Grant Agreement nº	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)
Topic	ICT-10-16 Software Technologies
Start date of project	January 1st, 2017
Duration	36 months

D7.1 - Trials and Evaluation Plan

Lead contractor: SIEMENS

Author(s): SIEMENS, Qt, WIND TRE

Submission date: October 2017

Dissemination level: PU

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

Abstract: A brief summary of the purpose and content of the deliverable.

This is deliverable (D7.1) of the OpenReq project as defined in the Grant Agreement: a report describing the overall evaluation strategy and instruments, a time and resource plan, with identified roles and actions per project partner. After the initial release of this document, and given the wide period of time covered by the trials, additional intermediate and adjusted versions for internal project purposes will be created on demand.

The overall evaluation objective is to cover all functionality (microservices) supplied by work packages WP2 - WP5 of the scientific partners. As the use cases of the industrial partners are quite diverse, not every microservice is relevant for each use case. Therefore, not every microservice will be evaluated in each trial, but all microservices are covered in at least one trial. As a strategy we choose a divide-and-conquer approach: The trials are planned, executed, and evaluated independently by each involved partner. They use those instruments which are best suited for their use-case: regression test tools, field studies, questionnaires, etc. The conjoint methodology follows roughly the ideas of the ISO-25000 standards family although we favor a pragmatic approach over a strict formalism. The results of the evaluation will be documented in Trial Reports by each involved partner independently (deliverables D7.2-4) and summarized in the conjoint Evaluation Report (deliverable D7.5) at the end of the project.

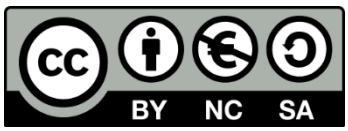

Table of Contents

List of Tables

List of Abbreviations

API	Application Programming Interface
BPM	Bid Project Manager
DoA	Description of Action
ETCS	European Train Control System
FN	False Negative
FP	False Positive
GA	Grant Agreement
RFP	Request For Proposal
RM	Requirements Management
RMiP	Requirements Manager in Project
SCADA	Supervisory Control And Data Acquisition
SM	System Manager
TN	True Negative
TP	True Positive
WP	Work package

Related Documents and References

GA	Grant Agreement Nr. 732463 OPENREQ, Annex A _Description of the Action_
ISO-25000	ISO/IEC 25000:2014 "Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE", 2014
Peffers2007	K.Peffers, et al.: A design science research methodology for information systems research, Journal of management information systems 24/3, 2007
Seaman1999	C.B.Seaman: Qualitative methods in empirical studies of software engineering, IEEE Transactions on Software Engineering 25/4, 1999.
Sokolova2009	M.Sokolova, G.Lapalme: A systematic analysis of performance measures for classification tasks, Information Processing and Management 45, 2009
Wohlin2012	C.Wohlin, P.Runeson, M.Höst, M.C.Ohlsson, and B.Regnell: Experimentation in software engineering, 2012

1 STRUCTURE OF THE DOCUMENT

Table of contents, abbreviations and a list of related or referenced documents are available above. This **Section 1** describes the structure of the document and is followed by **Section 2** containing information about the scientific methods and metrics to be selectively used at the trials. Then **Sections 3 to 5** describe the detailed plans for each trial. The last two **Sections 6 and 7** define the overall timeline and coverage of project objectives of all trials as well as open topics.

The trials will be executed over a period of at least one year in order to carry several iterations including each time new OpenReq functions as they are developed and fine-tuned by users' feedback.

The specification of each trial in Sections 3-5 follows the same structure:

- Overview
  - Problem description (as-is state)
  - Envisioned improvements (by OpenReq services)
  - Evaluation strategy and instruments.

  Scope
  - Covered features/services of the OpenReq framework
  - Used data sources (collected, generated, user groups, etc.)
  - Necessary evaluation activities (interviews, experiments, post-mortem analyses, etc.)
  - Participants, as preliminary planned (number of participants, role, experience, sampling)
- Metrics
  - Expected impact (efficiency, satisfaction, usefulness, completeness, etc.)
  - Selected metrics (what and how to measure? why?)
  - Aggregation of results (from raw to refined/condensed, value ranges)
  - Success criteria (at which measured results is impact achieved?)
- Evaluation procedure
  - Requirements for performing it (software available, access rights, etc.)
  - Archiving the input data (internally, for public re-use…)
  - Sequence of the necessary steps
  - Archiving the results (internally, for public review…)
  - Documentation and dissemination of the results
- Time and resource plan
  - List of tasks, incl. involved persons and estimated duration (e.g. interview with user group NN, deployment of the OpenReq platform at evaluation site, assessment of evaluation results, creation of evaluation report, scientific publication of the results)
  - Contacts of persons of interest (e.g., company's champions, people who can help us to set up the environment, recruiting within specific units)
  - Timeline (schedule, milestones, etc.)

