Data extraction

ID	TITLE	AUTHORS	YEAR	APPROACH	ARTIFACT	METHOD	DESCRIPTION	PROCESS	TOOL	NFR	SYSTEM TYPE	FOCUS	CHALLENGES
S1	A Model-Driven Approach to Requirements Engineering in Ubiquitous Systems	Ruiz-Lopez, Tomas and Rodriguez-Dominguez, Carlos and Noguera, Manueland Jose Rodriguez, Maria	2012	MD- UBI	Method	REUBI	REUBI is a method for specifying the system's requirements under development, paying particular attention to the treatment of Non-Functional Requirements, taking into account the specific characteristics of the Ubiquitous Computing and Ambient Intelligence fields.	Yes	No	General	Ubiquitous System	System in development	Systematic evaluation
S2	Applying model-driven engineering to a method for systematic treatment of NFRs in AmI systems	Ruiz-Lopez, Tomas and Rodriguez-Dominguez, Carlos and Noguera, Manueland Jose Rodriguez, Maria and Benghazi, Kawtar and Luis Garrido, Jose	2013	MD- UBI	Process	None	The authors propose a model-driven approach for the development of high-quality Ubiquitous Systems. This approach is designed as a complete development methodology, covering the entire software development life cycle. In the end, the approach provides four deliverables: The Requirements Analysis, an application design document, the system implementation, and the verification and validation report.	Yes	No	General	Ubiquitous System	System in development anf Final product	Context-awareness, adaptivity, heterogeneity
S3	Evaluating energy efficiency of Internet of Things software architecture based on reusable software components	Kim, Doohwan and Choi, Jae-Young and Hong, Jang-Eui	2017	Architecture-based energy evaluation	Approach	None	This paper proposes an architecture for IoT solution development focusing on better energy consumption, evaluates the architecture by software measurement, comparing it with another system that does not use it.	No	No	Energy efficiency	IoT Application	System in development	Security, performance power, interoperability
S4	Heuristics to Evaluate the Usability of Ubiquitous Systems	Rocha, Larissa C. and Andrade, Rossana M. C. and Sampaio, Andreia L. andLelli, Valeria	2017	None	Tool	None	This paper proposes usability evaluation of ubiquitous systems by heuristic inspection. Each heuristic has one or more quality characteristics for ubiquitous systems associated with it.	No	HUbis	Acceptability, Adaptation, Attention, Availability, Calmness, Context-awareness, Device Capability, Ease of use, Effectiveness, Efficiency, Familiarity, Interconnectivity, Mobility, Network Capability, Predictability, Privacy, Reliability, Robustness, Safety, Scalability Security, Simplicity, Transparency, Trust, Usability, User Satisfaction, Utility, Data input, Flexibility, Information display, Positioning of components	Ubiquitous System	System implemented	Context-awareness, transparency, attention, calmness, mobility
S5	Infrastructure for ubiquitous computing: Improving quality with modularisation	Munnelly, J. and Clarke, S.	2008	None	Method	GQM	The authors identify the major challenges for improving ubiquitous systems through modularization and use GQM (Goal Question Metric) to guide the evaluation of systems post improvement process.	No	No	Comprehensibility, Maneageability, Maintenability, Reusability, Scalability, Testability, and Usability	Ubiquitous System	System in development	Connectivity, adaptivity
S6	Quantification of the quality characteristics for the calculation of software reliability	Jazdi, N. and Oppenlaender, N. and Weyrich, M.	2016	Determination of the software reliability approach	Approach	None	The authors propose a method that uses fuzzy logic and neural networks to estimate software reliability.	No	None	Reliability	IoT Application	System in development	Lack of methods to evaluate reliability
