OpenReq

Grant Agreement n°	732463
Project Acronym:	OpenReq
Project Title:	Intelligent Recommendation Decision Technologies for Community-Driven Requirements Engineering
Call identifier:	H2020-ICT-2016-1
Instrument:	RIA (Research and Innovation Action
Торіс	ICT-10-16 Software Technologies
Start date of project	January 1 st , 2017
Duration	36 months

D1.2 Requirements analysis & design document

Lead contractor:	UPC
Author(s):	UPC, TUGraz
Submission date:	December 2017
Dissemination level:	PU



Project co-funded by the European Commission under the H2020 Programme.



Abstract: This deliverable describes the requirements and the architecture for the OpenReq project. The requirements were identified from the trial providers applying a rigorous protocol that included visits to the trial providers' sites. The requirements were consolidated and afterwards, they were documented in the project management tool, where they will evolve as the project proceeds.

The platform architecture documentation describes the system components and their relationships which are offered in the form of MicroServices.



This document by the OpenReq project is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 Unported License.

This document has been produced in the context of the OpenReq Project. The OpenReq project is part of the European Community's h2020 Programme and is as such funded by the European Commission. All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability is respect of this document, which is merely representing the authors view.



Table of Contents

1 INT	RODUCTION13
2 RE0	QUIREMENT ANALYSIS14
2.1	Methodology and technological support
2.1.1	Steps, schedule and roles
2.1.2	Instruments for requirements elicitation
2.1.3	Instruments for requirements documentation
2.2	Trial on-site visits
2.2.1	Cross-platform OSS trial: Qt25
2.2.2	Transportation trial: Siemens26
2.2.3	Telecom trial: WindTre27
2.2.4	Vogella potential trial29
2.3	Consolidation meetings 30
2.4	Requirements 30
2.4.1	Key findings regarding WP231
2.4.2	Key findings regarding WP331
2.4.3	Key findings regarding WP431
2.4.4	Key findings regarding WP532
2.4.5	Key findings regarding WP632
3 AR(CHITECTURAL DESCRIPTION OF OPENREQ34
3.1	OpenReq architecture
3.1.1	OpenReq knowledge infrastructure
3.1.2	OpenReq MicroServices
3.1.3	OpenReq prototype
3.1.4	Different stakeholders applications
3.2	OpenReq MicroServices 40
3.3	Catalogues of OpenReq MicroServices 41
3.4	Port Restrictions of OpenReq MicroServices
3.5	Naming Conventions of OpenReq MicroServices 42
REFE	RENCES43
ANNE REC	X A. ROLES AND RESPONSIBILITIES IN QUIREMENTS' ANALYSIS44



ANNEX B. DATA COLLECTION IN PRE- QUESTIONNAIRES TO TRIAL SITES	46
ANNEX C. DATA COLLECTION IN INTERVIEWS AT TRIAL SITES	50
ANNEX D. ATTRIBUTES AND SELECTION OF INSTRUMENTS FOR REQUIREMENTS' DOCUMENTATION	65
ANNEX E. EPICS DETAILED LIST	68
ANNEX F. USER STORIES DETAILED LIST	85
ANNEX G. DOMAIN MODELS	.112
ANNEX H. GLOSSARY	.118



List of Figures

Figure 1: Steps and schedule	14
Figure 2: Requirement elicitation instruments	16
Figure 3: Agenda of each trial partner visit	25
Figure 4: UML Component Diagram of the OpenReq overall architecture.	34
Figure 5: A specialization of OpenReq Ontology that is tailored towards the release plans scenario using the OpenReq prototype	ning 36
Figure 6: Examples of explicit and implicit feedback	38
Figure 7: Example: trend of different app review types	38
Figure 8: Complete call-chain of an user interaction with the OpenReq prototype	39



List of Tables

Table A.1. Roles developed by the different partners in each trial
Table A.2. Roles' responsibilities for each activity in Task preparation
Table A-3: Partners' responsibility for each activity
Table B.1. Information about the respondent of the general organization information part46
Table B.2. Information about the organization
Table B.3. Information about data, data privacy and data access 46
Table B.4. Information about strategic goals
Table B.5. Information about requirements engineering process 47
What are the methods, standards, and tools used for, and the people responsible for:47
Table B.6. Expectations on OpenReq47
Table B.7. Information about the respondent of the project-specific information part
Table B.8. Information about the selected project (a project could be the development of a whole product or just a part of it)
Table B.9. Information about the selected product
Table B.10. Information about methods, standards, tools, and responsible people on the project
Table B.11. Information about the templates used and documentation about the project49
Table C.ASIS-1. Information about the role of the requirements manager and the decision-maker or business expert in the organization
Table C.ASIS-2. Information about the way to work in the organization related with requirements engineering activities 50
Table C.ASIS-3. Information about the strategic goals of the organizations 52
Table C.ASIS-4. Information about the strategic goals of the project selected for the trial52
Table C.ASIS-5. Strategic goals of the organization and the project selected for the trial53
Table C.ASIS-6. Expectations for OpenReq results
Table C.ASIS-7. Other informations that can be relevant for OpenReq 53
Table C.WP2-1. Information about the role of the requirements engineer and the product manager in the organization
Table C.WP2-2. Information about data analytics
Table C.WP2-3. Information about explicit data collection 55
Table C.WP2-4. Information about implicit data collection
Table C.WP2-5. Information about visualization and integration 56
Table C.WP2-6. Expectations for OpenReq results
Table C.WP2-7. Other information that can be relevant for OpenReq



Table C.WP3-1. Information about the role of the requirements manager or analyst	57
Table C.WP3-2. Information about the requirements reuse and recommendations in the tripping project	ial 57
Table C.WP3-3. Expectations for OpenReq results	58
Table C.WP3-4. Other information that can be relevant for OpenReq	58
Table C.WP4-1. Information about the role of the requirements manager or analyst	59
Table C.WP4-2. Information about data and people that can be consulted for knowing how the organization take decisions in their trial projects	he 59
Table C.WP4-3. Information about the use of e-participation platforms in RE activities in the organizations	he 59
Table C.WP4-4. Expectations for OpenReq results	59
Table C.WP4-5. Other information that can be relevant for OpenReq	60
Table C.WP5-1. Information about the role of the requirements engineer and product manage	er; 61
Table C.WP5-2. Information about current situation of knowledge and dependend management in the trial project	су 61
Table C.WP5-3. Expectations for OpenReq results	62
Table C.WP5-4. Other information that can be relevant for OpenReq	62
Table C.WP6-1. Information about the role of the requirements engineer and project manage	ger 63
Table C.WP6-2. Information of the tools that are currently used in the trial project	63
Table C.WP6-3. Information about interfaces used, and regulations and standards required e	63
Table C.WP6-4. Other information that can be relevant for OpenReq	64
Table D.1. Attributes of user stories in functional requirements	65
Table D.2. Attributes of epics in functional requirements	65
Table D.3. Decision table of which instrument to use for documenting NFRs	67
Table E.1. Recommend related requirements from same projects	68
Table E.2. Recommend related requirements from same projects	68
Table E.3. Recommend requirement properties	68
Table E.4. Recommend relevant stakeholders	69
Table E.5. Recommend related requirements from previous projects	69
Table E.6. Measure requirements quality	69
Table E.7. Screen business requirements	70
Table E.8. Recommend improvements on the quality of requirement	70
Table E.9. Identify requirements	71



Table E.10. user	Recommend requirements and requirements metadata based on the context of the
Table E.11.	Dashboard for requirements analytics
Table E.12.	Analyze existing requirements from Requirement Management System
Table E.13.	Cluster user based on their input72
Table E.14.	Integrate with marketing tools72
Table E.15.	Get requirements based on similar products73
Table E.16.	Analyze customer tickets system73
Table E.17.	Analyze interaction data73
Table E.18.	Adhere to requirements engineering practices74
Table E.19.	Store requirements using a predefined ontology74
Table E.20.	Assign requirements to a domain74
Table E.21.	Extract dependencies from textual requirements75
Table E.22.	Comply with the relationships between requirements75
Table E.23.	Take into account relationships of the properties of requirements76
Table E.24.	Comply with the relationships affected by confidential requirements76
Table E.25.	Establish external relationships
Table E.26.	Ensure adherence to relationships over different versions
Table E.27.	Feedback on recommendations
Table E.28.	Recommend similar requirements
Table E.29.	Manage requirements metadata in OpenReq prototype78
Table E.30.	Resolve the inconsistencies in the compliance ratings79
Table E.31.	Resolve inconsistencies of solution approaches
Table E.32.	Manage user account of OpenReq prototype
Table E.33.	Support user of OpenReq prototype
Table E.34.	Manage project of OpenReq prototype
Table E.35.	Manage project release in OpenReq prototype
Table E.36.	Manage project requirements in OpenReq prototype
Table E.37.	Preference information of other OpenReq prototype users
Table E.38.	Recommend solution for inconsistencies of group user preferences
Table E.39.	Visualize an interactive release plan
Table E.40.	Analyze explicit feedback from development tools
Table E.41.	Adhere to external relationships
Table E.42.	Identify useful user feedback



Table E.43. Identify similar feedback	.83
Table E.44. Identify user rationale	.83
Table E.45. Repair conflicts in relationships	.84
Table F.1. Extract context data from Bugzilla	.85
Table F.2. See the data in a table	.85
Table F.3. See data in charts	.85
Table F.4. Be able to connect different issue tracking systems with openReq	.85
Table F.5. Be notified when the priority of a task is changed while implementing it	.86
Table F.6. Be notified when new requirements with a predefined priority are created	.86
Table F.7. Integrate openReq in Eclipse with Mylyn	.86
Table F.8. See openReq in Web UI	.87
Table F.9. Click on the "Last project's trend" button	.87
Table F.10. Send direct invitation via e-mail instead of the poll	.87
Table F.11. Send invitations to predefined user groups	.87
Table F.12. Allow comments	.88
Table F.13. Know when two requirements are talking about the same	.88
Table F.14. Know when two requirements are talking about the same	.88
Table F.15. Identify the bug reports related to the comment I am entering	.89
Table F.16. Be informed of the information missing in a bug	.89
Table F.17. Be recommended of the possible category of the requirement I am creating	.89
Table F.18. Be recommended of the possible component of the requirement I am creating	.90
Table F.19. Be recommended of the possible environment of the requirement I am creat	ing .90
Table F.20. Be asked questions related to the priority of the requirement I am creating	.90
Table F.21. Know the explanation of the priority assigned to a requirement created	.90
Table F.22. Automatically assign an assignee of a requirement that has been created	.91
Table F.23. Automatically differentiate the requirements contained in the same paragraph	.91
Table F.24. Identify if there was problems with a similar requirements to the one I am deal with at the moment	ing .92
Table F.25. Be recommended of the possible risk of the requirement I am in charge of	.92
Table F.26. Tell the system that a recommendation is wrong	.92
Table F.27. Extract business requirements from user requests	.93
Table F.28. Identify bad quality requirements	.93
Table F.29. Identify outdated requirements	.93



Table F.30. Be recommended of the possible classification of the requirements creat	ed93
Table F.31. Extract business requirements from user requests from tweets and Faceb	ook posts 94
Table F.32. Recommend prioritization for requirements identified from the social using the information in the social media to do so	media by 94
Table F.33. Visualize a summary of what users are commenting on social media	
Table F.34. View external information about requirements	
Table F.35. Have reports on social media data	
Table F.36. Get feedback from the system whether some piece of text is a requireme	ent or not
Table F.37. Get feedback from the system which of the entries from a list of requirement or not	luirement
Table F.38. Extract requirements candidates from a Microsoft word document	
Table F.39. Get a recommendation for an assignment of stakeholders roles (doma requirement	ains) to a
Table F.40. Get a recommendation which requirements are similar to a reference req	juirement 97
Table F.41. Get a recommendation regarding redundant requirements	
Table F.42. Get a recommendation regarding a technical solution approaches to requirement	o fulfil a 97
Table F.43. Customize "semantic thesaurus" for the domain	
Table F.44. Add a "semantic thesaurus" for the domain	
Table F.45. Rate a requirement as compliant, conditionally compliant or none compl	liant 98
Table F.46. Get a recommendation regarding how to best resolve the inconsisted compliance ratings	encies of 98
Table F.47. Create an user account	
Table F.48. Change the password	
Table F.49. Reset the password	
Table F.50. Upload a profile image	
Table F.51. Give feedback to the OpenReq community	
Table F.52. Update the user profile	
Table F.53. Get help in using the system the first time	
Table F.54. Create a project	
Table F.55. Add members to a project	
Table F.56. Modify project details	
Table F.57. Insert the releases of a project	



Table F.58.	Export project information to pdf101
Table F.59.	Delete a project
Table F.60.	Add requirements to a certain project102
Table F.61.	Add and modify a description of a requirement102
Table F.62.	Search for requirements and sort among releases and/or name102
Table F.63.	Prioritize the requirements of a project103
Table F.64.	Add and modify a metadata of a requirement103
Table F.65.	Add and modify attachments of a requirement103
Table F.66.	Add dependencies between requirements103
Table F.67.	Delete requirements from a project104
Table F.68.	Delete requirements from a project104
Table F.69.	Select the current project
Table F.70.	Get an overview of the current project104
Table F.71.	Get information about open issues105
Table F.72.	Get information about inconsistent preferences with respect to the release plans 105
Table F.73.	Have a chat possibility per requirement
Table F.74.	See the preferences and opinions of other users
Table F.75.	Be notified of the requirements assigned106
Table F.76.	Measure the quality of a requirement
Table F.77. the problem	Tell the system when I am having problems with a requirement and the reason for 106
Table F.78.	Receive lists of prioritised recommendations
Table F.79.	Recognize important requirements for commercial customers107
Table F.80.	Get a recommendation regarding conflicting requirements
Table F.81.	Modify the releases of a project
Table F.82.	Delete the releases of a project
Table F.83.	See ambiguous requirement text108
Table F.84.	See words not included in the project glossary108
Table F.85.	See text expressing uncertainty in requirement specification108
Table F.86.	See the violation to a specific requirement template109
Table F.87.	Delete recommended assignment and/or propose another domain109
Table F.88.	Extract Bugzilla requirement data
Table F.89.	Assess the sentiment of social media109



Table F.90. Obtain information from Gerrit	110
Table F.91. Obtain information from SonarQube	110
Table F.92. Cluster frequent errors of the Eclipse component	110
Table F.93. Classify English Twitter messages	
Table F.94. Classify Italian Twitter messages	111



1 INTRODUCTION

In any software project, the statement of the system requirements and its architecture represents one of the most important tasks to be performed. The system requirements will describe the main functionalities and qualities of the system, while the architecture will describe the main building blocks of the solution and their connections. OpenReq is not an exception to this rule. In this deliverable, we offer two sections that cover both parts, software requirements (**Section 2**) and software architecture (**Section 3**).

Requirements were elicited and consolidated by applying a rigorous protocol that involved all the scientific partners and all the trial providers. Scientific partners designed the instruments to gather the information and trial providers gave all the needed information that was collected through these instruments. This collection took place through some visits at trial providers' sites. In the visits, the appropriate stakeholders were interviewed using a questionnaire. Requirements were documented with the usual agile artefacts, namely epics, user stories and acceptance criteria. Furthermore, for each trial, an ontology was developed to describe in a precise manner all the concepts that they manage.

The OpenReq prototype, as well as the different trial applications, is designed to access the different services provided by the components. In general, the different components have to communicate in a bi-directional fashion with some exceptions such as the dependency service and the social service which just have to provide information to other components. The components will have access to the knowledge infrastructure which is based on the OpenReq Ontology.

Several Annexes at the end of the document provide the details of the different topics addressed in **Section 2**.

- Annex A deals with the roles and responsibilities involved in the process of requirements' analysis
- Annex B details the pre-questionnaires used to collect previous information at the trial sites
- Annex C details the structure of the information collected at the trial sites via face to face interviews
- Annex D presents the attributes and selection criteria of instruments for requirements' documentation
- Annex E contains the detailed description of Epics
- Annex F contains the detailed description of User Stories
- Annex G includes the initial versions of the domain models of the trial partners and also the domain model of OpenReq
- Finally, Annex H is a Glossary of terms



2 REQUIREMENT ANALYSIS

The requirement analysis task T1.2 was organized around the three trials proposed in OpenReq for the Qt, Siemens and WindTre partners together with an additional trial for the Vogella partner.

In this section, we describe the methodology and technological support used for this task, we describe the on-site visits to the trial providers' sites, which were organized with the aim of eliciting requirements for OpenReq and, finally, we describe the requirements obtained.

2.1 Methodology and technological support

In the following we describe the schedule corresponding to task T1.2 development, and the instruments used for requirements elicitation and documentation.

2.1.1 Steps, schedule and roles

The requirement analysis activities were organized in several steps. A general view of the steps and schedule of task T1.2 is shown in Figure 1.



Figure 1: Steps and schedule

- The *Task preparation* step had the goal to produce the artefacts, protocols and tools for requirements elicitation and documentation, and schedule the on-site visits.
- During *RE iteration 1* step, visits to each trial site were conducted in order to obtain the as-is situation, the issues where OpenReq can contribute to, and an initial set of requirements from each trial.
- *RE iteration 2* had the goal to consolidate the elicited requirements, as well as incorporating additional requirements coming from the academic partners and eventually external collaborators. In the schedule, we can see that, first, *RE iteration 1* and *RE iteration 2* were developed for the Qt and the Siemens trial, and afterwards, they were extended to the rest of trials (i.e. Vogella and WindTre trials). The main reason for this was the need to adapt to the availability of the trial providers, although at the end it demonstrated to have a positive effect because it allowed some extra consolidation (at the cost of some additional effort in the task).
- The last three steps had the goal of performing final adjustments and consolidating them into the present deliverable.



For each trial we defined several roles:

- *Requirement provider*: The company that will run the trial.
- *Accompanying partner*: An academic partner that will coordinate with the requirement provider to implement the defined steps in this particular case.
- *WP2-6 leaders*: All the partners that are WP2-6 leaders and are interested in eliciting requirements from the trial for their WP.
- *Coordinator*: The T1.2 task leader (UPC) coordinating the visits to ensure a holistic perspective of the trials when eliciting requirements

Table A-1 in Annex A includes the roles developed by the different partners for each trial.

2.1.1.1 Task preparation

The main activities of the task preparation were: 1) to prepare instruments for requirements elicitation; 2) to prepare requirements documentation artefacts and tools; 3) to schedule onsite visits and the required visit logistics. As a result of the task we obtained the artefacts, protocols and tools for requirements elicitation and documentation. We decided the schedule for on-site visits.

Table A-2 in Annex A lists the roles' responsibilities for each activity in *Task preparation*.

2.1.1.2 RE Iteration 1

The goal of this step was the elicitation of requirements from the trials facilitated by means of on-site visits to each requirement provider.

The main activities were: 1) prepare on-site visits to the requirements providers; 2) conduct the required visits; 3) collect results from these visits.

Table A-3 in Annex A lists the partners' responsibilities for each activity in *RE Iteration 1*.