The results of the evaluation will be documented in evaluation reports with the following content structure:

- Organizational information: Who did the evaluation? When? Which data were used? Which metrics?
- Results: Raw data, aggregated data, visualization

- Interpretation

Depending on the trial, some details of input data and resulting data may not be disclosed.

2 SCIENTIFIC METHODS

This section contains information about the scientific methods—qualitative (Seaman1999) and quantitative (Wohlin2012)—and metrics which will be used in at least one of the trials. Notice that not all the trials will use the same set of methods as described below.

2.1 Qualitative methods

The qualitative evaluation performed will rely on the judgment of experts in the case companies. The techniques include questionnaires, structured and semi-structured interviews. The qualitative results are used to augment the quantitative results (see Section 2.2). The principal advantage of using qualitative methods is that they help to answer questions about variables that are otherwise difficult to measure such as motivation, perception, and experience. Mainly, two techniques are employed, participants observations and interviews.

Observations capture behaviors and interactions of the participants with the OpenReq platform. However, the parts of the platform underlying the process (i.e., requirements engineering) that can actually be observed are limited because much of the decisions are usually not explicitly made (e.g., through vocalization).

We observe meetings in which the OpenReq platform is discussed (e.g., stakeholders requirements elicitation meetings) and we gather data about the discussed functionalities, technical information, and the impressions the different stakeholders have about the platform. When possible, video and audio recording will be used to support the researcher's observations. Observations must be carried out by two or more researchers in order to evaluate the validity and consistency of data collected.

Another approach to capture such information is to observe the participants by recording their keystrokes and mouse movements as they use the OpenReq platform. This can be complemented by employing the *think aloud* protocol which requires the subject to verbalize her thought process to the observer. As this process can be laborious for the researchers, a variation—referred to as *synchronized shadowing*—can be employed. This approach requires two researchers to watch a participant while performing requirements engineering tasks on the OpenReq platform while she is thinking aloud. Both researchers record different types of information but synchronized to the second. For example, one researcher might concentrate on the participant’s actions (e.g., navigation in the interface), while the other focuses on the participant’s vocalized motivations and strategies. The observations are timestamped to provide a detailed annotations.

Interviews are used to collect opinions or impressions about the OpenReq platform and to help identify issues that the OpenReq project should address. Through interviews, we will elicit how the new RE process is carried out in each trial case and compare it to the planned one. In this regards, we use interviews to clarify what happened while observing the participants when they are using the OpenReq platform.

The results of *structured interviews*—i.e., interviews with specific objectives for the type of information sought after (e.g., experiences in using the OpenReq platform)—can be transformed into quantitative data for easier analysis. Through *semi-structured interviews*, we will allow unforeseen types of information to be recorded. These interviews include a mixture of open-ended and specific questions, designed to elicit not only the information foreseen but also unexpected types of information (e.g., new requirements that should be taken into account in the platform for the next iteration). In semi-structured interviews, the interviewer starts with

a specific set of questions—representing the objective of the interview—but also includes an additional set of questions that are open-ended and intended for soliciting other information.

We will employ several **analysis strategies** to make sense of the qualitative data. One approach to carry out the interviews is the *Delphi method.* After the OpenReq platform is deployed and used by trial participants, we ask them to answer a questionnaire (e.g., about the platform support in achieving the goal of their requirements engineering tasks) and provide their rationale for the answers. Then, the answers are anonymized, summarized, and communicated back to the participants by the researchers. The participants have then the chance to discuss, reflect and revise their answers in the light of everyone else's opinions. Several rounds of refinement take place until consensus or stable results are reached.

To validate the benefits of the OpenReq platform for the trial partners, we use *negative case analysis*. This involves the search for evidence that logically contradicts a proposition on which the OpenReq platform was built. Therefore, such preposition will be revised to cover the negative gathered evidence, so that the new proposition can be checked against the existing data and the one that will be collected during the subsequent iterations. Searching contradictory evidence can include, for example, selecting a new project where to apply the OpenReq approach. This results in increased representativeness and collection of new data to triangulate the findings. For the case of the OpenReq trials, the triangulation consists in gathering different types of evidence (see Section 2.2) to support/disprove the envisioned benefits of the platform. In particular, the evidence comes from different sources and it is collected and analysed using different methods.

