software Architecture, IEEE, 157–166, 2008.

[17] R. de Boer, P. Lago, A. Telea, H. Van Vliet, Ontology-driven visualization of architectural design decisions, in: Software Architecture, European Conference on Software Architecture. WICSA/ECSA 2009. Joint Working IEEE/IFIP Conference on, Springer-Verlag, Berlin, Heidelberg, 51–60, 2009.

[18] S. Bode, M. Riebisch, Impact Evaluation for Quality-Oriented Architectural Decisions regarding Evolvability, in: Software Architecture, 4th European Conference, ECSA 2010, Copenhagen, Denmark, August 23-26, 2010. Proceedings, 182–197, 2010.

[19] B. Xu, Z. Huang, O. Wei, Making Architectural Decisions Based on Requirements: Analysis and Combination of Risk-Based and Quality Attribute-Based Methods, Ubiquitous, Autonomic and Trusted Computing, Symposia and Workshops on (2010) 392–397.

[20] A. Alebrahim, D. Hatebur, M. Heisel, A Method to Derive Software Architectures from Quality Requirements, in: 18th Asia Pacific Software Engineering Conference, APSEC 2011, Ho Chi Minh, Vietnam, December 5-8, 2011, 322–330, 2011.

[21] M. Kassab, G. El-Boussaidi, H. Mili, A Quantitative Evaluation of the Impact of Architectural Patterns on Quality Requirements, in: Software Engineering Research, Management and Applications 2011 [selected papers from the 9th International Conference on Software Engineering Research, Management and Applications, SERA 2011, Baltimore, MD, USA, August 10-12, 2011]., 173–184, 2011.

[22] D. Dermeval, J. Pimentel, C. Silva, J. Castro, E. Santos, G. Guedes, M. Lucena, A. Finkelstein, STREAM-ADD - Supporting the Documentation of Architectural Design Decisions in an Architecture Derivation Process, in: Proceedings of the 2012 IEEE 36th Annual Computer Software and Applications Conference, COMPSAC'12, IEEE CS, Washington, DC, USA, 602–611, 2012.

[23] U. van Heesch, P. Avgeriou, R. Hilliard, Forces on Architecture Decisions – A Viewpoint, in: Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, WICSA/ECSA, IEEE CS, 101–110, 2012.

[24] L. Shen, X. Peng, W. Zhao, Quality-Driven Self-Adaptation: Bridging the Gap between Requirements and Runtime Architecture by Design Decision, in: 36th Annual IEEE Computer Software and Applications Conference, COMPSAC 2012, Izmir, Turkey, July 16-20, 2012, 185–194, 2012.

[25] I. Lytra, H. Tran, U. Zdun, Supporting Consistency Between Architectural Design Decisions and Component Models Through Reusable Architectural Knowledge Transformations, in: Proceedings of the 7th European Conference on Software Architecture (ECSA), ECSA'13, Springer-Verlag, Berlin, Heidelberg, 224–239, 2013.

[26] M. Nowak, C. Pautasso, Team Situational Awareness and Architectural Decision Making with the Software Architecture Warehouse, in: Proceedings of the 7th European Conference on Software Architecture, ECSA'13, Springer-Verlag, Berlin, Heidelberg, 146–161, 2013.

[27] D. Ameller, X. Franch, Assisting software architects in architectural decision-making using Quark, CLEI electronic journal 17 (3) (2014) 1–20.

[28] I. C. Lopes Silva, P. H. S. Brito, B. F. dos S. Neto, E. Costa, A. A. Silva, A Decision-making Tool to Support Architectural Designs Based on Quality Attributes, in: Proceedings of the 30th Annual ACM Symposium on Applied Computing, SAC'15, ACM, New York, NY, USA, ISBN 978-1-4503-3196-8, 1457–1463, 2015.

[29] I. Lytra, G. Engelbrecht, D. Schall, U. Zdun, Reusable Architectural Decision Models for Quality-driven Decision Support: A Case Study from a Smart Cities Software Ecosystem, in: Proceedings of the Third International Workshop on Software Engineering for Systems-of-Systems, SESoS'15, IEEE Press, Piscataway, NJ, USA, 37–43, 2015.

[30] M. Saadatmand, S. Tahvili, A Fuzzy Decision Support Approach for Model-Based Tradeoff Analysis of Non-Functional Requirements, in: 12th International Conference on Information Technology : New Generations, IEEE, ISBN 978-1-4799-8827-3, 112–121, 2015.

[31] G. Me, C. Calero Munoz, P. Lago, Architectural patterns and quality attributes interaction, IEEE, doi:\bibinfo{doi}{10.1109/QRASA.2016.10}, 2016.

[32] A. Monteserin, J. A. D. Pace, I. Gatti, S. N. Schiaffino, Agent Negotiation Techniques for Improving Quality-Attribute Architectural Tradeoffs, in: Advances in Practical Applications of Cyber-Physical Multi-Agent Systems: The PAAMS Collection - 15th International Conference, PAAMS 2017, Porto, Portugal, June 21-23, 2017, Proceedings, 183–195, doi:\bibinfo{doi}{10.1007/978-3-319-59930-4_15}, URL https://doi.org/10.1007/978-3-319-59930-4_15, 2017.

[33] C. Carrillo, R. Capilla, Ripple effect to evaluate the impact of changes in architectural design decisions, in: Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings, ECSA 2018, Madrid, Spain, September 24-28, 2018, 41:1–41:8, doi:\bibinfo{doi}{10.1145/3241403.3241446}, URL https://doi.org/10.1145/3241403.3241446, 2018.

[34] S. Malakuti, T. Goldschmidt, H. Koziolek, A Catalogue of Architectural Decisions for Designing IIoT Systems, in: Software Architecture - 12th European Conference on Software Architecture, ECSA 2018, Madrid, Spain, September 24-28, 2018, Proceedings, 103–111, doi:\bibinfo{doi}{10.1007/978-3-030-00761-4_7}, URL https://doi.org/10.1007/978-3-030-00761-4_7, 2018.

[35] A. Sedaghatbaf, M. Abdollahi Azgomi, SQME: A Framework for Modelling and Evaluation of Software Architecture Quality Attributes, Software and Systems Modeling (2018) 1–24doi:\bibinfo{doi}{10.1007/s10270-018-0684-3}.

[36] Y. Schneider, A. Busch, A. Koziolek, Using Informal Knowledge for Improving Software Quality Trade-Off Decisions, in: Software Architecture - 12th European Conference on Software Architecture, ECSA 2018, Madrid, Spain, September 24-28, 2018, Proceedings, 265–283, doi:\bibinfo{doi}{10.1007/978-3-030-00761-4_18}, URL https://doi.org/10.1007/978-3-030-00761-4_18, 2018.