2.1.1.3 RE Iteration 2

The main activities of this step were:

- Analysis of requirements resulting from *RE Iteration 1* to detect redundant requirements, conflicting requirements and missing requirements.
- Negotiation to resolve conflicts.
- Evolve requirements according to analysis and negotiation results.

Table A-3 in Annex A lists the partners' responsibilities for each activity in *RE Iteration 2*.

2.1.1.4 Final adjustments

The goal of this last step was to obtain a set of requirements without redundancies, prioritized and all at the same level of abstraction. To achieve so, a face-to-face meeting was coordinated with all scientific partners.

2.1.2 Instruments for requirements elicitation

In this section, we introduce the instruments used during the trial visits. They are summarized in Figure 2 and described in individual subsections.

• One of the instruments used was a pre-questionnaire that allowed knowing better the organizations of the trials partners and the project selected for the trial in each organization. The pre-questionnaire was sent to the trial partners before the visits, and the partners



returned them answered to the technical partners. Through the pre-questionnaires, before the visits, the people from the technical working packages obtained useful knowledge about the organization and about the project selected to be a trial for OpenReq.

• The other instruments were interview guides used during the visits to each trial partner. The interview guides had as a goal to obtain information about the situation of the project selected for the trial and to elicit the requirements of OpenReq corresponding to each OpenReq work package (i.e., WP2, WP3, WP4, WP5 and WP6). Although the trial partners could decide to select one or more projects for the trials, finally all of them selected only one project. In order to prepare each interview guide, an interview template was prepared. The template was used by each WP2-6 leader partner to prepare an interview script document, and this is the reason that the different interviews follow a similar schema.



Figure 2: Requirement elicitation instruments

2.1.2.1 Pre-questionnaire

The pre-questionnaire had two parts. The first part consisted in questions about the general organization context. The second part consisted in questions about the characteristics of the project selected for the trials of OpenReq by each organization. We include below the description of each part of the questionnaire.

General organization information

The respondent of this part of the pre-questionnaire was the person that acts as contact of the requirement provider partner and, in case s/he has not the required information available, s/he designated an adequate alternative respondent.

The goals of this part of the pre-questionnaire were to obtain general knowledge about the organization, its strategic goals, how requirements engineering activities are currently performed in the organization, and the main expectations from OpenReq. The information required from the organization was:



- Demographic information about the respondent (see Table B.1 in Annex B), including his role in the project selected by the organization as trial of OpenReq.
- Organization information (see Table B.2 in Annex B). Besides several predefined aspects, the pre-questionnaire allowed the respondent to add any information s/he deemed of interest.
- Data, data privacy and data access information (see Table B.3 in Annex B). These questions were included to know under which conditions research partners could access organization data of interest for OpenReq, which are the data sources, and how data is structured.
- Strategic goals information, both about the organization and the project selected for the trial (see Table B.4 in Annex B).
- Requirements engineering process information (see Table B.5 in Annex B). These questions were included to know the methods, standards, and tools used in the different activities of requirements engineering of the organizations and to know the roles of people involved in each activity.
- Expectations about OpenReq (see Table B.6 in Annex B). The last section of the first part of the questionnaire was included to know how the organizations think that the results of OpenReq are going to support their requirements engineering activities, how the organization processes are going to change, and the constraints and limits of OpenReq results according to the organization.

Project-specific information

The goal of this part of the pre-questionnaire was to obtain general knowledge about the project selected as trial and how requirements engineering activities are currently performed in the project. The respondent was the requirements manager and/or an analyst of the project selected as trial for OpenReq.

The information required from the project selected as trial was:

- Demographic information about the respondent (see Table B.7 in Annex B).
- Project information (see Table B.8 in Annex B) both about the team and the type of project.
- Product information (see Table B.9 in Annex B). Including main functionality, domain, product type (e.g., proprietary or open source) and remarkably number of requirements and type of requirements used (e.g., user stories, epics, use cases, etc.)
- Project method, standard, tools, and responsible people information (see Table B.10 in Annex B). These aspects are required in relation to the different activities of the product development and specifically in each activity of the requirements engineering stage.
- Project templates and documentation information (see Table B.11 in Annex B). These questions were included in order to know the sources of information that the organization had about the selected project. Specifically, we asked if the organization maintained any documentation describing the processes, methods, and standards used. In addition, we asked if the organization has/uses standard templates for documenting aspects for project plans, functional requirements or non-functional requirements. It was also asked whether the templates and documentation are supported by a tool and, finally, which metadata is used for requirements

2.1.2.2 Interview Guides



Six interview guides were developed. The first one had as goal to obtain information about the current situation of the project selected for the trial. The other five had as goal to elicit the requirements of OpenReq corresponding to OpenReq work packages (i.e., WP2, WP3, WP4, WP5 and WP6). The six interviews are described in the following subsections.

As-is interview

There were two stakeholder roles among the expected interviewees, namely the requirements manager and the decision-maker or business expert.

This interview was designed to take 60 minutes.

The information required is described next. As it can be seen, some questions may seem repetitive with respect to the pre-questionnaire. The idea was to obtain more detailed information about the different issues exploiting face-to-face communication, and to obtain additional clarifications when needed.

- Information about the role of the requirements manager and the decision-maker or business expert in the organization (see Table C.ASIS-1 in Annex C).
- Information about tools or processes used in project selected for the trial in requirements engineering activities (see Table C.ASIS-2 in Annex C). Some examples of questions in this block are: the approach that the organization uses (e.g., waterfall, agile, scrum, etc.); how are requirements elicited in the context of the selected project; which methods, techniques and tools are used in the selected project for requirements management; and which are the challenges regarding requirements management that the people involved in the project have.
- Information about the strategic goals of the organization (see Table C.ASIS-3 in Annex C). These questions allow knowing how the organization measures the achievement of its strategic goals.
- Information about the strategic goals of the project selected for the trial (see Table C.ASIS-4 in Annex C). These questions allow knowing how the organization measures the achievement of the strategic goals of the selected project. In this section it was also included a question to know if the stakeholders think that the requirements engineering activities influence the achievement of the strategic goals of the project.
- Once the strategic goals of the organization and the selected project were known, the questions in the following block address the relationship among the goals (see Table C.ASIS-5 in Annex C). This allows knowing the influence of the success of the project on the success of the organization goals.
- Expectation for the OpenReq results (see Table C.ASIS-6 in Annex C). The stakeholders are required to give their opinion on how OpenReq could improve the requirements engineering process and activities of the project selected as trial. In this block, there are also questions to ask about who could be the users of the OpenReq framework and which strategic goals of the project and the organization could be influenced by OpenReq.
- Finally, the stakeholders attending the meeting were invited to give comments or indicate aspects that may be relevant to OpenReq related with the requirements engineering activities conducted in the selected projects (see Table C.ASIS-7 in Annex C).

WP2 interview: Software Requirements Intelligence



The expected roles for the interviewees were the requirements engineer and the product manager, who utilizes requirements in her work in decision making.

This interview was designed to take 60 minutes.

The information required is described next.

- Information about the role of the requirements engineer and the product manager and business expert (see Table C.WP2-1 in Annex C).
- Information about data analytics (see Table C.WP2-2 in Annex C). In this block, stakeholders were asked whether they use data-analytics, the specific data that they use, and which data could be interesting to be collected and currently is not. Stakeholders were also inquired about the criteria that a data analytic solution should have for requirements engineering to be successful.
- Information about explicit data collection (see Table C.WP2-3 in Annex C). Stakeholders indicated which explicit data are collected, how often, for which purpose, and the accessibility of these data. Legal and privacy aspects that can be relevant for the collection of explicit data in the context of OpenReq were also inquired.
- Information about implicit data collection (see Table C.WP2-4 in Annex C). The same questions included for explicit data (see above) are also included for implicit data.
- Information about visualization and integration of the results of data analysis (see Table C.WP2-5 in Annex C) with special mention to tool support.
- Expectations from the OpenReq results (see Table C.WP2-6 in Annex C). The stakeholders are required to give their opinion on how OpenReq could improve software requirements intelligence in the context of the trial organization.
- Finally, the stakeholders assisting to the meeting are invited to give comments or indicate aspects that may be relevant to OpenReq related with intelligence in requirements engineering activities conducted in the selected project or that are expected as OpenReq results (see Table C.WP2-7 in Annex C).

WP3 interview: Personal Recommendations for Stakeholders

The expected role of the interviewees was that of requirements manager, working on the requirement engineering process of the trial project.

This interview was designed to take 40 minutes.

The information required from the stakeholders is described next.

- Information about the current work of requirements manager or analyst in the trial project and its work in previous projects of the same or other organizations (see Table C.WP3-1 in Annex C).
- Information about requirements reuse and recommendations in the trial project (see Table C.WP3-2 in Annex C). Specifically whether in the trial project there is reuse of requirements and whether a knowledge base of reusable requirements is used. Other information related to reuse (e.g., quality of requirements, risk and effort associated to requirements) was requested. Finally, information about how the relevant stakeholders for the project are identified is required.
- Expectations from the OpenReq results (see Table C.WP3-3 in Annex C). The stakeholders are required to give their opinion on how OpenReq could contribute with



automatic recommendation of requirements to the trial project. The stakeholders were required to indicate which recommendations would be interesting in the context of the trial project (e.g., of reusable requirements, of the quality of requirements defined, of property values, of possible stakeholders) and the roles of people involved in the project that should receive the recommendations. Also, questions about which data sources could be used in order to give recommendations, and which tools should be connected to OpenReq platform to visualize recommendations were added to the interview. Finally, questions about the preferences on the user interface design and the challenges of OpenReq regarding the new features were included.

• Finally, the stakeholders assisting to the meeting were invited to give comments or indicate aspects that may be relevant to OpenReq related with requirements engineering activities conducted in the selected project or that are expected as OpenReq results (see Table C.WP3-4 in Annex C).

WP4 interview: Group Decision Support

The expected roles of the interviewees were the requirements manager and the community expert.

This interview was designed to take 100 minutes.

The information required from the stakeholders is described next.

- Information about the current work of the requirements manager or analyst and the community experts in the trial project and their work in previous projects of the same or other organizations (see Table C.WP4-1 in Annex C).
- Information about data and people that can be consulted for knowing how the organization makes decisions in their trial projects (see Table C.WP4-2 in Annex C). Specifically, the stakeholders were required to indicate if they have collaboration and negotiation documents used in the requirement engineering activities of the trial project. Also, information about communities that is necessary to interview for specific questions regarding decision making strategies and decision biases.
- Information about the use of e-participation platforms in RE (requirements engineering) activities in the organizations (see Table C.WP4-3 in Annex C). Specifically, the stakeholders were required to indicate if they use the concept from e-democracy and e-participation in the RE activities of their projects.
- Expectation for the OpenReq results (see Table C.WP4-4 in Annex C). The questions in this block aim to know how OpenReq results could contribute to take decisions in the trial projects.
- Finally, the stakeholders attending the meeting were invited to give comments or indicate aspects that may be relevant to OpenReq related with the collaboration and decision-making of the selected project (see Table C.WP4-4 in Annex C).

WP5 interview: Knowledge and Dependency Management

The expected roles of the interviewees were the requirements engineer and the product manager.

This interview was designed to take 45 minutes.



The information required from the stakeholders is described next.

- Information about the current work of requirement engineers and product managers in the trial project and its work in previous projects of the same or other organizations (see Table C.WP5-1 in Annex C).
- Information about current situation of knowledge and dependency management in the trial project (see Table C.WP5-2 in Annex C). Specifically, the stakeholders were required to indicate what artifacts are used in the project, and the nature of these artifacts (typology, properties, how are they defined, how are they documented). Concerning dependencies, the types of relationships defined among requirements are required (interdependencies and/or decompositions), and for each type the different subtypes.
- Expectations from the OpenReq results (see Table C.WP5-3 in Annex C). The questions in this block are the following: Besides natural language, where could 'machine readable' data come from? What (additional) artifacts could or should be adopted? What are the main points in requirement lifecycle? What kinds of dependencies or relations are problematic? What problems are caused by non-perfect management or identification of requirements or relations?
- Finally, the stakeholders assisting to the meeting were invited to give comments or indicate aspects that may be relevant to OpenReq related with the management of knowledge and dependencies of the selected project (see Table C.WP5-4 in Annex C).

WP6 interview: OpenReq Interfaces

The expected roles of the interviewees were the requirements engineer and the project manager.

This interview was designed to take 40 minutes.

The information required from the stakeholders is described next.

- Information about the current work of requirement engineers and product managers in the trial project and their work in previous projects of the same or other organizations (see Table C.WP6-1 in Annex C).
- Information of the tools that are currently used in the trial project (see Table C.WP6-2 in Annex C). This information is needed in order to connect the new services provided by the OpenReq platform with these tools. The stakeholders were also required to explain the methodology adopted in using the tools and their functionalities and limitations. Finally, the stakeholders were asked about the sources used in the trial projects as input to extract the requirements, and also stakeholders and users feedback.
- Information about interfaces used regulations, and standards required (see Table C.WP6-3 in Annex C). The questions in this block were necessary to derive the OpenReq requirements concerning interfaces implementation. The stakeholders were asked about what kind of interfaces or connectors is used in the tools to interact with external systems, and whether the tools can be extended with external provided functionalities or services. Questions about import/export formats, standards, and regulations of the organizations concerning interfaces were also included.
- Finally, the stakeholders assisting to the meeting were invited to give comments and indicate aspects about the interaction that may be relevant to OpenReq (see Table C.WP5-4 in Annex C).



2.1.3 Instruments for requirements documentation

In this section, we introduce the instruments used for requirements documentation. These instruments were implemented in Tuleap, a project management collaborative tool used by the OpenReq project partners. The instruments used are the following:

- User stories and epics for the documentation of functional requirements.
- User stories, epics and acceptance criteria for the documentation of non-functional requirements.
- UML domain models for the documentation of the current structure and properties of requirements managed by the trial partners.
- UML domain model for the documentation of the structure and properties of the requirements as they will be represented in the OpenReq framework and their integration with the models of each trial partner.
- A glossary of terms that appear in epics, user stories, and UML domain models.

In the following we describe each instrument.

2.1.3.1 User stories and epics for documenting functional requirements

First of all, it is important to note when epics and user stories are used:

- User stories are used to represent individual functional requirements
- Epics are used to represent groups of functional requirements related by a common theme. User stories and epics are described in Deliverable D1.4 of OpenReq.

Functional requirements user stories

When a new user story is defined for a functional requirement, the values of the attributes in Table D-1 of Annex D have to be introduced. Among them we mention:

- *As a*: Type of stakeholders concerned (e.g. as a manager, as a developer...).
- *I want to*: What functionality is required.
- *In order to*: Which is the goal of including the functionality as a requirement.
- Acceptance criteria: This attribute may not have a value initially when the requirement is created, but the acceptance criteria should incrementally emerge as the requirement is refined.
- *Links*: Epic with which the user story is related.

Functional requirements epics

When a new epic is defined for a group of functional requirements, the values of the attributes in Table D-2 of Annex D should be introduced. Among them we mention:

- *WP Task*: Two tasks should be selected:
 - One task (and optionally more) from WP2-6 to connect the epic to the technical WPs that its functional requirements are related to.
 - One task (and optionally more) from WP7 to relate the epic to the trial or trials that are the source of its functional requirements.
- *As a*: Type of stakeholders concerned (e.g. as a manager, as a developer...).



- *I want to*: What functionality is required.
- *In order to*: Which is the goal of including the functionality as a requirement.

2.1.3.2 User stories, epics and acceptance criteria for documenting non-functional requirements

The way in which non-functional requirements are documented, by means of an epic, a user story, or an acceptance criteria, depends on two dimensions: their scope and the detail of their description. The scope indicates if the requirement is local, i.e., it applies to a single functionality (user story), system-wide (applies to the whole OpenReq platform) or applies to a group of functionalities (several user stories). The detail of the requirement description indicates whether the requirement is generic or detailed.

Thus, depending on the combination of the two dimensions, a different type of instrument is used (see Table D.3 in Annex D). The idea behind the decision is that more concrete and measurable non-functional requirements are documented by means of user stories and their acceptance criteria, and less concrete or non-measurable non-functional requirements are documented by means of epics.

Non-functional requirements user stories attributes

When a new user story is defined for a non-functional requirement, it should have the following attributes: As a, I want to, In order to, Acceptance criteria, Links, Cross ref, Status, Submitted by, Submitted on, Type, and Value.

The semantics of most attributes is the same as for functional requirement user stories (see Table D.1 in Annex D). Some specific aspects of NFR user stories are the following:

- *Links*: They should relate the non-functional requirement user story to the functional requirement user stories to which it applies.
- *Type*: We use the types proposed by the Volere template [1]: look and feel, usability, performance, operational, maintainability and portability, security, cultural and political, legal. We also allow the value other for non-functional requirement that do not fall into the predefined categories.

Non-functional requirements epics attributes

When a new epic is defined for a non-functional requirement, the value of the following attributes should be defined: Epic Type, As a, I want to, In order to, Acceptance criteria, Links, Status, Submitted by, Submitted on, Type, Value, and WP Task.

The semantics of most attributes is the same as functional requirements (see Table D.3 in Annex D). A specific aspect is the following:

• Type: We propose to use the types proposed by the Volere template [1]: look and feel, usability, performance, operational, maintainability and portability, security, cultural and political, legal. We also allow the value other for non-functional requirements that do not fall into the predefined categories

2.1.3.3 Trial Partner domain models

During the requirements analysis phase of the OpenReq project, it became clear that the trial partners already have requirement management systems in place where they maintain the requirements of the projects in which they are working, and specifically of the project that was chosen as trial of OpenReq.



Another fact that emerged is that OpenReq framework is not going to be a system that could substitute completely the partner requirement management systems, but that should extend these systems adding to them new functionalities and features.

In order to satisfy the above constraint, one type of instruments chosen for documenting requirements was UML conceptual models. After the visit to each trial provider, the corresponding partner was required to provide a UML conceptual model that describes which is the structure and properties of the requirements that they use in their projects. These conceptual models, acting as domain models, are going to be used to describe and facilitate the understanding of the concepts used in the trial partners.

2.1.3.4 OpenReq domain model

The OpenReq domain model is the representation in a UML conceptual model of the concepts managed in the OpenReq framework. The OpenReq domain model was constructed during requirements analysis considering the trial partners domain models and several requirement engineering standards as ReqIF. At the same time, the OpenReq domain model was gradually integrated to the trial provider domain models also expressed in UML (see section 2.1.3.3).

2.1.3.5 Glossary

The use of a glossary allows the understanding of concepts in the requirements that could be ambiguous or misunderstood by the partners. For each concept that one of the partners consider necessary to be defined, a new entrance in the glossary (wiki document of the collaborative Tuleap tool) was introduced.

For each entrance in the glossary the following properties were introduced:

- *Term*: Concept to be defined.
- *Definition*: Explanation of the concept.
- *Trial*: Trial partner that induced the definition of the new concept.
- *Owner*: Person that has introduced the definition.
- *References*: Source or sources from which the term and its definition was obtained.