Anomalies analysis can help to discover how behaviors of the OpenReq platform and its users diverge from what was originally planned. We use this method to shape and explain new prepositions about the platform that were not initially foreseen. Detecting anomalies presents opportunities for identifying missing requirements, which can later be evaluated using the methods presented in this chapter.

Finally, we perform *member checking* by getting qualitative feedback on the findings from the stakeholders who provided the initial OpenReq platform requirements and related data. This approach is especially important as the results of the study may change the way in which stakeholders and participants are expected to perform their work. First, we will present our findings to the stakeholders and trial participants to help them to understand their contribution to the evaluation process. This should help them understand not just the results but also how the results were derived. Such understanding will help the OpenReq team to gain support for the conclusions that were reached through the trials. An evaluation workshop and a round of interviews can be devoted to this exercise.

2.2 Quantitative methods

There are many factors in our study that can be subject to quantitative evaluation. This can be based on a series of controlled experiments and quasi-experiments that can be replicated across cases. The first set of controlled experiment is needed to establish a baseline about the improvement (or lack of thereof) due to the adoption of the OpenReq approach in the case company/unit.

Before each experiment, the participants are trained (e.g., through a hands-on workshop) about the OpenReq tool and its features of interest for the company/unit.

In the **controlled experiments**, the participants are randomly divided into (at least) two groups. One group (the treatment) will perform a requirement engineering task targeting the OpenReq feature under evaluation (e.g., distinguish between an actual requirement and prose in a text) using the tool, whereas the other group (the control) will perform the same task as it was done before the introduction of the tool (e.g., manually). The two groups will be compared in terms of outcomes that are interesting for the specific feature (e.g., time to complete the task, precision). This first design is also referred to as *between subjects.*

The same design is applied to the other features of OpenReq, keeping the assignment to the groups random to not bias the sample. For example, avoiding that the same participants are always assigned to the treatment and, therefore, their performance influenced by their increased familiarity with the tool rather than due to the tool itself.

This replicated series of experiment, compared to a single, larger experiment has several benefits:

- *Isolate improvement windows for the tool.* Knowing exactly what feature is not showing the planned improvement (or worse causing an impairment) supports quick development iteration, so that for the subsequent release the specific feature will be given priority.
- *Mitigate risk.* For example, if a technical problem arises during the initial experiments, it can be fixed for the following iteration. Similarly, the design of the experiment itself can become more robust (e.g., by including other people in the sample which were originally overlooked).
- *Generalizability.* After the experiments takes place, several (i.e., the most important) use cases will be covered allowing to generalize the result over different requirements engineering tasks.

However, at the moment, we envision two sources of bias that might affect the validity of the studies:

- *Domain.* The results may be applicable in the specific sub domain for which the tasks are executed during the experimentation. Therefore, tasks and participants should be carefully chosen to cover several (or at least more than one) subdomains while keeping the task similar in order to be comparable.
- *Individual skills.* As often happens, individual skills and experience with a specific task might swamp the effect of the tool itself. If that is foreseen to be the case, the experiment design should take this into account.

One approach to overcome the aforementioned biases is to employ a blocking design by covering these two different dimensions as homogeneously as possible—e.g., making sure that the sample, before random assignment, includes participants from different domains and with different skill levels.

Alongside controlled experiments, which present a snapshot of the effect of the tool at a given time, we will employ also a **quasi-experiment** design—in particular, longitudinal *within subjects*.

With this approach, the participants use the OpenReq tool for a prolonged amount of time (e.g., a month) to carry-out their activities related to requirements engineering. No control group is present, but the outcome of interest (e.g., number of requirements successfully worked on) is collected at regular intervals (e.g., end of working day) and compared over a moving

window of time (e.g., daily or weekly improvement/decrease). Therefore, each participant acts as her own control group and the results are calculated as the aggregation of individual performances. This approach is complementary to the controlled experiment one, because of:

- *Reduced individual skills bias.* Since each participant is compared to herself this threat is factored out by design.
- *Ecological validity.* As the findings are generalized to real-life settings as the study approximates the actual look, feel and procedure of a normal working day.
- *Flexibility.* Based on the monitoring of the participants performances, improvement measure could be deployed in the tool at a cut-off time. This approach allows further evaluation (e.g., comparing performance before and after the cut-off).

2.3 Metrics

The following metrics can be used in various trials, see e.g. Sokolova2009 for background information.