S7	REUBI: A Requirements Engineering method for ubiquitous systems	Ruiz-López, T. and Noguera, M. and Rodríguez, M.J. and Garrido, J.L. and Chung, L.	2013	None	Method	REUBI	REUBI is a method for specifying the system's requirements under development, paying particular attention to the treatment of Non-Functional Requirements, taking into account the specifics characteristics of the Ubiquitous Computing and Ambient Intelligence fields.	No	None	General	Ubiquitous System	System in development	Context-awareness, dynamicity, heterogeneity, adaptivity, personalization
S8	Snap4City: A scalable IOT/IOE platform for developing smart city applications	Badii, C. and Belay, E.G. and Bellini, P. and Cenni, D. and Marazzini, M. and Mesiti, M. and Nesi, P. and Pantaleo, G. and Paolucci, M. and Valtolina, S. and Soderi, M. and Zaza, I. and Hachem, F.	2018	Performance evaluation	Approach	No	The authors propose a platform for developing smart city applications, focusing on practicality and communication performance, and scalability. To prove the platform's applicability, the authors conduct a performance evaluation on applications developed from the platform.	No	None	Performance and Scalability	IoT Application	System in development	Context-awareness
S9	Structural and behavioral reference model for IoT-based elderly healthcare systems in smart home	Ghasemi, Farideh and Rezaee, Ali and Rahmani, Amir Masoud	2019	ATAM scenario-based approach	Approach	No	The authors proposed an architecture evaluation based on the Architecture Tradeoff Analysis Method (ATAM) scenario-based approach. ATAM is a method for analysis and evaluation of software systems architecture.	No	None	Availability , Performance, Modifiability. Interoperability, Security and Usability.	IoT Application	System in development	Availability, performance, security, interoperability, modifiability
S10	Using Reference Architectures for Design and Evaluation of Web of Things Systems: A Case of Smart Homes Domain	Chauhan, M.A. and Babar, M.A.	2017	Reference Architectures for Design and Evaluation of Web of Things Systems	Approach	No	They propose a process for defining the architecture for IoT subsystems. This process contains an evaluation step, where previously defined quality attributes are evaluated using questionnaires. As a result of the evaluation, a report shows commonly unmet quality attributes, risky design decisions, sensitivity and benchmarks, and suggestions for improvement in IoT subsystem architectures.	No	None	Security, Availability, Scalability, Elasticity, Reliability, Multi-tenancy and, Interoperability	IoT Application	System in development	Security, availability, scalability, reliability, interoperability
S11	Comparing Heuristic Evaluation and MALTU Model in Interaction Evaluation of Ubiquitous Systems	José Cezar de Souza Filho, Marcos Randel Freitas Brito, Andréia Libório Sampaio	2020	Heuristic evaluation and MALTU model	Approach	No	It proposes usability evaluation of ubiquitous systems through heuristic inspection. Each heuristic has one or more quality characteristics for ubiquitous systems associated with it. In this study, a comparison is made between HUBIS and MALTU, a model for interaction evaluation in ubiquitous systems based on textual analysis, using texts about the system posted by users on social networks.	No	HUbis	Acceptability, Adaptation, Attention, Availability, Calmness, Context-awareness, Device Capability, Ease of use, Effectiveness, Efficiency, Familiarity, Interconnectivity, Mobility, Network Capability, Predictability, Privacy, Reliability, Robustness, Safety, Scalability Security, Simplicity, Transparency, Trust, Usability User Satisfaction, Utility, Adaptability. Data input, Flexibility, Information display, Positioning of components	Ubiquitous System	System implemented	Transparency, attention, calmness, mobility, context-awareness
S12	RC-ASEF: An Open-Source Tool-Supported Requirements Elicitation Framework for Context-Aware Systems Development	Unai Alegre-Ibarra, Juan Carlos Augusto, Carl Evans	2018	No	Tool	Reubi and NFR Framewor	The authors use the NFR Framework and REUBI for the requirements analysis of context-aware applications.	No	RC-ASEF	General	Ubiquitous System	System in development	Context-awareness