2.2 Trial on-site visits

The agenda of the on-site visits had the structure that can be seen in Figure 3. In general, we had two or three interviewees depending on the trial. The interviews were conducted by one interviewer with the assistance of all the participants of the visit who could eventually act as co-interviewers when additional clarifications were needed. The interview slots were organized sequentially to facilitate plenary assistance (except for the first visit). Finally, the visits ended with a slot dedicated to wrap up and extract final conclusions.





Figure 3: Agenda of each trial partner visit

Next, we report the trial on-site visits. For each visit we indicate the assistants roles, the agenda and other relevant aspects such as detected positive and negative aspects of the visit execution and lessons learned.

2.2.1 Cross-platform OSS trial: Qt

The assistants to the Qt visit were the following:

- From Qt, there were three assistants that we will refer to as Qt1, Qt2 and Qt3. Qt1 is a Product Manager and a key person in the RE process and product management, Qt2 is R&D manager, and Qt3 is a RMS expert and R&D manager.
- From technical work packages, the assistants were: Davide Fucci (HITEC, WP2), Cristina Palomares (UPC, WP3), Martin Stettinger (TUGraz, WP4), Mikko Raatikainen (UH, WP5). Dolors Costal (UPC) also assisted as coordinator of T1.2, which has as a goal the specification of requirements for OpenReq platform. From WP6, no people assisted, and the role of interviewer for this work package was taken by Davide Fucci.

At the start of the day, Mikko made a presentation about OpenReq and the Qt trial and answered questions of the interviewees about it. Qt1 already had previous knowledge of OpenReq, since he was in the kick-off OpenReq meeting. The opinion of interviewees was that OpenReq may help them in requirements prioritization, decision support, recommendations, etc. However, they stated their concern about the possible burden that OpenReq could cause to their requirements engineering activities.

Positive aspects of the visit were the following:

• The interviewees were willing to collaborate. They tried to give good explanations and think about the expected benefits that OpenReq can provide to their requirements engineering processes.



- The plenary session for the "as-is" and WP2 interview worked well. In general, it was interesting for the rest of WP partners to attend the rest of the interviews (especially the as-is interview) because it gave them the possibility to get knowledge of the Qt context and ask for clarifications when needed. The rest of interviews were carried out in parallel sessions although the preferred option for the assistants was that of plenary sessions for interviews. This is the reason why in the following visits the interviews were done sequentially.
- Having various interviewees in the same interview at the same time was not a problem. In general, the interviewees had a balanced participation in giving the answers to the questions of the interviews. However, in the interviews for WP3 and WP4, having Qt1 (who covers a more senior position) in the interview with Qt2 could have influenced Qt2 answers, this scenario seemed to have not caused any problem as the Qt hierarchy is quite relaxed However, we have to take into account that in other companies the hierarchy is well marked and such situation may pose a problem, the interviewees might not give full or sincere answers.
- The interviewers think that the visit has improved their knowledge of the requirements of the Qt trial.

A negative aspect of the visit was that Qt1 was not present in the morning sessions. It would have been positive to have had also him to answer the as-is interview, as he was probably the best person to ask those questions.

As final conclusions:

- Redundancies were identified in the different interview scripts used. There was no time to eliminate the redundancies from the scripts for the next visit (i.e., Siemens) since it was conducted in a few days from the Qt visit but it was planned to be done for the remaining interviews.
- The order of the questions in the as-is interview could be improved. As it was used in Qt, the interview went from the general aspects (goals and so on), to the specifics of the RE process. After the visit, the conclusions were that it would be better to do it the other way round (from RE particularities to general), since while answering the general questions about goals the interviewees were giving details of their RE process because it was necessary to understand the goals. Because of that, it was decided to change the questions ordering for the subsequent visits.

The post visit work was:

- The WP partners should record the requirements obtained from the visit in the Tuleap tool.
- The WP partners should prepare a report with the key findings regarding their WP from the visit. The report had to be sent to Qt (the interviewees asked for this), so they could point out or amend what they wished.

2.2.2 Transportation trial: Siemens

The assistants to the Siemens visit were the following:



- From Siemens we had four assistants that we refer to as Si1, Si2, Si3 and Si4. Two of them, Si1 and Si2, are Siemens' OpenReq members and Si3 and Si4 are both requirements managers.
- From technical work packages the assistants were: Zijad Kurtanovic (HITEC, WP2), Cristina Palomares (UPC, WP3), Martin Stettinger (TUGraz, WP4), Mikko Raatikainen (UH, WP5). Dolors Costal (UPC) also assisted as coordinator of T1.2, which has as a goal the specification of requirements for OpenReq platform. From WP6, no people assisted, and the role of interviewer for this work package was taken by Dolors Costal.

We had two types of interviewees. On the one hand, Si1 and Si2 that are directly involved in the OpenReq project. They provided, during the interviews, a lot of information on what Siemens expects from OpenReq. On the other hand, Si3 and Si4 who are not so familiar with the OpenReq project, have a very detailed knowledge of the requirement engineering activities of the trial project. Si3 and Si4 received information about OpenReq objectives during the OpenReq project presentation before the interviews.

Positive aspects of the visit were the following:

- The trial is well defined and has clear objectives that were explained by Si1 and Si2 during the visit.
- The interviews were conducted sequentially, without parallel interviews, as decided in the previous visit to Qt (see Section 2.2.1). The reason was to facilitate the assistance of all the participants to all the interviews.
- The interviewees provided good collaboration. When interviewing two people together (e.g., Si3 and Si4) they both had a quite balanced participation in the responses. The impression was that they both were able to express their opinions.

Negative aspects of the visit were the following:

- Si3 and Si4 were not available during the second day except for a short slot in the morning. Fortunately, Si1 and Si2 have a good knowledge of the trial and could answer most of the questions and the pending questions were asked during the slot when Si3 was available. This mitigated the negative effects of this aspect.
- There were redundancies in the different interview scripts used. This already happened in the Qt visit, however, there was no time to eliminate these redundancies from the scripts due to the few days between the two visits. As all the interviewers were present during all the interviews, they tried to skip the redundancies.

As final conclusions, sequential and plenary interviews were considered positive by all the present partners and it was perceived that one day and a half is needed to perform all the interviews sequentially.

The post visit work was:

- The WP partners should record the requirements obtained from the visit in the Tuleap tool.
- The WP partners should prepare a report with the key findings regarding their WP from the visit. This will be given to Siemens so they can provide final feedback if necessary.

2.2.3 Telecom trial: WindTre

The assistants to the WindTre visit were the following:



- From WindTre there were three assistants that we refer to as Wi1, Wi2 and Wi3. The three of them are technology architects.
- From technical work packages the assistants were: Davide Fucci (HITEC, WP2), Cristina Palomares (UPC, WP3), Elisabetta Angelillo, Claudia Pandolfo (ENG, WP3), Martin Stettinger (TUGraz, WP4), Mikko Raatikainen (UH, WP5). Dolors Costal (UPC) also assisted as coordinator of T1.2, which has as a goal the specification of requirements for OpenReq platform.

Wi1 is directly involved in the OpenReq project and had a good knowledge of it and clear expectations on what the OpenReq project can contribute to WindTre. Wi1 had already presented the OpenReq project to Wi2 and Wi3 prior to the visit so we decided not to include the initially planned slot "OpenReq project presentation to the interviewees" in the final agenda.

Positive aspects of the visit were the following:

- The interviews were conducted sequentially, without parallel interviews. This made possible the assistance of all the participants to the interviews, a fact that had been perceived as positive by all the partners in previous visits.
- The interviewees provided good collaboration. However, there was not a balanced participation of the three interviewees since two of them provided most of the responses. In this case, this should not be considered negative since the two predominant interviewees provided the two complementary visions of the two companies that have been merged in WindTre and the third provided additional comments when needed and, as a result, the positive aspect is that the development of the interviews was quite agile.
- The trial is well defined and has clear objectives related to user explicit feedback analysis that were explained by Wi1. The interviews facilitated the identification of relevant sources of information other than social media, e.g., ticketing system.
- In the wrap up and final conclusions slot the accompanying partners of the OpenReq trials summarized the most relevant goals of each trial.

Negative aspects of the visit were the following:

- During the interviews of the first day Wi2 and Wi3 were not present in the room (due to transportation unexpected problems) and assisted through video conference which made difficult to have a balanced participation and sometimes a well understanding. Additionally, Wi2 was not present during the first interview of the first day. Fortunately, both attended in person the second day and this facilitated the clarification of some points mainly related to the as-is situation of the trial that had not been fully developed the day before.
- The difficulty to obtain the as-is situation of the trial is that WindTre is in a merge phase, and they do not have a clear definition of the requirement engineering processes that will be used after the merge. To overcome this, at some points, questions were answered regarding the two merged companies.

As final conclusions, the visit has been positive to obtain a good knowledge of the trial objectives and sequential and plenary interviews, especially when all the interviewees attended in person, worked well to facilitate the assistants to improve their general knowledge about the requirements of the WindTre trial.

The post visit work was:



- The WP partners should record the requirements obtained from the visit in the Tuleap tool.
- The WP partners should prepare a report with the key findings regarding their WP from the visit. This will be given to WindTre so they can provide final feedback if necessary.

2.2.4 Vogella potential trial

The assistants to the Vogella visit were the following:

- From Vogella there were three assistants that we refer to as Vo1, Vo2 and Vo3 and who are all Vogella's OpenReq members.
- From technical work packages the assistants were: Davide Fucci, Walid Maalej, Zijad Kurtanovic, Timo Johann (HITEC, WP2), Cristina Palomares (UPC, WP3), Martin Stettinger (TUGraz, WP4), Hanna Mäenpää (UH, WP5). From WP6, no people assisted, and the role of interviewer for this work package was taken by Cristina Palomares.

In this case, we only had one type of interviewee, the three of them being members of the OpenReq project. In this case, therefore, the presentation of the OpenReq project was not necessary. In addition, Vo2 is a contributor to the Eclipse Project and is the one with more experience in the project. This is the reason why most of the questions were answered by Vo2, and only some follow-up comments were answered by Vo1 and Vo3.

Positive aspects of the visit were the following:

- The interviews for the as-is and all the work packages were conducted sequentially, without parallel interviews, so all the participants could assist to all the interviews.
- The interviewees were willing to collaborate. They tried to give good explanations and think about the expected benefits that OpenReq can provide to their requirements engineering processes.
- The interviewers think that the visit has been positive to improve their knowledge about the requirements of the Vogella trial.

Negative aspects of the visit were the following:

- Due to the availability of the trial partner, only three hours were available for conducting the interviews. This forced the interviewers to skip some of the questions of their script.
- Additionally, during the interview, some follow-up questions were made by other participants different than the main interviewee. Although these follow-up questions provided interesting insights, some of them were far of the focus of the main questions.
- These two facts made that in the case of Eclipse the data gathered was not as detailed as for the rest of trial partners, but sufficient for the gathering of their needs for the OpenReq system.

As final conclusions, sequential and plenary interviews were considered positive by all the present partners and the visit has been positive to obtain an overall knowledge of the trial objectives. However, it was perceived that only half day is not enough for gathering all the details that are needed.

The post visit work was:

- The WP partners should record the requirements obtained from the visit in the Tuleap tool.
- The WP partners should prepare a report with the key findings regarding their WP from the visit. This will be given to Vogella so they can provide final feedback if necessary.



2.3 Consolidation meetings

Several consolidation meetings were handled with the aim of obtaining a set of requirements without redundancies, prioritized and all at the same level of abstraction.

The first meeting was carried out internally at UPC to provide an analysis of requirement redundancies and level of abstraction, which served as a basis for the discussion of the next meeting.

After that, a face-to-face meeting was coordinated with all scientific partners where the following points were handled:

- Analysis of requirements elicited from the trial visits to ensure the homogenization of both wording and level of abstraction.
- Final analysis of requirements to detect missing requirements corresponding to requirements stated in the DoA (but not appearing during the visits) and other requirements that scientific partners thought would be convenient to have.
- Negotiation to resolve conflicts, especially related to which task(s) or work package(s) the requirement was related to.
- Final prioritization of requirements. Three levels of priority were established in this meeting:
 - Priority 1. Requirements that shall be incorporated in the OpenReq system by M12.
 - Priority 2. Requirements that shall be incorporated in the OpenReq system by M18.
 - Priority 3. Requirements that will be revisited after M18 to establish their priority.

Levels 1 and 2 correspond to the first and second releases of the OpenReq system, while requirements at level 3 correspond to requirements that will be incorporated in further releases of OpenReq.

After this meeting, HITEC and UPC evolved the requirements according to the analysis, prioritization and negotiation results. Telco calls were coordinated as needed to clarify further details with a scientific partner. This last evolution of the requirements gave place to the final version of the requirements.

2.4 Requirements

As a result of the T1.2 requirements analysis we have obtained a total of 45 epics and 94 (preliminary) user stories. There are 10 epics related to WP2, 12 epics for WP3, 4 epics for WP4, 11 epics for WP5 and 7 epics for WP6. Regarding trials, we have that there are 27 epics related to Qt, 19 epics related to Siemens, 22 epics for WindTre and 17 epics for Vogella. As a result of T1.2 an initial version of the domain models of the trial partners have also been obtained, jointly with an initial version of the OpenReq domain model and its integration with the trial partner domain models.

This section describes the key findings about requirements regarding each of the scientific work packages of the OpenReq project.

The complete set of epics and user stories that form the requirements and the glossary associated to them can be found in the annexes to this document, more concretely, epics can be found in Annex E, user stories in Annex F, domain models in Annex G and the glossary in Annex H.



2.4.1 Key findings regarding WP2

WP2 addresses the use of artificial intelligence features in the definition of software requirements. The key findings concerning this working package are organized by trial.

In Qt we have found an extensive use of Jira and mailing lists for the management of requirements. Regarding Jira, they use basic Jira dashboard integrated with Aha!. They do not use social media or usage data for the elicitation of requirements. Finally, it was detected the explicit need of obtaining information for requirements from the sales meetings.

Siemens is interested in extracting requirements data from technical documents. Currently, the extraction is done from the documents in MS Word and they are imported to DOORS. There is no implicit feedback (or anonymized) about requirements.

WindTre is interested in obtaining requirements data from Facebook and Twitter (in Italian) suggesting requirements evaluation. They concentrate information (effort, ROI) in a holistic dashboard where they share information from different departments In order to check the requirements derived from social media, they also want to triangulate data with customers' tickets.

In the case of Vogella they would like to integrate OpenReq visualization within the existing one from Bitergia and they want to detect usage patterns and perform A/B testing.

2.4.2 Key findings regarding WP3

WP3 addresses the use of recommendations of requirements for stakeholders.

Regarding Qt it was observed that there are duplicate issues in their current tool and they have too much information that needs to be filled out by users when entering a new issue. Thus, they indicate that the use of recommendations could eliminate the duplicates, since the existing requirements could be reused in new projects, and at the same time, they would avoid entering too much information.

The interest of Siemens regarding WP3 is in the help for differentiating different requirements contained in the same paragraph of a call for tenders document, the help for differentiating text that corresponds to actual requirements from the rest of text (prose) in a call for tenders document and the identification of requirements in previous projects that are similar to requirements of the current projects they are working on.

In the case of WindTre and Vogella there are no key findings specifically related to WP3.

2.4.3 Key findings regarding WP4

WP4 addresses the use of group decision support during requirements engineering evaluation processes. The key findings concerning this working package are organized by trial.

Qt discusses and takes decisions during requirements evaluation just sitting together in groups.

For Siemens, group decision support is needed. They did not use it until now, but they think it could be used in several activities that they do as: to take general decisions, to take strategic or ad-hoc decisions, to classify requirements, to assign stakeholders, to decide the risk of requirements, to decide on the available technologies for resolving the requirements, for deciding on requirements similarities, and for taking decisions on usability. Also, for role allocation to requirements.



In the case of WindTre, taking into account that their main concern is the extraction of requirements from messages in social networks, decision support could be used in case of doubts on the new requirements to be added from the analysis of messages in networks as Facebook or Twitter.

Vogella needs decision support for the negotiation of requirements and for decisions on what to do next. Currently if a group thinks some requirement is needed then there is a discussion round and if one team member does not agree then the requirement is not accepted and they are not allowed to implement it because everything is very consensus driven.

2.4.4 Key findings regarding WP5

WP5 addresses the management of knowledge and dependencies during requirements engineering. The key findings concerning this working package are organized by trial.

Qt uses Jira for the requirements management. Currently they have requirements as Jira items that sometimes overlap and can be related to each other by hierarchical links or references. They need that the new OpenReq framework integrates with Jira and extends the Jira functionalities improving them, because they would like to continue using that system.

Siemens uses DOORS for the requirement management. They define requirements and relationships between them. Ontologies, both RE and domain ontologies could be used to help in requirements management. They also want requirements and metadata reuse.

WindTre uses word and excel files for including the requirements. Afterwards, they are uploaded in a document repository. They think that they are not interested in the possible contributions of WP5, since they are mainly interested in the extraction of requirements more than on the requirements management at least in the short term. For the long term, they would be interested in a description of the requirements ontology, in adding requirements reuse to their requirements engineering processes, and in the general requirements engineering process improvement.

Vogella uses Bugzilla for the requirements management. The requirements are defined in free text and accompanied with rich metadata about their characteristics and related activities. They are stored in a dedicated, freely available requirements management system. They would like that OpenReq framework could be integrated. However, they do not have a current specific interest in the WP5 contributions in the short term.

The summary of the key findings in the different visits is that all the trial partners are interested in maintaining the requirements management systems they are currently using and that OpenReq could be integrated with those systems, extending them with the new functionalities and features that are going to be provided by the framework. However, in the case of WindTre and Vogella, they do not see any specific WP5 contributions to their projects in the short time. The other two trial partners think that the requirements management and the interdependencies that are going to be managed in OpenReq can contribute to improve the quality of the requirements managed in their projects.

2.4.5 Key findings regarding WP6

WP6 addresses the interfaces that will be needed in OpenReq to be used by final users or by existent requirements management systems. The key findings concerning this working package are organized by trial.



For the trial project Siemens wants to continue using DOORS for the requirements management. With this idea, the OpenReq interfaces to be developed must be able to exchange data with DOORS.

In the case of WindTre the requirements engineering process formalization is in progress due to the recent merge of Wind and H3G to form WindTre. Key findings for WindTre are that currently the requirements are written in Word and Excel files which are uploaded on an internal document repository and that cloud is not used for security reasons.



3 ARCHITECTURAL DESCRIPTION OF OPENREQ

3.1 OpenReq architecture

We describe the OpenReq system components and their relationships. See Figure 4. The OpenReq prototype, as well as the different trial applications, access the different services provided by the components. In general the different components talk in a bi-directional fashion to each other. Exceptions from this bi-directional communications are, for instance, the dependency service as well as the social service which just provide information to other components. The components have access to the knowledge infrastructure which is based on the OpenReq Ontology. Figure 4 shows the major OpenReq components including the corresponding communication interfaces. For instance, the release planning service, which is developed within the scope of the work package 4 component, uses the dependency management service which is part of work package 5 component.



Figure 4: UML Component Diagram of the OpenReq overall architecture.



The OpenReq overall architecture is divided into the following parts:

- OpenReq Knowledge Infrastructure (OpenReq global database based on the OpenReq Ontology)
- OpenReq MicroServices services (RESTful services)
- OpenReq Prototype (a showcase of selected OpenReq functionalities)
- Different, stakeholder-specific applications (developed by the trial partners: Wind Tre, QT, Siemens and Vogella)

3.1.1 OpenReq knowledge infrastructure

The OpenReq knowledge infrastructure component is responsible for managing OpenReq ontologies, glossaries, indexes (e.g., references), stakeholder profiles, user feedback as well as usage and interaction logs. In some cases, specific OpenReq MicroServices will use their own very specific database. In these cases, it won't make sense to include very specific details of a single MicroService into the global OpenReq database.

OpenReq Ontologies can be categorized into the following categories:

- RE ontology (language for modelling requirements including concrete instances, i.e., the requirements model).
- Different domain ontologies (modelling the domains of the applications, e.g., the trials).
- Reuse and patterns ontology (language for requirements reuse and patterns). The OpenReq reuse and pattern catalogue provides a reusable knowledge base of requirements. The catalogue will be organized according to one or more classification schemas that will be derived from different assets, such as the domain ontologies mentioned above or specific standards.

The current OpenReq RE Ontology for the OpenReq prototype is illustrated in Figure 5. The OpenReq prototype serves as showcase for selected OpenReq functionalities. Besides others, release planning is one selected key functionality of the OpenReq prototype as to why the following Ontology is tailored towards the release planning scenario using the prototype. A more detailed description of the OpenReq Ontologies will be included in deliverable D5.3: OPENREQ Ontologies & Pattern Catalogue.





Figure 5: A specialization of OpenReq Ontology that is tailored towards the release planning scenario using the OpenReq prototype

3.1.2 OpenReq MicroServices

The different components of OpenReq (see Figure 4) will serve different services (MicroServices) for specific functions such as calculating a group recommendation. The OpenReq Prototype and the industry trials exploit the basic functionalities provided by the OpenReq services. Furthermore, services themselves exploit/integrate the services of other components, for example, release planning service use functionalities of dependency detection service, recommendation, and group decision making. In the following, we give a short overview of the different components shown in Figure 4 (for more details, we refer to the OpenReq DoA as well as the deliverables which deal with the architecture of the specific work packages D2.1, D3.1, D4.1 and D5.1). The components are a set of MicroServices (REST services) that includes, among others, the following components (engines):

- Recommender Component (work package 3)
- Dependency Component (work package 5)
- Group Decision Component (work package 4)
- Requirement Intelligence Component (work package 2)

In the following, we describe the different major components with regard to the services they will provide to the OpenReq users.


Recommender Component (Work Package 3)

The recommender component is provided in terms of a set of micro-services, for instance, querying for concrete requirements (in contrast to other information included in requirements documents that are not requirements), recommendations of responsible stakeholders as well as recommendations of reusable requirements (e.g., similar / related requirements from past projects). A conversation with the system can be triggered depending on the type of the query and the stakeholder activity. Recommendations are calculated, among others, on the basis of collaborative, content-based or knowledge-based recommendation technologies. Other components using the services of this component are, for example, the Group Decision Component, for the purpose of stakeholder recommendation in the context of release planning. A detailed technological description of this component can be found in deliverable D3.2 (Recommender Engine Version 1 and D3.4 Recommender Engine Version 2).

Dependency Component (Work Package 5)

This component uses different natural language processing approaches that support the detection and extraction of requirements dependencies. Furthermore this component detects conflicts that could occur, for instance, in Release Planning. For repairing inconsistent release assignments, a conflict resolver is used. Other components using this component are, for example, the Group Decision Component in the context of release planning and the Recommender Component in the context of requirements quality analysis. Example services of this component in the context of release planning are: (1) check whether the assigned requirements to a release exceed the maximum capacity of the release and (2) check whether the assigned requirements are assigned to meaningful releases so that, e.g., high priority requirements are not violated. A detailed technological description of this component can be found in deliverable D5.2: Requirements Dependency Engine Version 1 and D5.4: Requirements Dependency Engine V2.

Group Decision Component (Work Package 4)

This component supports groups of users (stakeholders) in the context of group decision making. First, the individual preferences of the different group members (stakeholders) have to be collected. An example of individual stakeholder preferences is, individual ratings regarding a specific requirement property (for example, the estimated effort of the requirement). After the preference acquisition is done by a group of users, this component calculates a group recommendation based on the given stakeholder preferences which satisfies the group as a whole. Furthermore this component also supports the moderation of decision processes in such a way that consensus among stakeholders can be achieved (in case of contradicting preferences). Group decision services are used, for example, by the OpenReq prototype (release planning scenario), the Siemens trial, and the Vogella Trial. The detailed technological description of this component can be found in D4.2 Group Decision Engine Version 1 and D4.3 Group Decision Engine Version 2.

Requirement Intelligence Component (Work Package 2)

This component is based on an analytics backend and includes text-mining algorithms [2] that allow the analysis of natural language texts such as textual documents and different types of user feedback. Figure 6 shows the different types of user feedback—explicit or implicit. In addition to this, interactive visualisation will be supported by this component. In particular,



interactive visualisation supports stakeholders in visualizing descriptive and predictive analytics data. Figure 7 shows an example of such a visualisation and presents the trend of different app review types (e.g., a user requests a new feature or a bug to be fixed) over time, for a specific app and over its different versions. In general, this component will extract potential requirements and their metadata (e.g., popularity) from the feedback presented in Figure 3 and make them available to the other components. For example, the Group Decision Component (Work Package 4) can use this component to extract stakeholder preferences. A detailed technological description of this component can be found in D2.2 Requirements Intelligence Engine Version 1 and D2.3 Requirements Intelligence Engine Version 2.



Figure 6: Examples of explicit and implicit feedback





3.1.3 OpenReq prototype

The OpenReq prototype serves as a showcase for the major OpenReq functionalities. This is also the major showcase for the end users, therefore a special focus will be given on user experience and user interface design. This design will be successively adapted and improved depending on the results of conducted user studies. In all cases where end users interact with OpenReq functionalities, a special focus will be given on usability. A major focus of the User Interface (UI) design is to follow a responsive design style which allows the application of OpenReq functionalities also in mobile environments. The OpenReq Prototype is based upon



the latest web-technologies and will run on a Spring Boot application including the Thymeleaf as well as the Bootstrap framework. A major goal of the OpenReq Prototype is to show OpenReq functionalities in an integrated fashion which goes beyond the application in the individual OpenReq trials. As a basis for the implementation of the OpenReq prototype, a corresponding user interface design was created on the basis of Adobe XD.

3.1.4 Different stakeholders applications

Besides the OpenReq prototype, the trial partner applications of Siemens, QT, WindTre, and Vogella will access the different OpenReq Services. Communication among OpenReq components and the integration of OpenReq components with the trial clients (e.g., the Siemens DOORS and Angular clients) will be supported by the interfaces of the OpenReq MicroServices (see the following section).

The OpenReq MicroService Interfaces will provide open, unified interfaces for easily integrating OpenReq into different external tools in form of REST connectors (see Figure 1). Examples of high-level interfaces include (a) get Stakeholder Recommendation, (b) check For Consistency, (c) extract Requirements From Document, (d) search For Stakeholders With Special Skills. In this context (a) and (d) are related to the Recommender Component, (b) is related to the Dependency Component and (c) is related to Requirements Intelligence Component.

The following describes a complete scenario showcasing the interaction of the different layers of the architecture in the context of release planning. In this case, a user interacting with the OpenReq prototype adds a new requirement to a project. While the user enters the requirement to the project the system automatically screens via the recommendation service if there are other relevant requirements (e.g., from past projects) which can be of relevance for the current user (e.g., other requirements which are related to the currently entered one). If the recommendation service finds new potential requirements these are returned and presented to the user via the OpenReq prototype user interface. The following figure (Figure 8) depicts the described example call chain in the context of release planning scenario.



Figure 8: Complete call-chain of an user interaction with the OpenReq prototype



Within the scope of the trial implementations, the trial partners will integrate these interfaces into the tools they use in the context of their scenarios (e.g., the Siemens DOORS client) and associated Open Source communities, for example the Eclipse community.

3.2 OpenReq MicroServices

MicroServices developed within the scope of the OpenReq Project will be described as follows. The MicroServices can be called via a unique method name (see - <u>Naming</u> <u>Conventions of OpenReq MicroServices</u>). Parameters as well as return values are described in JSON format. For the detailed API description, we will use the Swagger framework.

The following example represents a MicroService following the OpenReq description rules:

GET openreq.esl.eng.it:8080/api/projects/1/release?id=2

A Java-specific example of this call could look like the following:

ConsistencyResult checkConsistency(Project project, Release release);

This method checks whether a given set of requirements is consistent with a given release or not. In a first implemented version, this function (implemented on the basis of constraint solving technologies) checks whether the sum of the efforts exceeds the maximum capacity of the release. If a release is "overbooked" the method returns a description as to why the current setting is not consistent including indicators for resolving the inconsistencies (for instance, release assignments which have to be changed).

The following exemplifies a project description in JSON format as output of:

GET openreq.esl.eng.it:8080/api/projects/1

```
Project (in JSON Description):
[{
  "id": 1,
       "version": 1.2,
  "name": "OpenReq Prototype",
  "requirements":
               [{"requirement":{
                       "id": 1,
                       "effort": 254,
                       "assignedRelease": 2,
                       "dependencies":
                         {"RequiresDependency": 3,
                               "RequiresDependency": 8}
                                      L
                              }
               }]
  "releases":
I
  {"release":{
               ʻid": 2,
               "maxCapacity": 300}}
```



```
]
}]
```

```
ConsistencyResult (in JSON Description):
[{
    "consistent": true,
    "explanation": "Explanation of the Consistency Check"
}]
```

3.3 Catalogues of OpenReq MicroServices

There will be two different documents describing the available set of OpenReq MicroServices. The first document contains all public available MicroServices developed by the OpenReq partners. These MicroServices are well described following the OpenReq naming conventions and documentation rules. Due to the fact that some MicroServices need internal calls to other MicroServices to serve their functionality, a second document will be made available where a complete list of all MicroServices developed in OpenReq is described in detail. To avoid duplicate development of related functionality, each partner can request a non-public MicroService to transfer it to a public MicroService. The transfer process has to be fulfilled by the creator of the MicroService. As a first step, these two documents are shared Google documents. In the future, we plan to have a tooling that allows to search for a specific MicroService by means of the functionality which is needed.

3.4 Port Restrictions of OpenReq MicroServices

To follow our integration partner's (ENG) security restrictions, all deployed MicroServices are available over one single Port (8080) from outside. The ports are mapped via a reverse proxy by ENG to the specific ports of the MicroServices. To call a MicroService developed by Graz University of Technology which is called "groupRecommendations" from outside, one has to call: GET

<u>https://openreq.esl.eng.it:8080/</u> tugraz/grouprecommendationservice/grouprecommendations.

To ensure an independent execution of the OpenReq MicroServices as well as the continuous integration and extension of the OpenReq functionalities, the OpenReq MicroServices will use a dedicated internal port per partner. The following list shows the assigned port pool for all the OpenReq partners.

- TU Graz owns internal Ports from: 9000 9200
- UH owns internal Ports from: 9201 9400
- UPC owns internal Ports from: 9401 9600
- HITEC owns internal Ports from: 9601 9800
- Vogella owns internal Ports from: 9801 10000
- WindTre owns internal Ports from: 10001 10200
- QT owns internal Ports from: 10201 10400
- Siemens owns internal Ports from: 10401 10600
- ENG owns internal Ports from: 10601 10800
- External Others: 10801 11000



3.5 Naming Conventions of OpenReq MicroServices

To ensure a common and understandable description of all the developed OpenReq MicroServices, we use the following naming convention:

"partner name" + "/" + "unique name of service"

For instance:

"partner name" = tugraz "unique name of the service" = GroupRecommendationService

results in a REST-Endpoint like:

GET <u>https://openreq.esl.eng.it:8080/</u> tugraz/grouprecommendationservice/grouprecommendations



REFERENCES

[1] Robertson, S.; Robertson, J, Mastering the requirements process: getting requirements right, 3rd ed, Addison-Wesley, 2013.

[2] C. Castro-Herrera, C. Duan, J. Cleland-Huang, and B. Mobasher, "Using Data Mining and Recommender Systems to Facilitate Large-Scale, Open, and Inclusive Requirements Elicitation Processes," in 2008 16th IEEE International Requirements Engineering Conference, 2008, pp. 165–168.



ANNEX A. ROLES AND RESPONSIBILITIES IN REQUIREMENTS' ANALYSIS

	Requirement provider	Accompanying partner	WP2-6 leaders	Coordinator
Wind Tre	Wind Tre + ENG	UPC	HITEC (WP2)	UPC
Siemens	SIEMENS	TUGraz	UPC (WP3) TUGraz (WP4)	
Qt	QT	UH	UH (WP5)	
Vogella	VOGELLA	HITEC	ENG (WP6)	

Table A.1. Roles developed by the different partners in each trial

Table A.2. Roles'	' responsibilities	for each	activity in	n Task pr	reparation
	100000000000000000000000000000000000000				

	Requirement provider	Accompanying partner	WP2-6 leaders	Coordinator (UPC)
Prepare Instruments for req. elicitation	Identify one or two past or current projects that will be used as context to obtain current needs in req. elicitation instruments (e.g. questions will be asked about these projects' way of working and issues to be improved by OpenReq) Identify relevant scenarios / workflows for OpenReq. Suggestion: Organize internal session (with internal stakeholders) to present OpenReq (and mentioning the pre-questionnaires and visits) and get them involved. This session can be useful to identify context projects and relevant scenarios for OpenReq in case they have not been identified in advance.	Identify stakeholde obtain requirements scientific interests (technical WPs they Propose questions f role to obtain requi scientific interests (questionnaire or in Questions may be b related to the select If a workshop is ne creative requirement scientific interests, to conduct during of Participate in the had different designed if questionnaires and	rs roles relevant to s about their (i.e. about the lead) for each stakeholder rements about their (to be asked in a an interview). both general or red real projects eded to obtain nts related to their propose a workshop on-site visits. armonization of the instruments (pre- interview guides)	Coordinate preparation of elicitation instruments Propose pre- questionnaires with common/simple-to- answer questions Define how the information will be recorded in the Tuleap tool
Prepare req. documentation artifacts		Provide feedback o documentation arte	n requirement facts	Prepare initial proposal of req. documentation artefacts Adapt initial proposal to feedback received
Prepare visits	Communicate date constraints for the visits Communicate in advance any NDA to be signed or any other data protection procedure needed to be known for the protocol design.	Plan dates for visits (taking into account involved partners constraints)	Provide constraints for visits	



Organize the Tuleap tool space	
for the visit	

Table A-3: Partners' responsibility for each activity

	Requirement provider	Accompanying partner	WP2-6 leaders	Coordinator (UPC)
Prepare on-site visit	Help accompanying partner to identify on-site stakeholders for OpenReq Arrange infrastructure for the visits Arrange participation of selected stakeholders in visits	Identify stakeholders (with the help of the requirements provider) Customize pre- questionnaires for stakeholders Obtain pre-questionnaires responses and record them in Tuleap Prepare agenda for activities Prepare activities (considering NDAs; also logistics -copies, tape recorder,)	Customize instruments such as interview guides for each visit (if needed) Assist accompanying partner in identifying stakeholders and customizing pre- questionnaires	Coordinate preparation
Conduct visit	Participate in the visit activities	Prepare introduction of project Prepare wrap-up	Conduct interviews on their scientific interests Participate in visit activities	
Collect results from visit	Revise results from visits (e.g., validate transcriptions)	Record visit resulting assets in Tuleap (including transcribing interviews, if NDA allows)	Record resulting req. obtained about their scientific interests in Tuleap Revise results from visits Identify key findings regarding technical WPs (from trials and scientific requirements)	Record as-is situation Coordinate consolidation of results
Visit retrospective	Participate in the visit retrospective	Participate in visit retrospective	Participate in visit retrospective	Coordinate visit retrospective



ANNEX B. DATA COLLECTION IN PRE-QUESTIONNAIRES TO TRIAL SITES

Pre-questionnaire parts and questions

Part 1 - General organization information

Table B.1. Information about the respondent of the general organization information

part

Respondent name
Respondent email
Respondent telephone
Job position in the organization
Role in the project (i.e. OpenReq trial)
Years of experience in requirements engineering

Table B.2. Information about the organization

For which application domains does your organization develop software?

What services does your organization provide?

How many different products do you currently have in the market?

How many different products do you have under development / maintenance?

How many employees did your organization have at the end of 2016?

How many employees were working in software development at the end of 2016?

How many employees were working in R&D at the end of 2016?

Other relevant details of the organization

Table B.3. Information about data, data privacy and data access

For the development of OpenReq, the research partners need access to project, product, and usage data, especially regarding requirements. Please, describe under which conditions you could provide access to these data.

Specifically, what are these data sources?

Do you have documentation about these data sources and their structure?



Table B.4. Information about strategic goals

Which are the main strategic goals of the organization?

Which are the main strategic goals of the trial/project?

Table B.5. Information about requirements engineering process

What are the methods, standards, and tools used for, and the people responsible for:

- identifying/eliciting requirements?

- documenting requirements?
- negotiating requirements?
- prioritising requirements?
- validating requirements?
- managing requirements?

Table B.6. Expectations on OpenReq

OpenReq will provide support in requirement engineering for...

OpenReq will help my organization to improve / increase / reduce...

OpenReq should not...

Part 2 – Project-specific information

Table B.7. Information about the respondent of the project-specific information part

Respondent name
Respondent email
Respondent telephone
Job position in the organization
Role in the project (i.e. OpenReq trials)

Years of experience in requirements engineering



Table B.8. Information about the selected project (a project could be the development of a whole product or just a part of it)

Project name	
Team composition	
Number of people involved	
Kind of project (i.e. new development, adaptation or delivery of pre-existi solutions, etc.)	ng
Other project details	

Table B.9. Information about the selected product

Product main functionality

Product domain (e.g. financial services, telecom, manufacturing industry, etc.)

Product type (proprietary or open source)

Number of releases completed (if any)

Number of requirements and its type (e.g. user stories, epics, use cases, etc.)

Programming languages used for the development

Is the product general or customized to different customer segments?

Other relevant details of the product

Table B.10. Information about methods, standards, tools, and responsible people on the project

What are the methods, standards, and tools used for, and the people responsible for:		
Software life cycle?		
 Planning, monitoring, managing and controlling the project? 		
Eliciting functional requirements?		
 Documenting functional requirements? 		
Prioritizing functional requirements?		
Eliciting non-functional requirements?		
 Documenting non-functional requirements? 		
Prioritizing non-functional requirements?		
Designing?		
Coding?		
Testing?		
What other methods, standards and tools are used in the project?		



Table B.11. Information about the templates used and documentation about the project

Is there any documentation describing the processes, methods and standards used in the project?

Does your organization have/use standard templates for documenting:

- (1) project plans, WBS, time sheets, etc.?
- (2) functional requirements?
- (3) non-functional requirements?