Approaches, methods, process and tools

ID	APPROACH	METHOD	PROCESS	TOOL
S1	-	REUBI	MD- UBI	-
S2	-	-	MD- UBI	-
S3	Architecture-based energy evaluation.	-	-	-
S4	Heuristic evaluation	-	-	HUbis
S5	-	GQM	-	-
S6	Determination of the software reliability approach	-	-	-
S7	-	REUBI	-	-
S8	Performance evaluation	-	-	-
S9	ATAM scenario-based approach	-	-	-
S10	-	-	-	-
S11	Heuristic evaluation and MALTU model	-	-	HUbis
S12	-	Reubi and NFR Framework	-	RC-ASEF
	6	**3**	**1**	**2**

RNFs Identified

Quality characteristics	Definition	Cited by	ISO 25000	Carvalho (2017)
Acceptability	Represents the desire to use an application and its utilization rates	S4, S11	0	1
Adaptation	System ability to perceive the environment and adapt the information display accordingly.	S4, S11	1	1
Attention	The ability to keep the user's attention to her/his main activity and not on the system and the technology involved	S4, S11	0	1
Availability	The service is always available, regardless of hardware, software or user fault, and it is often taken for granted until downtime occurs	S4, S9, S10, S11	1	1
Calmness	The ability to prevent users from feeling overwhelmed by information system	S4, S11	0	1
Context-awareness	The ability to perceive contextual information system and proactively adapt its functionality	S4, S11	0	1
Comprehensibility	Ease with which application developers understand the software.	S5	0	0
Data Input	Verify that the different ways of providing data entry to the system are pleasing to the user.	S4, S11	0	0
Device Capability	Properties of the device where the application will run (e.g., screen size, color depth, battery life)	S4, S11	0	1
Ease of use	The system should be easy to use by a target user group	S4, S11	1	1
Effectiveness	It refers to completeness in performing tasks proactively adapt its functionality	S4, S11	1	1
Efficiency	It refers to the amount of effort and resources required to reach a certain goal in the system	S4, S11	1	1
Energy efficiency	P17	S3	0	0
Familiarity	User interactions with the system should improve the quality of her/his work. The user should be treated with respect. The design should be aesthetically pleasing.	S4, S11	0	1
Flexibility	The application should give the users the ability to customize configurations according to their needs and experiences.	S4, S11	0	0
Information display	The way the information is presented on the screen is consistent with what is expected.	S4, S11	0	0
Interconnectivity	An interconnected network between devices.	S4, S11	0	1
Interoperability	Degree which the system components are coordinated and compatible in relation with each other.	S9, S10	1	0
Maintenability	Maintainability is the ease with which software components can be changed and updated with least disruption to the software system.	S5	1	0
Maneageability	Manageability is the ease with which software components can be developed in isolation and later composed.	S5	0	0
Mobility	The ability to provide users with continuous access to resources and information system, regardless of its location within the limits of the systems	S4, S11	0	1
Modifiability	The system's ability to withstand changes	S9	1	0
Network Capability	Represents the collection of network information (e.g., signal strength, delay, jitter)	S4, S11	0	1
Performance	Refers to this fact that how long it takes to respond to an event.	S8, S9	1	0
Positioning of components	The elements in the interface have their positioning to please the users.	S4, S11	0	1
Predictability	The ability, from past experiences, to predict the result of the system.	S4, S11	0	1
Privacy	The ability to maintain information and data protected.	S4, S11	0	1
Reliability	The ability to maintain a particular level of performance when used under specific software conditions	S4, S6, S10, S11	1	1
Reusability	Reusability is the extent to which software components can be used again in other systems.	S5	1	0
Robustness	Degree to which a system or component can execute correctly in the presence of invalid inputs or stressful environmental conditions.	S4, S11	0	1
Safety	The level of risk of harm to people, business, software, hardware, property or the environment in a specified context of use.	S4, S11	0	1
Scalability	The ability to provide services to a few or a large number of users.	S4, S5, S8, S10, S11	0	1
Security	The protection to transport and to store information and also security controls who can access, use and modify context information.	S4, S9, S10, S11	1	1
Simplicity	The user interface and the instructions should be simple.	S4, S11	0	1
Testability	Testability is the ease with which checks can be carried out on various aspects of the software.	S5	1	0
Transparency	The ability to hide the system, so users may not be aware of it. Moreover, the interaction is performed through natural interfaces.	S4, S11	0	1
Trust	It is the belief of the user that the system uses data properly and not cause any harm. Implies awareness, privacy and control.	S4, S11	0	1
Usability	The ability of the software to be understood, learned, used and attractive to the user, when used under specified conditions.	S4, S5, S9, S11	1	1
User Satisfaction	The degree of user satisfaction and how the system is attractive for the user	S4, S11	0	1
Utility	The ability to provide value to user. The system provides a contribution to user that was not available before its development.	S4, S11	0	1
Elasticity	The ability of the system to provide particular service on demand during a time interval.	S10	0	0
Multi-tenancy	Multi-tenancy is the capacity of the system to provide isolation among the data and services belonging to different tenant (user groups).	S10	0	0