Are the templates or documentation supported by a tool? Which ones?

What, if any, metadata (or properties) is attached to requirements according to the tools or templates?



ANNEX C. DATA COLLECTION IN INTERVIEWS AT TRIAL SITES

As-is and working packages interviews

As-is interview

Table C.ASIS-1. Information about the role of the requirements manager and the decision-maker or business expert in the organization

Warm-up questions (<5 minutes)

Q1.1: Explain your role in the organization

- How long have you been working in the organization?
- How long have you been working in the trial/project?
- What is your professional background? (e.g., management, informatics...)
- What is your role in the project (especially in relation to requirements)?
- What are the decision-making tasks associated to your role in the organization? And in the project? How do your decisions affect the trial/project?
- Do you have any previous experience in requirements engineering? If so, how long and in what roles?

Table C.ASIS-2. Information about the way to work in the organization related with requirements engineering activities

Tools or processes currently used in the trial/project for requirements engineering (<20 minutes)

Q2.1 How would you describe your way of working? [Do you use waterfall, agile, scrum, hybrid, etc?]

Q2.2 What do you understand by requirement in your organization?

• Do you have a shared and established definition/conceptualization of requirements?

Q2.3 Do you differentiate between functional and non-functional (i.e. quality) requirements?

• If yes, do you treat functional and non-functional requirements at the same time or separately?

Q2.4 What are the sources of requirements in the context of your project?

Q2.5 How do you identify and elicit requirements in the context of your project?

- Who (which role) is responsible for requirements elicitation?
- What kind of tools, practices and techniques do you use for eliciting requirements?
- What are the current challenges in the requirements identification and elicitation processes?
- Is there any distinction between functional and non-functional requirements on that matter?

Q2.6 How do you document requirements in the context of your project?

- Who is responsible for requirements documentation?
- What kind of tools, practices and techniques do you use for documenting requirements? Do you specify or model requirements?
- What are the current challenges in the documentation of requirements?
 - In particular, are there any challenges related to the properties or quality of the requirements?
- How do you communicate requirements with customers?



•	How do you communicate requirements internally? [within the team and with other teams]
•	Is there any distinction between functional and non-functional requirements on these matters?
Q2.7 H	ow do you negotiate requirements in the context of your project? Who is responsible for requirements negotiation? What kind of tools, practices and techniques do you use for negotiating requirements?
• Q2.8 H	 What are the current challenges in the negotiation of requirements? ow do you validate requirements in the context of your project? Who is responsible for requirements validation? What kind of tools, practices and techniques do you use to validate requirements? o How do you check the quality of the requirements? o How is user feedback regarding requirements taken into account? What are the current challenges regarding requirements validation? Is there any distinction between functional and non-functional requirements on these matters?
Q2.9 H	 ow do you manage requirements in the context of your project? What methods, techniques, tools, or practices regarding requirements management do you use in the context of your project? What kind of tools, practices and techniques do you use in requirements prioritization? Are there any rules for prioritizing requirements? When and how often are they prioritised? How do you communicate information regarding requirements prioritization with relevant stakeholders? How do you communicate information regarding requirements prioritization internally? [within the team and with other teams] Is there any distinction between functional and non-functional requirements on that matter? What are the current challenges regarding requirements management and specifically regarding requirements prioritization?
Q2.10 H relate to	How does requirements engineering (requirements development and management process) to other areas of development in the context of your project?



Table C.ASIS-3. Information about the strategic goals of the organizations

Strategic goals of the organization (<5 minutes)

Q3.1: Which are the main strategic goals1 of the organization?

Q3.2: Which roles define and manage the previous strategic goals of the organization?How do you interact with these roles?

Q3.3: How the success of the previous organization strategic goals is measured? (strategic indicators)

- Do you use indicators? Which ones?
- How are these indicators measured?
- Which roles define and manage these indicators?

Table C.ASIS-4. Information about the strategic goals of the project selected for the trial

Trial/project strategic goals related to RE (< 10 minutes)
Q4.1: Which are the trial/project strategic goals?
Q4.2: Which roles define and manage the trial/project strategic goals?How do you interact with these roles?
 Q4.3: How is the success/failure of the trial/project strategic goals measured? Do you use indicators? Which ones?
How are these indicators measured?
Q4.4: Do requirements engineering activities influence the achievement of trial/project strategic goals? How?
requirements elicitation?
requirements documentation?
requirements negotiation?
requirements validation?

• ...requirements management and prioritization?



Table C.ASIS-5. Strategic goals of the organization and the project selected for the trial

Relation between the strategic goals of the organization and the goals of the trial/project (< 5 minutes) $\,$

Q5.1: Which strategic goals of the organization are related to the trial/project strategic goals?

Q5.2: How the strategic indicators of the organization are related to the trial/project strategic goals?

Q5.3: How the trial/project success (trial/project indicators) is related to the strategic goals of the organization?

- How does the success of the trial/project contribute to accomplish the strategic goals of the organization?
- How the failure of the trial/project makes difficult to accomplish the strategic goals of the organization?

Table C.ASIS-6. Expectations for OpenReq results

OpenReq expectations (< 10 minutes)

Q6.1 What areas of improvement would you like to see in the requirements engineering process in the context of your project?

- What areas of improvement would you like to see related to requirements elicitation?
- What areas of improvement would you like to see related to requirements documentation?
- What areas of improvement would you like to see related to requirements negotiation?
- What areas of improvement would you like to see related to requirements validation?
- What areas of improvement would you like to see related to requirements management and prioritization?
- Do you have any suggestions for improvement in any of these areas?

Q6.2 Which information do you think that would be valuable (and when) to improve your requirements engineering process (for your project in particular and for your organization in general)?

Q6.3 Which roles of the organization are the target user of the OpenReq framework?

Q6.4: Which strategic goals of the organization can be contributed to by OpenReq results?

Table C.ASIS-7. Other informations that can be relevant for OpenReq

Additional comments (< 5 minutes)

Q7.1: Are there any related issues that we missed and that you would like to reflect on?



WP2 interview: Software Requirements Intelligence

Table C.WP2-1. Information about the role of the requirements engineer and theproduct manager in the organization

Warm-up questions (<5 minutes)

Q1.1: Explain your role in the organization?

- How long have you been working in the organization?
- How long have you been working in the trial/project?
- What is your professional background? (e.g., management, informatics...)
- What is your role in the project (especially in relation to requirements)?
- Do you have any previous experience in requirements engineering? If so, how long and in what roles?

Table C.WP2-2. Information about data analytics

General questions related with the WP2 (<10 minutes)

Q2.1: Do you use data-analytics?

- In which context?
- Do you use data analytics specifically for requirement engineering?
- Why? What is your goal in using data in the context of requirement engineering
- Can you please briefly describe it (from a technical standpoint)
- What kind of algorithms do you use?
- How do you evaluate the accuracy and performance of the data analytics?

Q2.2: What data do you currently use?

- For which purpose? for each data
- Where does the data come from? for each data
- Is it hard/easy to access? for each data
- How accessible is this data that can be used in OpenReq? for each data

Q2.3: What data that you do not currently collect you think might be useful for requirement engineering?

Q2.3: To what extent scalability (due to data) can be an issue?

Q2.4: What do you think are the criteria that a data analytics solution for requirement engineering should have to be successful?

- [Help by suggesting some criteria, e.g. from literature]
- [Better if these can be prioritized]



Table C.WP2-3. Information about explicit data collection

Explicit data collection (<10 minutes)

Q3.1: Do you collect explicit data (e.g., from social media (SM), trackers, Q&A site, forum)? Why?

- Do you collect explicit data from your competitors?
- Do you take into account the type of user that generated such data?
- For how long have you collected such data?
- How often do you collect such data?
- What is the amount of data you collected?
- How important are explicit data for your company?
- And specifically for requirements/release -related decisions?
- How important is giving feedback to your user about their request via these channels?
- What (other) purpose you collect explicit data for? (e.g., profiling customers)
- How accessible is this data for the use in OpenReq?

Q3.2: Do you analyze this data? Why?

- How?
- How often?
- How do you use the analysis result?
- And specifically for requirement/release?

Q3.3: Are you aware of any legal and privacy aspects that could be relevant for the collection of explicit data in the context of OpenReq?

Table C.WP2-4. Information about implicit data collection

Implicit data collection (<10 minutes)

Q4.1: Do you currently collect usage data (e.g., from sensors)? Why?

- Which one?
- How often?
- What is the amount of data you collected?
- Do you take into account the type of user that generated such data?
- How important is this data for taking decision your company?
- And specifically for requirement engineering?
- How accessible is this data to be used in OpenReq?

Q4.2: Do you analyze this data?

- How?
- How often?
- How do you use the analysis result?
- And specifically for requirement/release?

Q4.3: Are you aware of any legal and privacy aspects and limitation that could be relevant for the collection of implicit data in the context of OpenReq?



Table C.WP2-5. Information about visualization and integration

Visualization and integration of results of data analysis (<10 minutes)

Q5.1: Do you currently make use of the results of data analytics?

- How do you visualize the results? Can you show us an example?
 - Are the visualization created automatically? How?
 - How this information visualization is used to take decision?
 - How is the visualized data then put into action?
 - For what purpose?
 - Can you guide us through the decision making based on the previous example?

Q5.3: What tools do you currently use for requirements engineering tasks?

• How are these tools integrated with 3rd party? What kind of integration you use, if any?

Q5.4: What is the result you trying to obtain with such integration, specifically for requirement engineering?

Table C.WP2-6. Expectations for OpenReq results

OpenReq expectations (< 10 minutes)

Q6.1: What are the main expectations of the OpenReq framework related to the software Requirements Intelligence in the context of your company?

• Is there any preference about the user interface design?

Table C.WP2-7. Other information that can be relevant for OpenReq

Additional comments (< 5 minutes)

Q7.1: Are there any related issues that we missed and that you would like to reflect on?



WP3 interview: Personal Recommendations for Stakeholders

Table C.WP3-1. Information about the role of the requirements manager or analyst

Warm-up questions (<5 minutes)

Q1.1 Describe your work experience. [Subject's experience may impact his/her perspective on understanding the needs related to the requirements engineering process]

- How long have you been working in the organization?
- How long have you been working with the product?
- What is your role in the project (especially in relation to requirements)?
- Do you have any previous experience in requirements engineering? If so, how long and in what roles?

Table C.WP3-2. Information about the requirements reuse and recommendations in the trial project

Requirements engineering in the trial project (<20 minutes)
 Q2.1 Do you do reuse requirements in the context of your project? How do you reuse requirements? In which stage of the requirements engineering process (elicitation, specification, analysis, etc.) do you reuse requirements? What kind of tools, practices and techniques do you use for reusing requirements? Is there any distinction between functional and non-functional requirements on that matter? Who is responsible for reusing requirements? What are the current challenges in the requirements reuse process? How do you create and maintain a requirements reuse base of knowledge (i.e. the
 requirements knowledge to reuse)? Who is responsible for creating and maintaining this base of knowledge? What are the current challenges in creating and maintaining this base of knowledge?
 Q2.2 What do you do to achieve requirements of high quality in the context of your project? What kind of tools, practices and techniques do you use to achieve requirements of high quality? Is there any distinction between functional and non-functional requirements on that matter? Who is responsible of evaluating the requirements quality? What are the current challenges in order to achieve requirements of high quality?
 Q2.3 How do you give value to requirements properties (such as risk and effort) in the context or your project? What requirements properties are you exactly using? What kind of tools, practices and techniques do you use to give value to requirements properties? Is there any distinction between functional and non-functional requirements on that matter?
 Who is responsible of giving value to requirements properties?

• What are the current challenges in order to give value to requirements properties?

Q2.4 How do you identify relevant stakeholders in the context of your project?



- What kind of tools, practices and techniques do you use to identify relevant stakeholders?
- Who is responsible of identifying relevant stakeholders?
- What are the current challenges in the identification of relevant stakeholders?

Table C.WP3-3. Expectations for OpenReq results

OpenReq expectations (< 10 minutes)

Q3.1 What are the main expectations of the OpenReq framework related to automatic recommendations in the context of your project?

- Which are the automatic recommendations that you think that would be required from the OpenReg framework?
 - Are there any recommendations that could be related to the screening and reuse of requirements?
 - Are there any recommendations that could be related to achieve requirements of high quality?
 - Are there any recommendations that could be related to give value to requirements properties?
 - Are there any recommendations that could be related to identify relevant stakeholders?
 - Are there any other recommendations related to other area that you think could be interesting to have?
- Which data could be used as source data to make recommendations to users?
- Which are the roles that would need these recommendations?
- What tools used by managers, developers, users, etc. are envisaged to be connected to the OpenReq framework in relation to recommendations?
- Is there any preference on when to receive these recommendations?
- Is there any preference about the user interface design used for the recommendations?
- What are the main challenges that you think that are related to the possible recommendations provided by OpenReq?

Table C.WP3-4. Other information that can be relevant for OpenReq

Additional comments (< 5 minutes)

Q7.1: Are there any related issues that we missed and that you would like to reflect on?



WP4 interview: Group Decision Support

Table C.WP4-1. Information about the role of the requirements manager or analyst

Warm-up questions (<5 minutes)

Q1.1 Describe your work experience

- How long have you been working in the organization?
- How long have you been working with the product?
- What is your role in the project (especially in relation to requirements)?
- Do you have any previous experience in requirements engineering? If so, how long and in what roles?

Table C.WP4-2. Information about data and people that can be consulted for knowing how the organization take decisions in their trial projects

Decision biases (<20 minutes)

Q2.1: Do you have collaboration and negotiation documents from past RE Tasks?

• Can we have access to it?

Q2.2: Which communities can we interview for specific questions regarding decision making strategies and decision biases?

Q2.3: Have you ever thought of decision biases during your last decisions?

- Which ones?
- What did you do against it?

Table C.WP4-3. Information about the use of e-participation platforms in RE activities in the organizations

Decision biases (<10 minutes)

Q3.1:Do you use Concept from E-Democracy and E-Participation?

• Which ones?

Table C.WP4-4. Expectations for OpenReq results

Decision group support approach (< 80 minutes)

Q4.1: Which decision support tools do you use?

- Are there indicators for inconsistent preferences inside this tool? Which?
- Are there techniques to motivate users/stakeholders to increase their engagement?
- Are users/stakeholders be informed if ned information is available (new alternatives / new comments etc.)

Q4.2: How is guaranteed that all the decision relevant information / knowledge is exchanged between the stakeholders / users?

Q4.3: Does information of previous decisions influence ongoing (future) decisions?

Q4.4: How is the progress of the requirements model measured?

• Is there any help for stakeholders to resolve the open issues?

Q4.5: Which types of groups exist in the different RE scenarios of the company?

Q4.6: Which requirements engineering tools are currently in use?

Q4.7: In which ways are user communities integrated into RE processes?

Q4.8: In which way can communities be integrated in order to better figure out the most relevant requirements?

Q4.9: Is planning poker used to evaluate requirements (w.r.t. related efforts)?



Q4.10: In which way are stakeholders integrated into quality assurance processes of requirements?

- Any need to support the (automated) identification of innovative stakeholders (or members of the user community)?
 - Any need to integrate further concepts that help to increase stakeholder motivation to actively participate in RE processes (e.g., in requirements gathering)?

Q4.11: What is the size of the user communities and what are ideas for future (maybe more intensive) integration of these communities into RE processes?

Q4.12: How can user communities provide feedback to planned new functionalities/ requirements?

Q4.13: How does release planning work in current development scenarios and which tools are used?

Q4.14: How are requirements evaluated (w.r.t. which dimensions?)?

Q4.15: How are requirements negotiated? -is there any triage mechanism?

Q4.16: Which mechanisms are used to foster communication and information exchange in online RE platforms?

Q4.17: What functionalities are expected for future release planning?

Q4.18: In which way should communities be integrated in future requirements engineering scenarios (e.g., in the context of evaluating requirements, release planning etc.)?

Q4.19: In which situations would a group recommendation make sense (e.g., a recommendation of a release plan that takes into account all preferences of the individual stakeholders)?

Q4.20: In the case of contradicting preferences (e.g., regarding the prioritization of a requirement or the evaluation of a requirement), what are the corresponding decision processes and approach to "tie break"?

Q4.21: What are the practices to achieve consensus in the case of contradictory opinions?

Q4.22: We plan to conduct a user study on existing biases in decision making - which communities can be contacted w.r.t. such a study?

Q4.23: In which contexts does a group decision support in terms of a tool make sense?

Q4.24: Which functionalities are regarded as key functionalities to support group decision processes in requirements engineering?

Table C.WP4-5. Other information that can be relevant for OpenReq

Additional comments (< 5 minutes)

Q5.1: Are there any related issues that we missed and that you would like to reflect on?



WP5 interview: Knowledge and Dependency Management

Table C.WP5-1. Information about the role of the requirements engineer and product

manager

Warm-up	questions (<5 min	utes)
0115			

- Q1.1 Describe your work experienceHow long have you been working in the organization?
 - How long have you been working with the product?
 - What is your role in the project (especially in relation to requirements)?
 - Do you have any previous experience in requirements engineering? If so, how long and in what roles?

Table C.WP5-2. Information about current situation of knowledge and dependencymanagement in the trial project

Knowledge and dependency management in the current situation (<20 minutes)

Q2.1: What artifacts (or concepts) are used in requirements engineering (or are directly related to requirements engineering)

- Potential concepts such as (needs), issue, requirement, user stories, "shall" requirements, epics, use cases, features, bug, software component, patterns and templates, roadmap, release plan, others
- What terminology (about concepts etc) is used in requirements management, how terms are understood)

Q2.2: What is the nature of each artefact (or concept) as identified in the previous question

- How is it differentiated whether an artefact is new or a modification for an old one such as bug fix, improvement, change, etc.
- What, if any, kinds of typology is applied (e.g., non-functional and functional requirements, requirement pattern /template types in use)
- How well-defined are the artifacts
 - \circ ~ as concept: How clearly and similarly used throughout the organization
 - individual instances of concept conformance to concept definition and its facets
 - homogeneity e.g. in granularity / strictness; are all same kind of artifacts similar
- What kinds of properties or metadata is applied
- How the artefact is documented
 - What kind of information system, tools etc.is used for the artifact? (e.g to document, store, manage, analyze)
- What kinds of operations are carried out for the artefacts (e.g., kill, merge, split, refine, prioritise, approve, assign)

Q2.3: What kinds of relations artefacts have

- What kinds of parallel relations exist
 - within same type of artefact (e.g. requirement-requirement)
 - what kinds of models artefacts constitute
 - how big models / how many instances/types
- What kinds of refinement / traceability / generalization relations and their traceability exist (e.g. from requirement to feature, feature to sw component)



 how do you determine what artefact instances are affected (e.g., what sw components are affected by a requirement, issue or feature)

Q2.4: What kinds of dependencies and relationships exist

- What types and their importance, e.g. requires, blocks 'needed before', incompatible / conflicting, competes of same resources, (is-a), composed of/part-of, implements, others...
- How hard or strict the relationships are? (soft vs hard constraints)
- How complete the relationships are? (How much explicated)
- Where do the relations and dependencies emerge and are noticed (Origins, sources etc)
- How relationships are documented or modelled?

Table C.WP5-3. Expectations for OpenReq results

How can OpenReq contribute (< 15 minutes)

Q3.1: Besides natural language, where could 'machine readable' data come from

Q3.2: What (additional) artifacts could or should be adopted?

Q3.3: What are the pain points in requirement lifecycle?

Q3.4: What kinds of dependencies or relations are problematic?

Q3.5: What problems caused by non-perfect management or identification?

Table C.WP5-4. Other information that can be relevant for OpenReq

Additional comments (< 5 minutes)

Q4.1: Are there any related issues that we missed and that you would like to reflect on?



WP6 interview: OpenReq Interfaces

Table C.WP6-1. Information about the role of the requirements engineer and project manager

Warm-up questions (<5 minutes)

Q1.1 Describe your work experience

- How long have you been working in the organization?
- How long have you been working with the product?
- What is your role in the project (especially in relation to requirements)?
- Do you have any previous experience in requirements engineering? If so, how long and in what roles?

Table C.WP6-2. Information of the tools that are currently used in the trial project

Tools used in RE activities (< 15 minutes)

Q2.1: What tools are you currently using in RE activities of the trial project? And in other organization projects?

- Which is the provider of these tools?
- Are these tools extended with some plugin?
- Are these tools integrated with other development tools?

Q2.2: What methodology is adopted in the trial project?

Q2.3: Are you missing some functionality or are there limitations in the tools indicated before? Which ones?

Q2.4: Which sources are used for requirements?

Q2.5: Which sources are used to obtain for stakeholders' and users' feedback?

Q2.6: What issue trackers do you currently use, if any?

- If you do use any, which ones?
- Can you guide us through the structure of your issue trackers?
- Can you give a short example of how the tracker is used? (e.g., fields (custom), status sequence).

Table C.WP6-3. Information about interfaces used, and regulations and standards required

Interfaces in the trial project (<20 minutes)

Q3.1: What kind of interfaces or connectors is used in the tools indicated in previous questions to interact with external systems?

Q3.2: What kind of formats is used in the tools, indicated before, for importing and exporting information?

Q3.3: Could the tools, indicated before, be extended with extensions that implement complementary functionalities?

Q3.4: Is there some standard or some regulation in your organization of the interfaces implemented in the systems that you use?

Q3.5: Is there or some regulation in your organization of the interfaces implemented in the systems that you use?



Table C.WP6-4. Other information that can be relevant for OpenReq

Additional comments (< 5 minutes)

Q4.1: Are there any related issues that we missed and that you would like to reflect on?



ANNEX D. ATTRIBUTES AND SELECTION OF INSTRUMENTS FOR REQUIREMENTS' DOCUMENTATION

Attribute	Definition
ID	Identifier of the user story
As a	Type of stakeholders concerned (e.g. as a manager, as a developer)
I want to	What functionality is required
In order to	Which is the goal of including the functionality as a requirement
Acceptance criteria	We may not have a value for this attribute initially when the requirement is created, but the acceptance criteria should incrementally emerge as the requirement is refined
Links	They should relate the user story to its epic
Cross ref	A link to other entities in Tuleap
Status	Corresponds to the status of the user story (the status can be: proposed, under negotiation, accepted, rejected, ongoing, implemented, tested, and finished).
Submitted By	Who defined the user story
Submitted On	When the user story was defined
Туре	Its value should be "Functional" (for functional requirements user stories)
Value	The value that the requirement gives to potential adopters of OpenReq. (the value may be: differentiating, must-have, nice-to-have)

Table D.1. Attributes of user stories in functional requirements

Table D.2. Attributes of cpies in functional requirements	
Attribute	Definition
ID	Identifier of the epic
Еріс Туре	Its value should be "Epic".
WP Task	Two tasks should be selected:
	• One task (and optionally more) from WP2-6 to connect the epic to the technical WPs that its functional requirements are related to.
	 One task (and optionally more) from WP7 to relate the epic to the trial or trials that are the source of its functional requirements. Task 7.1

Table D.2. Attributes of epics in functional requirements



	refers to the Vogella trial ² , task 7.2 refers to the OSS trial, task 7.3 refers to the transportation trial and task 7.4 refers to the telecom trial.
As a	Type of stakeholders concerned (e.g. as a manager, as a developer)
I want to	What functionality is required
In order to	Which is the goal of including the functionality as a requirement
Acceptance criteria	We may not have a value for this attribute initially when the requirement is created, but the acceptance criteria should incrementally emerge as the requirement is refined
Links	Artifacts in other trackers pointing to an epic (assigned automatically)
Status	Corresponds to the status of the epic (the status can be: proposed, under negotiation, accepted, rejected, ongoing, implemented, tested, and finished).
Submitted By	Who defined the epic
Submitted On	When the epic was defined
Туре	Its value should be "Functional" (for functional requirements epics)
Value	The value that the requirement gives to potential adopters of OpenReq. (the value may be: differentiating, must-have, nice-to-have)

² Since Vogella has not a dedicated task in the DoA, we have chosen to assign task T7.1 to identify it.



Scope	Detail	Example NFR	Tuleap representation
local	generic	The functionality of user story #1 must have a good response time	 User story (NFR user story) With a link to the functional user story to which it applies
	detailed	The functionality of user story #1 must take less than 1 second, 90% of the times	Acceptance criteriaOf the functional user story to which it applies
system-wide	generic	The system must be usable	Epic • The description of the epic must clarify it is system-wide (by referring to "the system" or "OpenReq platform")
	detailed	The system must allow reaching any functionality in no more than 3 clicks	 User story The description of the user story must clarify it is system-wide
group of functionalities	generic	The critical functions of the system must have a good time response	 Epic The description of the epic must clarify to which group of functionalities it applies (e.g. "critical functions of the system")
	detailed	The critical functions of the system must take less than 0,25 seconds, 90% of the times	 User story The description of the user story must clarify to which group of functionalities it applies or Links to the user stories it applies or Acceptance criteria
			• Of the user stories to which it applies



ANNEX E. EPICS DETAILED LIST

Table E.1. Recommend related requirements from same projects

Attribute	Definition
ID	Epic 141
Title	Recommend related requirements from same projects
Туре	Functional
Goal	Stakeholders want to know if there are related requirements to the one requirement they are creating
	Stakeholders want to improve the requirements database
Priority	Medium
Related with Work Package Tasks	WP3 (T3.2), WP7 (T7.2, Qt)

Table E.2. Recommend related requirements from same projects

Attribute	Definition
ID	Epic 142
Title	Recommend missing information in requirements
Туре	Functional
Goal	Stakeholders want to know if something is missing in her requirements
	Stakeholders want to improve the requirements database
Priority	Low
Related with Work Package Tasks	WP3 (T3.3), WP7 (T7.2 – Qt)

Table E.3. Recommend requirement properties

Attribute	Definition
ID	Epic 143
Title	Recommend requirement properties
Туре	Functional
Goal	Stakeholders want to get recommendations for requirements properties
	Stakeholders want to improve the requirements database



Priority	Medium
Related with Work	WP3 (T3.3), WP7 (T7.2 – Qt)
Package Tasks	

Table E.4. Recommend relevant stakeholders

Attribute	Definition
ID	Epic 144
Title	Recommend relevant stakeholders
Туре	Functional
Goal	Stakeholders want to automatically assign assignees
	Stakeholders want to ease and improve the requirements creation process
Priority	Low
Related with Work Package Tasks	WP3 (T3.5), WP7 (T7.2 – Qt)

Table E.5. Recommend related requirements from previous projects

Attribute	Definition
ID	Epic 157
Title	Recommend related requirements from previous projects
Туре	Functional
Goal	Stakeholders want to get recommendations of requirements from previous projects related to the one they are entering
	Stakeholders want to ease and improve the requirements creation process
Priority	Medium
Related with Work Package Tasks	WP3 (T3.2), WP7 (T7.3 – Siemens)

Table E.6. Measure requirements quality

Attribute	Definition
ID	Epic 158
Title	Measure requirements quality
Туре	Functional
Goal	Stakeholders want to measure the quality of a requirement



	Stakeholders requirements	want	to	detect	problematic
Priority	Medium				
Related with Work Package Tasks	WP3 (T3.3), W	/P7 (T7.	1- Vo	gella, T7	3 – Siemens)

Attribute	Definition		
ID	Epic 171		
Title	Screen business requirements		
Туре	Functional		
Goal	Stakeholders want to get recommendations to screen requirements		
	Stakeholders want to ease and improve the requirements creation process		
Priority	Medium		
Related with Work Package Tasks	WP3 (T3.2)		

Table E.7. Screen business requirements

Table E.8. Recommend improvements on the quality of requirement

Attribute	Definition
ID	Epic 172
Title	Recommend improvements on the quality of requirement
Туре	Functional
Goal	Stakeholders want to be aided with tips Stakeholders want to improve the quality of the requirements created
Priority	Medium
Related with Work Package Tasks	WP3 (T3.3), WP7(T7.1-Vogella, T7.4-WindTre)



Attribute	Definition		
ID	Epic 173		
Title	Identify requirements		
Туре	Functional		
Goal	Stakeholders want to get recommendations for identifying requirements		
	Stakeholders want improve the requirements creation process		
Priority	High		
Related with Work Package Tasks	WP3 (T3.2), WP7(T7.1-Vogella, T7.3-Siemens)		

Table E.10. Recommend requirements and requirements metadata based	on the
context of the user	

Attribute	Definition
ID	Epic 181
Title	Recommend requirements and requirements metadata based on the context of the user
Туре	Functional
Goal	Stakeholders want to be informed of recommendations taking into account my contextual information Stakeholders want to be more efficient and not
	disturb my work routine
Priority	Low
Related with Work Package Tasks	WP3 (T3.6), WP7(T7.1-Vogella, T7.2-Qt, T7.3- Siemens, T7.4-WindTre)

Table	E.11.	Dashboard	for r	equirements	analytics

Attribute	Definition
ID	Epic 185
Title	Dashboard for requirements analytics
Туре	Functional
Goal	Stakeholders want to concentrate the information in a dashboard visible to all departments
	Stakeholders want to share a general vision inside the company
Priority	Low



Related with Work	WP2 (T2.5), WP7(T7.4-WindTre)
Package Tasks	

Table E.12. Analyze existing requirements from Requirement Management System

Attribute	Definition
ID	Epic 189
Title	Analyze existing requirements from Requirement Management System
Туре	Functional
Goal	Stakeholders want to provide Excel spreadsheets
	One or two Excel spreadsheets present the detailed requirements about the name of the product, and how much it costs, and what is the effort to be implemented
Priority	Low
Related with Work Package Tasks	WP2 (T2.2, T2.4), WP7(T7.3-Siemens, T7.4-WindTre)

Table E.13. Cluster user based on their input

Attribute	Definition
ID	Epic 190
Title	Cluster user based on their input
Туре	Functional
Goal	Stakeholders want to receive recommendations based on the type of customers There can be different types of users (for example, based on monthly spending or influencer)
Priority	Medium
Related with Work Package Tasks	WP2 (T2.4), WP7(T7.4-WindTre)

Table E.14. Integrate with marketing tools

Attribute	Definition
ID	Epic 194
Title	Integrate with marketing tools
Туре	Functional
Goal	Stakeholders want to include data from the marketing department in order to create requiements


	The marketing departments generate requirements after seen the analytics
Priority	Low
Related with Work Package Tasks	WP2 (T2.6), WP7(T7.4-WindTre)

Attribute	Definition
ID	Epic 195
Title	Get requirements based on similar products
Туре	Functional
Goal	Stakeholders want to see what are the most successful requirements in order to propose similar requirements.
Priority	Low
Related with Work Package Tasks	WP7(T7.4-WindTre)

Table E.15. Get requirements based on similar products

Table E.16. Analyze customer tickets system

Attribute	Definition
ID	Epic 198
Title	Analyze customer tickets system
Туре	Functional
Goal	Stakeholders want to use existing ticketing system data in order to have a more precise requirement analytics
Priority	Low
Related with Work Package Tasks	WP2(T2.2, T2.4), WP7(T7.4-WindTre)

Table E.17. Analyze interaction data

Attribute	Definition
ID	Epic 206
Title	Analyze interaction data
Туре	Functional
Goal	Stakeholders want to use the analysis of interaction data in order to improve the workflow
Priority	Medium



Related with Work	WP2(T2.3,	T2.4),	WP7(T7.1-Vogella,	T7.4-
Package Tasks	WindTre)			

Table E.18. Adhere to requirements engineering practices

Attribute	Definition
ID	Epic 275
Title	Adhere to requirements engineering practices
Туре	Functional
Goal	Stakeholders want to have the development teams to adhere to defined prescription in order to be compliant with OpenReq
Priority	Low
Related with Work Package Tasks	WP5(T5.4), WP7(T7.2-Qt, T7.3-Siemens, T7.4-WindTre), WP9 (T9.4)

Attribute	Definition
ID	Epic 276
Title	Store requirements using a predefined ontology
Туре	Functional
Goal	Stakeholders want to have the requirements stored adhering to a predefined structure in order to harmonize the work and facilitate integrability
	Provide common ontology (conceptualization) of requirements as artifacts including hierarchies such as epics, user stories and bugs.
	This ontology should be adhered to throughout the process but especially when storing. I tried to avoid too generic 'manage' wording.
Priority	High
Related with Work Package Tasks	WP5(T5.4), WP7(T7.2-Qt, T7.3-Siemens)

Table E.19. Store requirements using a predefined ontology

Table E.20. Assign requirements to a domain

Attribute	Definition
ID	Epic 277
Title	Assign requirements to a domain
Туре	Functional



Goal	Stakeholders want to be able to assign requirement for a predefined domain in order to improve the quality of requirements engineering	
	Domain ontology defines concepts specific for certain application domain. For example, railway system include interlocking, control room etc.	
Priority	Medium	
Related with Work Package Tasks	WP5(T5.4), WP7(T7.3-Siemens)	

Attribute	Definition
ID	Epic 278
Title	Extract dependencies from textual requirements
Туре	Functional
Goal	Within the requirements, there are several, on the one hand, written relationships and, on the other hand, implicit relationships that are not in the properties or meta-data of requirements. In order that these kinds of relationships could be exploited in full extent and machine automated or assisted manner, the relationships need to be extracted from the text.
Priority	Medium
Related with Work Package Tasks	WP5(T5.3), WP7(T7.2-Qt, T7.3-Siemens)

Table E.22. Comply with the relationships between requirements

Attribute	Definition
ID	Epic 288
Title	Comply with the relationships between requirements
Туре	Functional
Goal	Stakeholder want to have relationship between requirements explicit in order the product and its releases are valid.
	Capture (elicit or establish, maintain) and adhere to dependencies between different requirements entities in a context that already contains numerous (thousands) requirements.
Priority	High



Related with Work	WP5(T5.1, T5.2, T5.3, T5.4, T5.5), WP7(T7.2-Qt)
Package Tasks	

Table E.23. Take into account relationships of the properties of requirements

Attribute	Definition
ID	Epic 289
Title	Take into account relationships of the properties of requirements
Туре	Functional
Goal	Stakeholder want to take the properties into account in product management in order to be able to make better justified decisions about product management Establish and maintain dependencies between the different properties of requirements; and between properties and requirements including accumulate, compare or otherwise calculate properties. For example, plan a release so that the accumulated effort of included requirements is not over defined limit.
Priority	Medium
Related with Work Package Tasks	WP5(T5.2, T5.4, T5.5), WP7(T7.2-Qt)

Table E.24. Comply with the relationships affected by confidential requirements

Attribute	Definition
ID	Epic 290
Title	Comply with the relationships affected by confidential requirements
Туре	Functional
Goal	Stakeholder want to establish and take into account relationships to confidential requirements in order to decide on releases that are meaningful for all stakeholders. Dependency management of artifacts of different confidentiality. E.g. open source, internal requirements and customer-specific requirements.
Priority	Low
Related with Work Package Tasks	WP5(T5.4, T5.5), WP7(T7.2-Qt)

Attribute	Definition
ID	Epic 291
Title	Establish external relationships
Туре	Functional
Goal	Stakeholder want to have my product development to follow standards etc.external things in order to stay compliant.
	System and its requirements have relationships to external entities such as standards, technologies or external stakeholders. One example concrete example is C++ standard and its different versions. The focus here is to establish relationships to such an external relationship.
Priority	Low
Related with Work Package Tasks	WP5(T5.3, T5.5), WP7(T7.2-Qt)

Table E.25. Establish external relationships

Table E 26	Ensure adherence	to relationshins	over different	versions
1 abit 1.20.	Elisare auterence	to relationships	over uniterent	

Attribute	Definition
ID	Epic 292
Title	Ensure adherence to relationships over different versions
Туре	Functional
Goal	Stakeholder want to follow relationships over versions in order to ensure integrity and long-time support
	Take into account relation through requirements life- cycle. For example, legacy requirements or implementation, versions or revisions, out-dated or bending requirements, obsolete or deprecated systems or technologies.
	Take account different version or releases of a system as well as long-time support (LTS) and continuation of the support for specific versions for specific customers
Priority	Low
Related with Work Package Tasks	WP5(T5.2, T.5.4, T5.5), WP7(T7.2-Qt)



Attribute	Definition	
ID	Epic 296	
Title	Feedback on recommendations	
Туре	Functional	
Goal	Stakeholder wants to tell the system that a recommendation is wrong.	
	In order to teach the system and improve the recommendations in the future.	
Priority	Low	
Related with Work Package Tasks	WP3(T3.2, T.3.3, T3.4, T3.5), WP7(T7.1-Vogella, T7.2-Ot, T7.3-Siemens, T7.4-WindTre)	

 Table E.27. Feedback on recommendations

Attribute	Definition	
ID	Epic 305	
Title	Recommend similar requirements	
Туре	Functional	
Goal	Stakeholder wants to identify similar requirements from a specific requirement. In order to identify duplicates, not enter the same requirements twice or reuse information.	
Priority	High	
Related with Work Package Tasks	WP3(T3.2), WP5(T.5.3), WP7(T7.2-Qt, T7.3- Siemens)	

Table E.28. Recommend similar requirements

Table E.29. Manage requirements metadata in OpenReq prototype

Attribute	Definition
ID	Epic 312
Title	Manage requirements metadata in OpenReq prototype
Туре	Functional
Goal	
Priority	Low
Related with Work Package Tasks	WP6(T6.2), WP7(T7.1-Vogella, T7.2-Qt, T7.3- Siemens, T7.4-WindTre)





Attribute	Definition
ID	Epic 313
Title	Resolve the inconsistencies in the compliance ratings
Туре	Functional
Goal	Stakeholder as a bid manager in the project wants to see and resolve the inconsistencies in the compliance ratings of the requirements from the stakeholders
Priority	Low
Related with Work Package Tasks	WP4(T4.4)

Table E.30. Resolve the inconsistencies in the compliance ratings

Table E.31. Resolve	inconsistencies	of solution	approaches

Attribute	Definition	
ID	Epic 314	
Title	Resolve inconsistencies of solution approaches	
Туре	Functional	
Goal	Stakeholder as a system manager wants resolve inconsistencies of solution approaches on the preference and system level (group-based configuration).	
Priority	Low	
Related with Work Package Tasks	WP4(T4.4)	

Table E.32. Manage user account of OpenReq prototype

Attribute	Definition
ID	Epic 315
Title	Manage user account of OpenReq prototype
Туре	Functional
Goal	
Priority	Medium
Related with Work Package Tasks	WP6(T6.2), WP7(T7.1-Vogella, T7.2-Qt, T7.3- Siemens, T7.4-WindTre)



Attribute	Definition
ID	Epic 316
Title	Support user of OpenReq prototype
Туре	Functional
Goal	
Priority	Medium
Related with Work Package Tasks	WP6(T6.2), WP7(T7.1-Vogella, T7.2-Qt, T7.3- Siemens, T7.4-WindTre)

 Table E.33. Support user of OpenReq prototype

Table E.34. Manage project of OpenReq prototype

Attribute	Definition
ID	Epic 317
Title	Manage project of OpenReq prototype
Туре	Functional
Goal	
Priority	Medium
Related with Work Package Tasks	WP6(T6.2), WP7(T7.1-Vogella, T7.2-Qt, T7.3- Siemens, T7.4-WindTre)

Table E.35. Manage project release in OpenReq prototype

Attribute	Definition
ID	Epic 318
Title	Manage project release in OpenReq prototype
Туре	Functional
Goal	
Priority	High
Related with Work Package Tasks	WP6(T6.2), WP7(T7.1-Vogella, T7.2-Qt, T7.3- Siemens, T7.4-WindTre)

Table E.36. Manage project requirements in OpenReq prototype

Attribute	Definition
ID	Epic 321
Title	Manage project requirements in OpenReq prototype
Туре	Functional
Goal	
Priority	Medium



Related with Work	WP6(T6.2),	WP7(T7.1-Vogella,	T7.2-Qt,	T7.3-
Package Tasks	Siemens, T7.	4-WindTre)		

Table E.37. Preference information of other OpenReq prototype users

Attribute	Definition
ID	Epic 322
Title	Preference information of other OpenReq prototype users
Туре	Functional
Goal	
Priority	Low
Related with Work Package Tasks	WP6(T6.2), WP7(T7.1-Vogella, T7.2-Qt, T7.3- Siemens, T7.4-WindTre)

Table E.38. Recommend solution for inconsistencies of group user preferences

Attribute	Definition
ID	Epic 323
Title	Recommend solution for inconsistencies of group user preferences
Туре	Functional
Goal	Stakeholder wants to get a group recommendation regarding how to resolve inconsistencies of user preferences with respect to releases
Priority	Low
Related with Work Package Tasks	WP4(T4.4)

Table E.39. Visualize an interactive release plan

Attribute	Definition
ID	Epic 324
Title	Visualize an interactive release plan
Туре	Functional
Goal	Stakeholder wants to have an interactive visualization of a release plan
Priority	Low
Related with Work Package Tasks	WP4(T4.6)



Attribute	Definition
ID	Epic 327
Title	Analyze explicit feedback from development tools
Туре	Functional
Goal	Stakeholder as a developer or release manager wants to analyze textual data from development tools in order to get assistance about a requirement
Priority	High
Related with Work Package Tasks	WP2(T2.2,T2.4), WP7(T7.1-Vogella, T7.2-Qt)

 Table E.40. Analyze explicit feedback from development tools

Attribute	Definition	
ID	Epic 330	
Title	Adhere to external relationships	
Туре	Functional	
Goal	Stakeholder as a product owner wants to have the product development to adhere to standards in order to stay compliant.	
	System and its requirements have relationships to external entities such as standards, technologies or external stakeholders. One example concrete example is C++ standard and its different versions. The focus here is to maintain relationship to such an external relationship.	
Priority	Low	
Related with Work Package Tasks	WP5(T5.3,T5.5), WP7(T7.2-Qt)	

Table E.41. Adhere to external relationships

Table E.42. Identify useful user feedbacl

Attribute	Definition
ID	Epic 332
Title	Identify useful user feedback
Туре	Functional
Goal	Stakeholder wants to see automatically discard useless user feedback in order to get support in creating/updating requirements.



	The user feedback should be distinguished between useful for the requirements intelligence and not (e.g., spam).
Priority	High
Related with Work Package Tasks	WP2(T2.4), WP7(T7.1-Vogella,T7.2-Qt, T7.4- WindTre), WP9(T9.4)

Table E.43. Identify similar feedback

Attribute	Definition
ID	Epic 333
Title	Identify similar feedback
Туре	Functional
Goal	Stakeholder wants to see groups of similar feedback in order to get support in the creation/modification of requirements.
	features).
Priority	High
Related with Work Package Tasks	WP2(T2.4), WP7(T7.1-Vogella,T7.2-Qt, T7.4- WindTre)

Table E.44. Identify user rationale

Attribute	Definition
ID	Epic 334
Title	Identify user rationale
Туре	Functional
Goal	Stakeholder wants to see the rationale of the user feedback in order to support the creation/modification of requirements.
	The rationale behind the user feedback should be automatically mined, when possible.
	Rationale may represent another attribute that can be used for other tasks (e.g., clustering)
Priority	Medium
Related with Work Package Tasks	WP2(T2.4), WP7(T7.1-Vogella,T7.2-Qt, T7.4- WindTre)



Table E.45. Repair conflicts in relationships





ANNEX F. USER STORIES DETAILED LIST

Table F.1. Extract context data from Bugzilla

Attribute	Definition
ID	User Story 26
Title	Extract context data from Bugzilla
Туре	Functional
Goal	Stakeholder wants to extract context data from Bugzilla in order to analyze requirements data based on user data.
Related with Epics	Epic 29, Epic 68, Epic 69, Epic 70, Epic 72
Links	User Story 36

Table F.2. See the data in a table

Attribute	Definition
ID	User Story 34
Title	See the data in a table
Туре	Functional
Goal	All relevant data should be displayed in a table in order to see row data.
Links	Epic 33

Table F.3. See data in charts

Definition
User Story 35
See data in charts
Functional
See data in charts in order to analyse requirements data.
Epic 33

Table F.4. Be able to connect different issue tracking systems with openReq

Attribute	Definition
ID	User Story 36
Title	Be able to connect different issue tracking systems with openReq
Туре	Functional



Goal	Be able to connect different issue tracking systems with openReq in order to analyze requirements data from different systems.
	Charts should show the data.
Related with Epics	Epic 29
Links	User Story 25, User Story 26

Table F.5. Be notified when the priority of a task is changed while implementing it

Attribute	Definition
ID	User Story 38
Title	Be notified when the priority of a task is changed while implementing it
Туре	Functional
Goal	Be notified when the priority of a task is changed while implementing it in order to react to changes. Notification is sent when there is a priority change of the task.
Links	Epic 37

Table F.6. Be notified when new requirements with a predefined priority are created

Attribute	Definition
ID	User Story 39
Title	Be notified when new requirements with a predefined priority are created
Туре	Functional
Goal	Be notified when new requirements with a predefined priority are created in order to react. Receive notifications considering user constraints (priority and new reqs.).
Links	Epic 37

Table F.7. Integrate openReq in Eclipse with Mylyn

Attribute	Definition
ID	User Story 40
Title	Integrate openReq in Eclipse with Mylyn
Туре	Functional
Goal	Integrate openReq in Eclipse with Mylyn in order to see openReq data in Eclipse.
	OpenReq data is displayed in Eclipse.



Links	Epic 37, Epic 73
	Table F.8. See openReq in Web UI
Attribute	Definition
ID	User Story 42
Title	See openReq in Web UI
Туре	Functional
Goal	See openReq in Web UI in order to be platform independent
	Easier access to data from multiple devices and OS.
Links	Epic 33

Table F.9. Click on the "Last project's trend" button

Attribute	Definition
ID	User Story 64
Title	Click on the "Last project's trend" button
Туре	Functional
Goal	Stakeholder wants to click on the "Last project's trend" button in order to visualize the analytics related to the project. A linechart is plotted.
Links	Epic 63

Table F.10. Send direct invitation via e-mail instead of the poll

Attribute	Definition
ID	User Story 135
Title	Send direct invitation via e-mail instead of the poll
Туре	Functional
Goal	Stakeholder as a decision poll creator wants to send direct invitation via e-mail instead of the poll in order to invite people to participate in a decision pool.
Links	

Table F.11. Send invitations to predefined user groups

Attribute	Definition
ID	User Story 136
Title	Send direct invitation via e-mail instead of the poll
Туре	Functional



Goal	Stakeholder as a decision poll creator wants to send invitations to predefined user groups in order to invite a specific group of people.
Links	

Table F.12. Allow comments

Attribute	Definition
ID	User Story 138
Title	Allow comments
Туре	Functional
Goal	Stakeholder as a decision poll creator wants to allow comments in order to defend decisions made by the participants.
Links	

Table F.13. Know when two requirements are talking about the same

Attribute	Definition
ID	User Story 147
Title	Know when two requirements are talking about the same
Туре	Functional
Goal	Stakeholder as a member of the Qt community wants to know when two requirements are talking about the same in order to not create a requirement that already exists.
Links	Epic 305

Table F.14. Know when two requirements are talking about the same

Attribute	Definition
ID	User Story 148
Title	Know when two requirements are talking about the same
Туре	Functional
Goal	Stakeholder as a Qt employee wants to know when two requirements are talking about the same in order to consolidate existing requirements.
Links	Epic 305



Attribute	Definition
ID	User Story 149
Title	Identify the bug reports related to the comment I am entering
Туре	Functional
Goal	Stakeholder as a member of the Qt community wants to identify the bug reports related to the comment I am entering in order to ease the task of creating a comment.
Links	Epic 141

Table F.15. Identify the bug reports related to the comment I am entering

Table F.16. Be informed of the information missing in a bug

Attribute	Definition
ID	User Story 150
Title	Be informed of the information missing in a bug
Туре	Functional
Goal	Stakeholder as a member of the Qt community wants to be informed of the information missing in a bug in order to provide all the information needed to be able to resolve it.
Links	Epic 142

Table F.17. Be recommended of the possible category of the requirement I am creating

Attribute	Definition
ID	User Story 151
Title	Be recommended of the possible category of the requirement I am creating
Туре	Functional
Goal	Stakeholder as a member of the Qt community wants to be recommended of the possible category of the requirement I am creating in order to ease and improve the task of creating a requirement.
Links	Epic 143



Table F.18. Be recommended of the possible component of the requirement I am
creating

Attribute	Definition
ID	User Story 152
Title	Be recommended of the possible component of the requirement I am creating
Туре	Functional
Goal	Stakeholder as a member of the Qt community wants to be recommended of the possible component of the requirement I am creating in order to ease and improve the task of creating a requirement.
Links	Epic 143

 Table F.19. Be recommended of the possible environment of the requirement I am creating

Attribute	Definition
ID	User Story 153
Title	Be recommended of the possible environment of the requirement I am creating
Туре	Functional
Goal	Stakeholder as a member of the Qt community wants to be recommended of the possible environment of the requirement I am creating in order to ease and improve the task of creating a requirement.
Links	Epic 143

Table F.20. Be asked questions related to the priority of the requirement I am creating

Attribute	Definition
ID	User Story 154
Title	Be asked questions related to the priority of the requirement I am creating
Туре	Functional
Goal	Stakeholder as a member of the Qt community wants to be asked questions related to the priority of the requirement I am creating in order to properly assign the priority of the requirement.
Links	Epic 143

Table F.21. Know the explanation of the priority assigned to a requirement created

Attribute	Definition		
ID	User Story 155		



Title	Know the explanation of the priority assigned to a requirement created
Туре	Functional
Goal	Stakeholder as a member of the Qt community wants to know the explanation of the priority assigned to a requirement created in order to trust the system.
Links	Epic 143

Table F.22. Automatically assign an assignee of a requirement that has been created

Attribute	Definition
ID	User Story 156
Title	Automatically assign an assignee of a requirement that has been created
Туре	Functional
Goal	Stakeholder as a Qt employee wants the system to automatically assign an assignee of a requirement that has been created in order to ease the assignee process and be able to act faster with the person that created the requirement.
Links	Epic 144

 Table F.23. Automatically differentiate the requirements contained in the same paragraph

Attribute	Definition
ID	User Story 161
Title	Automatically differentiate the requirements contained in the same paragraph
Туре	Functional
Goal	Stakeholder as a Siemens employee wants the system to automatically differentiate the requirements contained in the same paragraph in order to ease and improve the requirements identification process.
Links	Epic 173



Table F.24. Identify if there was problems with a similar requirements to the one I amdealing with at the moment

Attribute	Definition
ID	User Story 163
Title	Identify if there was problems with a similar requirements to the one I am dealing with at the moment
Туре	Functional
Goal	Stakeholder as a Siemens employee to identify if there was problems with a similar requirements to the one I am dealing with at the moment in order to avoid future problems.
Links	Epic 308

Table F.25. Be recommended of the possible risk of the requirement I am in charge of

Attribute	Definition
ID	User Story 167
Title	Be recommended of the possible risk of the requirement I am in charge of
Туре	Functional
Goal	Stakeholder as a Siemens employee wants to be recommended of the possible risk of the requirement I am in charge of in order to ease and improve the requirements processing.
Links	Epic 143

Table F.26. Tell the system that a recommendation is wrong

Attribute	Definition
ID	User Story 170
Title	Tell the system that a recommendation is wrong
Туре	Functional
Goal	Stakeholder as a requirements manager wants to tell the system that a recommendation is wrong in order to teach the system and improve the recommendations in the future.
Links	Epic 296



Attribute	Definition
ID	User Story 174
Title	Extract business requirements from user requests
Туре	Functional
Goal	Stakeholder as a Vogella employee wants to extract business requirements from user requests in order to create requirements that are important for users.
Links	Epic 178

 Table F.27. Extract business requirements from user requests

Table F.28. Identify bad quality requirements

Attribute	Definition
ID	User Story 175
Title	Identify bad quality requirements
Туре	Functional
Goal	Stakeholder as a Vogella employee wants identify bad quality requirements in order to consolidate existing requirements.
Links	Epic 158

Table F.29. Identify outdated requirements

Attribute	Definition
ID	User Story 176
Title	Identify outdated requirements
Туре	Functional
Goal	Stakeholder as a Vogella employee wants identify outdated requirements in order to consolidate existing requirements.
Links	Epic 171

Table F.30. Be recommended of the possible classification of the requirements created

Attribute	Definition
ID	User Story 177
Title	Be recommended of the possible classification of the requirements created
Туре	Functional
Goal	Stakeholder as a Vogella employee wants to be recommended of the possible classification of the



	requirements created in order to ease and improve the task of creating a requirement.
Links	Epic 173

Table F.31. Extract business requirements from user requests from tweets andFacebook posts

Attribute	Definition
ID	User Story 182
Title	Extract business requirements from user requests from tweets and Facebook posts
Туре	Functional
Goal	Stakeholder as a WindTree employee wants to extract business requirements from user requests (e.g., feature requests) from tweets and Facebook posts in order to create requirements that are important for users.
Links	Epic 178

Table F.32. Recommend prioritization for requirements identified from the social media by using the information in the social media to do so

Attribute	Definition
ID	User Story 183
Title	Recommend prioritization for requirements identified from the social media by using the information in the social media to do so
Туре	Functional
Goal	Stakeholder as a WindTree employee wants to be recommended of prioritizations for requirements identified from the social media by using the information in the social media to do so in order to increase the satisfaction of users.
Links	Epic 143

Table F.33. Visualize a summary of what users are commenting on social media

Attribute	Definition
ID	User Story 186
Title	Visualize a summary of what users are commenting on social media
Туре	Functional



Goal	Stakeholder wants to visualize a summary of what users are commenting on social media in order to
	support the prioritization of the requirements.
Links	Epic 185

Table F.34. View external information about requirements

Attribute	Definition
ID	User Story 187
Title	View external information about requirements
Туре	Functional
Goal	Stakeholder wants to view external information about requirements as for example: estimated effort to develop a new requirement, estimated time for ROI, etc. in order to support the prioritization of the requirements.
Links	Epic 185

Table F.35. Have reports on social media data

Attribute	Definition
ID	User Story 188
Title	Have reports on social media data
Туре	Functional
Goal	Stakeholder wants to have reports on social media data in order to have suggestions about new requirements or evaluation about already developed functionalities.
Links	Epic 184

Table F.36. Get feedback from the system whether some piece of text is a requirement or not

Attribute	Definition
ID	User Story 211
Title	Get feedback from the system whether some piece of text is a requirement or not
Туре	Functional
Goal	Stakeholder as a Siemens requirement manager wants to get feedback from the system whether some piece of text is a requirement or not in order to ease and improve the requirements identification process.
Links	Epic 173



Attribute	Definition
ID	User Story 212
Title	Get feedback from the system which of the entries from a list of requirement candidates is a requirement or not
Туре	Functional
Goal	Stakeholder as a Siemens requirement manager wants to get feedback from the system which of the entries from a list of requirement candidates is a requirement or not in order to ease and improve the requirements identification process.
Links	Epic 173

Table F.37. Get feedback from the system which of the entries from a list of	of
requirement candidates is a requirement or not	

Table F.38. Extract requirements candidates from a Microsoft word document

Attribute	Definition
ID	User Story 213
Title	Extract requirements candidates from a Microsoft word document
Туре	Functional
Goal	Stakeholder as a requirement manager wants to extract requirements candidates from a Microsoft word document.
Links	Epic 189

 Table F.39. Get a recommendation for an assignment of stakeholders roles (domains) to a requirement

Attribute	Definition
ID	User Story 214
Title	Get a recommendation for an assignment of stakeholders roles (domains) to a requirement
Туре	Functional
Goal	Stakeholder as a Siemens requirement manager wants to get a recommendation for an assignment of stakeholders roles (domains) to a requirement in order to reduce click & search effort.
Links	Epic 143



Table F.40. Get a recommendation which requirements are similar to a reference
requirement

Attribute	Definition
ID	User Story 215
Title	Get a recommendation which requirements are similar to a reference requirement
Туре	Functional
Goal	Stakeholder as a Siemens requirement manager wants to get a recommendation which requirements are similar to a reference requirement in order to reuse decision knowledge from past projects
Links	Epic 305

Table F.41. Get a recommendation regarding redundant requirements

Attribute	Definition
ID	User Story 216
Title	Get a recommendation regarding redundant requirements
Туре	Functional
Goal	Stakeholder wants to get a recommendation regarding redundant requirements in order to find redundancies and inconsistencies.
Links	Epic 215

 Table F.42. Get a recommendation regarding a technical solution approaches to fulfil a requirement

Attribute	Definition
ID	User Story 217
Title	Get a recommendation regarding a technical solution approaches to fulfil a requirement
Туре	Functional
Goal	Stakeholder wants to get a recommendation regarding a technical solution approaches to fulfil a requirement in order to ease and improve the requirements processing.
Links	Epic 143



Attribute	Definition
ID	User Story 218
Title	Customize "semantic thesaurus" for the domain
Туре	Functional
Goal	Stakeholder as a requirement manager wants to customize "semantic thesaurus" for the domain (add and delete the stakeholder roles and solution approaches for the project).
Links	Epic 277

Table F.43. Customize "semantic thesaurus" for the domain

Table F.44. Add a	"semantic thesaurus"	' for the domain
-------------------	----------------------	------------------

Attribute	Definition
ID	User Story 219
Title	Add a "semantic thesaurus" for the domain
Туре	Functional
Goal	Stakeholder as a requirement manager wants to add a "semantic thesaurus" for the domain
Links	Epic 218

Table F.45. Rate a requirement as compliant, conditionally compliant or none compliant

Attribute	Definition
ID	User Story 221
Title	Rate a requirement as compliant, conditionally compliant or none compliant
Туре	Functional
Goal	Stakeholder wants to rate a requirement as compliant, conditionally compliant or none compliant.
Links	Epic 312

Table F.46. Get a recommendation regarding how to best resolve the inconsistencies of compliance ratings

Attribute	Definition
ID	User Story 223
Title	Get a recommendation regarding how to best resolve the inconsistencies of compliance ratings



Туре	Functional
Goal	Stakeholder as a bid manager wants to get a recommendation regarding how to best resolve the inconsistencies of compliance ratings.
Links	Epic 222

Table F.47. Create an user account

Attribute	Definition
ID	User Story 225
Title	Create an user account
Туре	Functional
Goal	Stakeholder wants to create an user account.
Links	Epic 315

Table F.48. Change the password

Attribute	Definition
ID	User Story 226
Title	Change the password
Туре	Functional
Goal	Stakeholder wants to change the password.
Links	Epic 315

Table F.49. Reset the password

Attribute	Definition
ID	User Story 227
Title	Reset the password
Туре	Functional
Goal	Stakeholder wants to reset the password.
Links	Epic 315

Table F.50. Upload a profile image

Attribute	Definition
ID	User Story 228
Title	Upload a profile image
Туре	Functional
Goal	Stakeholder wants to upload a profile image to her account.
Links	Epic 315



Attribute	Definition
ID	User Story 229
Title	Give feedback to the OpenReq community
Туре	Functional
Goal	Stakeholder wants to give feedback to the OpenReq community via eMail.
Links	Epic 316

Table F.51. Give feedback to the OpenReq community

Table F.52. Update the user profile

Attribute	Definition
ID	User Story 230
Title	Update the user profile
Туре	Functional
Goal	Stakeholder wants to update her user profile (username, profile image)
Links	Epic 315

Table F.53. Get help in using the system the first time

Attribute	Definition
ID	User Story 231
Title	Get help in using the system the first time
Туре	Functional
Goal	Stakeholder wants to get help in using the system the first time.
Links	Epic 316

Table F.54. Create a project

Attribute	Definition
ID	User Story 232
Title	Create a project
Туре	Functional
Goal	Stakeholder as a project administrator wants to create a project.
Links	Epic 317



Table	F.55.	Add	membe	rs to	ุล เ	project
Table	I .JJ.	Auu	memor	15 10	a	μυία

Attribute	Definition
ID	User Story 233
Title	Add members to a project
Туре	Functional
Goal	Stakeholder as a project administrator wants to add members (stakeholders) to a project.
Links	Epic 317

Table F.56. Modify project details

Attribute	Definition
ID	User Story 234
Title	Modify project details
Туре	Functional
Goal	Stakeholder as a project administrator wants to modify project details (description, members).
Links	Epic 317

Table F.57. Insert the releases of a project

Attribute	Definition
ID	User Story 235
Title	Insert the releases of a project
Туре	Functional
Goal	Stakeholder as a project administrator wants to insert the releases of a project.
Links	Epic 318

Table F.58. Export project information to pdf

Attribute	Definition
ID	User Story 236
Title	Export project information to pdf
Туре	Functional
Goal	Stakeholder as a project administrator wants to export project information to pdf.
Links	Epic 317



Table F.59. Delete a project

Attribute	Definition
ID	User Story 237
Title	Delete a project
Туре	Functional
Goal	Stakeholder as a project administrator wants to delete a project.
Links	Epic 317

Table F.60. Add requirements to a certain project

Attribute	Definition
ID	User Story 238
Title	Add requirements to a certain project
Туре	Functional
Goal	Stakeholder wants to add requirements to a certain project.
Links	Epic 317

Table F.61. Add and modify a description of a requirement

Attribute	Definition
ID	User Story 239
Title	Add and modify a description of a requirement
Туре	Functional
Goal	Stakeholder wants to add and modify a description of a requirement.
Links	Epic 321

Table F.62. Search for requirements and sort among releases and/or name

Attribute	Definition
ID	User Story 240
Title	Search for requirements and sort among releases and/or name
Туре	Functional
Goal	Stakeholder wants to search for requirements and sort among releases and/or name.
Links	Epic 321



Attribute	Definition
ID	User Story 241
Title	Prioritize the requirements of a project
Туре	Functional
Goal	Stakeholder wants to prioritize the requirements of a project.
Links	Epic 321

Table F.63. Prioritize the requirements of a project

Table F.64. Add and modify a metadata of a requirement

Attribute	Definition
ID	User Story 242
Title	Add and modify a metadata of a requirement
Туре	Functional
Goal	Stakeholder wants to add and modify a metadata (effort, risk, profit, cost information) of a requirement.
Links	Epic 312

Table F.65. Add and modify attachments of a requirement

Attribute	Definition
ID	User Story 243
Title	Add and modify attachments of a requirement
Туре	Functional
Goal	Stakeholder wants to add and modify attachments of a requirement.
Links	Epic 312

Table F.66. Add dependencies between requirements

Attribute	Definition
ID	User Story 244
Title	Add dependencies between requirements
Туре	Functional
Goal	Stakeholder wants to add dependencies between requirements.
Links	Epic 312, Epic 278



Attribute	Definition
ID	User Story 245
Title	Delete requirements from a project
Туре	Functional
Goal	Stakeholder wants to delete requirements from a project.
Links	Epic 321

Table F.67. Delete requirements from a project

 Table F.68. Delete requirements from a project

Attribute	Definition
ID	User Story 246
Title	Delete requirements from a project
Туре	Functional
Goal	Stakeholder wants to delete requirements from a project.
Links	Epic 321

Table F.69. Select the current project

Attribute	Definition
ID	User Story 247
Title	Select the current project
Туре	Functional
Goal	Stakeholder wants to select the current project.
Links	Epic 317

Table F.70. Get an overview of the current project

Attribute	Definition
ID	User Story 249
Title	Get an overview of the current project
Туре	Functional
Goal	Stakeholder wants to get an overview of the current project (requirements, dependencies and releases).
Links	Epic 317



Attribute	Definition
ID	User Story 250
Title	Get information about open issues
Туре	Functional
Goal	Stakeholder wants to get information about open issues (need for closure).
Links	Epic 317

Table F.71. Get information about open issues

Table F.72. Get information about inconsistent preferences with respect to the release plans

	L L L L L L L L L L L L L L L L L L L
Attribute	Definition
ID	User Story 251
Title	get information about inconsistent preferences with respect to the release plans
Туре	Functional
Goal	Stakeholder wants to get information about inconsistent preferences with respect to the release plans.
Links	Epic 322

Table F.73. Have a chat possibility per requirement

Attribute	Definition
ID	User Story 255
Title	Have a chat possibility per requirement
Туре	Functional
Goal	Stakeholder wants to have a chat possibility per requirement.
Links	Epic 321

Table F.74. See the preferences and opinions of other users

Attribute	Definition
ID	User Story 258
Title	See the preferences and opinions of other users
Туре	Functional
Goal	Stakeholder wants to see the preferences and opinions of other users.
Links	Epic 322



Attribute	Definition
ID	User Story 306
Title	Be notified of the requirements assigned
Туре	Functional
Goal	Stakeholder as a Qt requirements assignee wants to be notified of the requirements that have been assigned to him without being interrupted all the time in order to act faster with the people that created these requirements.
Links	Epic 144

Table F.75. Be notified of the requirements assigned

Table F.76. Measure the quality of a requirement

Attribute	Definition
ID	User Story 307
Title	Measure the quality of a requirement
Туре	Functional
Goal	Stakeholder as a Siemens employee wants to measure the quality of a requirement in order to detect in a new project if this requirement is going to need clarifications
Links	Epic 158

Table F.77. Tell the system when I am having problems with a requirement and the
reason for the problem

Attribute	Definition
ID	User Story 308
Title	Tell the system when I am having problems with a requirement and the reason for the problem
Туре	Functional
Goal	Stakeholder as a Siemens employee wants to tell the system when I am having problems with a requirement and the reason for the problem in order to be notified in the future of the these complications when I am dealing with similar requirements.
Links	Epic 163



Attribute	Definition
ID	User Story 309
Title	Receive lists of prioritised recommendations
Туре	Functional
Goal	Stakeholder wants to receive lists of prioritised recommendations in order to be able to evaluate the value of the recommendations.
Links	Epic 141, Epic 143, Epic 144, Epic 157, Epic 171, Epic 178, Epic 305

 Table F.78. Receive lists of prioritised recommendations

Table F.79. Recognize important requirements for commercial customers

Attribute	Definition
ID	User Story 310
Title	Recognize important requirements for commercial customers
Туре	Functional
Goal	Stakeholder as a Qt employee wants to recognize important requirements for commercial customers in order to be able to evaluate, for the modules that are not actively maintained by the community, the interests of the commercial partners.
Links	Epic 178

Table F.80. Get a recommendation regarding conflicting requirements

Attribute	Definition
ID	User Story 311
Title	Get a recommendation regarding conflicting requirements
Туре	Functional
Goal	Stakeholder wants to get a recommendation regarding conflicting requirements in order to find redundancies and inconsistencies.
Links	Epic 288

Table F.81. Modify the releases of a project

Attribute	Definition
ID	User Story 319
Title	Modify the releases of a project
Туре	Functional



Goal	Stakeholder as a project administrator wants to modify the releases of a project.
Links	Epic 318

Table F.82. Delete the releases of a project

Attribute	Definition
ID	User Story 320
Title	Delete the releases of a project
Туре	Functional
Goal	Stakeholder as a project administrator wants to delete the releases of a project.
Links	Epic 318

Table F.83. See ambiguous requirement text

Attribute	Definition
ID	User Story 337
Title	See ambiguous requirement text
Туре	Functional
Goal	Stakeholder wants to see ambiguous requirement text.
Links	Epic 172

Table F.84. See words not included in the project glossary

Attribute	Definition
ID	User Story 337
Title	See words not included in the project glossary
Туре	Functional
Goal	Stakeholder wants to see words not included in the project glossary in order to correct them.
Links	Epic 172

Table F.85. See text expressing uncertainty in requirement specification

Attribute	Definition
ID	User Story 339
Title	See text expressing uncertainty in requirement specification
Туре	Functional


Goal	Stakeholder wants to see text expressing uncertainty in requirement specification in order to understand the level of uncertainty associated to it.
Links	Epic 172

 Table F.86. See the violation to a specific requirement template

Attribute	Definition
ID	User Story 340
Title	See the violation to a specific requirement template
Туре	Functional
Goal	Stakeholder wants to see the violation to a specific requirement template in order to correct them.
Links	Epic 172

Table F.87. Delete recommended assignment and/or propose another domain

Attribute	Definition
ID	User Story 273
Title	Delete recommended assignment and/or propose another domain
Туре	Functional
Goal	Stakeholder as a domain exert wants to delete recommended assignment and/or propose another domain in order to make domain assignments better (but need to avoid ping-pong effects)
Links	Epic 214

Table F.88. Extract Bugzilla requirement data

Attribute	Definition
ID	User Story 25
Title	Extract Bugzilla requirement data
Туре	Functional
Goal	Stakeholder as a developer wants to extract Bugzilla requirement data in order to analyze requirements data.
Links	Epic 327

Table F.89. Assess the sentiment of social media

Attribute	Definition
ID	User Story 325
Title	Assess the sentiment of social media



Туре	Functional
Goal	Stakeholder as a requirement engineer wants to assess the sentiment of social media in order to validate a new requirement.
Links	Epic 184

Table F.90. Obtain information from Gerrit

Attribute	Definition
ID	User Story 326
Title	Obtain information from Gerrit
Туре	Functional
Goal	Stakeholder wants to obtain information from Gerrit in order to take a decision on a requirement.
Links	Epic 327

Table F.91. Obtain information from SonarQube

Attribute	Definition
ID	User Story 328
Title	Obtain information from SonarQube
Туре	Functional
Goal	Stakeholder wants to obtain information from SonarQube in order to take a decision on a requirement.
Links	Epic 327

Table F.92. Cluster frequent errors of the Eclipse component

Attribute	Definition
ID	User Story 329
Title	Cluster frequent errors of the Eclipse component
Туре	Functional
Goal	Stakeholder as a developer wants to cluster frequent errors of the Eclipse component.
Links	Epic 206

Table F.93. Classify English Twitter messages

Attribute	Definition
ID	User Story 342
Title	Classify English Twitter messages
Туре	Functional



Goal	Stakeholder as a requirement engineer wants to classify English Twitter messages in order to suggest requirements.
Links	Epic 184

 Table F.94. Classify Italian Twitter messages

Attribute	Definition
ID	User Story 343
Title	Classify Italian Twitter messages
Туре	Functional
Goal	Stakeholder as a requirement engineer wants to classify Italian Twitter messages in order to suggest requirements.
Links	Epic 184



ANNEX G. DOMAIN MODELS

This annex includes the initial versions of the domain models of the trial partners and also the domain model of OpenReq. These domains models are going to evolve and the final version will be included as the OpenReq ontology in deliverable D5.3.

Wind Tree Domain Model



Next the main concepts in the domain model are described.

Feedback is the main entity which will be used to propose requirements (e.g., instances of the appropriate Requirement entity in other WP ontologies). Feedback (or sets of Feedback) is augmented with meta information derived by the requirements intelligence component in WP2 (e.g., the topic of the feedback, whether it is a feature request or bugfix).



The Author reputation and status are calculated properties which can be used to weight the importance of his/her feedback.

It has to be decided if Usage is a Feedback (in which case some properties of feedback, like Text, do not make sense) or Usage contributes to Feedback (e.g., it represents its context.).

Importance is calculated (e.g., priority + Author's reputation).

Flow is an array of status through which the issue passed through.

Comment rating is dependent on the source. For source == Twitter, popularity can be #retweets; for source == Jira it could be upvotes. Comment popularity is a calculated property. For example: Comment author popularity + number rating.

Action and Event represent recordings of what happens in a device. Action represent what the user does (e.g., button pressed in View xyz), Event capture outside events (e.g., network down, received message).

The requirement file representation should match the one of the other ontologies.

Context represents the context in which Action/Event take place and depends on the device.

Content can be a label associated to a specific sequence of Actions/Events in a given Context.

Call Detail Record represents the typical information registered by a network antenna. It can be intended as implicit feedback for the telco case.



Siemens Domain Model



Next the main concepts in the domain model are described.

Domain: Domains correspond to the stakeholders' roles/departments.

Requirement: In Siemens a requirement is classified as DEF (meaning that it is really a Requirement), Prose or Not Classified. The requirements are extracted from a document.

Document: A document is provided by a customer in the context of a Project. The requirements are extracted from a document.

ProjectDomain: This class represents the participation of a stakeholder in a project, in the context of a Siemens-Domain, in a Project.

Stakeholder: A stakeholder of a project is a person or an organization that has a (direct or indirect) influence on the requirements of the project.

RequirementDomainAssignment: This class represents the assignment of a requirement (in fact it is a requirement candidate) to a project domain. E.g. if a requirement has a domain assignment to a specific project domain, the stakeholder of this project domain must provide an answer for that requirement concerning compliance etc. considered for being implemented.



Answer: Requirements that are classified of type DEF (meaning that they are really a requirement, so not prose) are assessed to be Compliant, CompliantWithComment or NotCompliant, and are assessed to be implemented by means of an approach that can be by using an ExistingProduct or by Developing it. The answer indicates the result of the assessment, done by a stakeholder in the context of a domain and a project.

Vogella Domain Model



Next the main concepts in the domain model are described.

Bug: A bug in Vogella is a type of requirement that may correspond to a requirement, an actual bug or a feature request.

Comment: A bug can have zero or more comments defined.

Attachment: A bug may have zero or more attached documents.

Qt Domain Model

In the case of Qt no domain model was defined since the trial considered that their model fits with the OpenReq domain model included in the next section.



OpenReq Domain Model



Next the main concepts in the OpenReq domain model are described.

Attachment: An attachment is a computer file related with a requirement to give extra data about it.

Comment: A comment can be a remark giving an opinion about a requirement.

Dependency: A dependency is any kind of relationship between requirements. Different types of dependencies may be conflicts, excludes, is part of, etc.

Decomposition: A decomposition is a kind of relationship among requirements. One requirement may be decomposed into several requirements.

Priority: The priority corresponds to the importance of a requirement in comparison to other requirements according to given criteria.

Project: A project is a set of interrelated activities, carefully planned usually by a project team, to be executed over a fixed period of time and within certain cost and other limitations of resources, to implement a certain software system.

Release: A release is a configuration of a software system that has been prepared for installation and use by customers.

NL-Requirement: A type of requirement that specifies in natural language a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.

Responsibility: The responsibility of a Team Member in different NL-Requirements. One requirement may have different team members involved.



Role: A Team Member may participate with different roles in a Project.

Status: Progress of acceptance and implementation of a requirement in a Project.

Team Member: A person that participates in some way in the definition of requirements of a project.



ANNEX H. GLOSSARY

Term	Definition	Trial	Owner
Demand	Team in charge of network-related requirements	Wind Tre	Davide Fucci
Dependency	Generally any kind of relationship between requirements. The term is used, at least sometimes, interchangeably with, e.g., relationship. A more details ontology of dependencies shall be defined that could include conflicts, excludes, is part of etc.	All	Mikko Raatikainen
Domain	Where the platform is used (i.e., mobile, automotive.	Qt	Davide Fucci
Project	A specific component developed within the company (i.e., Bluetooth, UI).	Qt	Davide Fucci
Eclipse Foundation	The non-profit foundation that sponsors, among other projects the Eclipse IDE.	Eclipse	Davide Fucci
Member of the Qt community	Any user of the Qt community that is not a Qt employee.	Qt	Cristina Palomares
Ontology	An ontology covers the key concepts (and their properties) that can be represented as "boxes", and relationships "named links between boxes" in order to define a shared understanding what we are talking about when are talking about requirements engineering. Ontology can also be defined more rigorously with formal semantics but inthis context we use a broad meaning of ontology covering both loosely and strictly defined ontology. Ontology can be described by various means but in <u>OpenReq?</u> no specific means has been specified. Related and similar terms: Conceptualization and meta-model.	Siemens, Qt	Mikko Raatikainen
Requirement	Global term to name the different type of assets that are used in a requirements management system: bug, epic, user story, etc.	Qt, Siemens, Vogella, Wind Tre	Cristina Palomares