Basics:

- Quantity: natural number
- Ratio: real number
- Proportion: ratio between 0.00 and 1.00

Binary Classification (e.g., boolean decisions) according to standard, e.g. http://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers:

- True positives (TP): quantity (correct "yes"-decisions of the tool w.r.t. expert decisions)
- False positives (FP): quantity (number of tool's "yes"-decisions where experts decided "no")
- True negatives (TN): quantity (correct "no"-decisions of the tool w.r.t. expert decisions)
- False negatives (FN): quantity (number of tool's "no"-decisions which should have been "yes")
- Prevalence: proportion (ratio of "yes"-decisions in the whole data set) = (TP+FN) / (TP+TN+FP+FN)
- Precision (measures amount of false positives, dependent on prevalence - the lower the prevalence, the lower the precision): proportion (ratio of correct "yes"-decisions to all "yes"-decisions of the tool, 1.0 is best, meaning no false positives) = TP / (TP+FP)
- Recall (sensitivity, measures amount of false negatives, independent of Prevalence): proportion (ratio of correct "yes"-decisions to all "yes"-decisions of the experts, 1.0 is best, meaning no false negatives) = TP / (TP+FN)
- Specificity (measures amount of "missing" positives, independent of prevalence): proportion (ratio of correct "no"-decisions to all "no"-decisions of the experts, 1.0 is best, meaning no false positives) = TN / (TN+FP)
- F-measure (balanced F-score, dependent on prevalence): proportion (harmonic mean of Precision and Recall, 1.0 is best) = 2 * Precision * Recall / (Precision + Recall)

- F2-measure (puts double weight on recall, i.e. false negatives, 1.0 is maximum): proportion (weighted harmonic mean of Precision and Recall) = 5 * Precision * Recall / (4 * Precision + Recall)
- Diagnostic Odds Ratio (independent of prevalence): ratio (relation of odd ratios, i.e. positive odds relative to negative odds, the higher > 1 the better) = (TP+0.5)*(TN+0.5) / (FP+0.5)*(FN+0.5)
- Accuracy (dependent on prevalence): proportion (ratio of correct tool decisions in the whole data set) = (TP+TN) / (TP+TN+FP+FN)
- ROC curve (e.g. http://en.wikipedia.org/wiki/Receiver_operating_characteristic): plotting the true positive rate (i.e. sensitivity) against the false positive rate (i.e. 1 - specificity) at various threshold settings, thus achieving more robustness, especially in case of large class unbalances

Multi-class Classification is transformed to binary classification:

- One-against-all approach: each class is separately tested ("positive" side) against the combination of all the others ("negative" side)
- This is a popular approach with some weaknesses (ambiguous selection depending on cut-off, different confidences, much fewer positives than negatives)

Multi-label Classification:

- Binary Relevance: test each class separately and add to the set of labels depending on a cut-off (threshold) value
- Hamming Distance: ratio of wrong label to all labels
- Cosine Similarity
- Jaccard index: ratio of intersection over union of predicted and true values

Aggregation:

- Averages of such values
- In case of confidence values for the binary predictions, cut-off values (thresholds) are used to classify strictly to yes/no

3 CROSS-PLATFORM OSS TRIAL (QT)

This trial is expected to run from M12 to M28

3.1 Overview

Qt is a cross-platform application framework, which is widely used for developing applications that can be run on various software and hardware platforms with little or no change in the underlying codebase, while still being a native application. The software framework is used in over 70+ industries, which develop Qt-based products for desktop, embedded, and mobile operating systems. The Qt software is licensed as open source software (OSS) with a dual licensing model (free/commercial). Qt is developed by an open developer community, which consists of both companies, independent application developers and non-profit organizations.

3.1.1 Problem description (as-is state)

The Qt trial has several input channels for requirements:

- Community raised issues on the public RM tool (http://bugreports.qt.io)
- Commercial customers raise requirements directly to the company
- The Qt Company's internal long term planning process.
- The open developer community's long term planning process on public mailing lists.

There are problems that include, for example:

- All decision-making data is not public by default, which encumbers combining and prioritising the requirements input channels. However, we see an opportunity here for augmenting the processes with both decision support systems research and intelligent recommendations.
- Management and decision-making is burdened by duplicate or very similar issues in the requirements database. This also results in inaccuracies in defining the requirements.
- Issues are being allocated to wrong people, or other incorrectly set properties or attribute values cause unnecessary work.
- End users do not necessarily know what the correct place for reporting issues is.
- Related issues and other similar tacit relationships are not being grouped, which makes understanding the whole and ensuring integrity and correctness challenging.

As distinct problems none of the above are severe, yet as the problems accumulate they cause management overhead and unnecessary work to developers.

3.1.2 Envisioned improvements

An intelligent system can be built with the aim of increasing community involvement. The system could for example:

- Help to identify users that could potentially contribute to issues related to distinct topics. This can be one based on developers' interaction profiles, their comments to discussions, and historic contribution data.
- Proactively recommend relevant issues to less experienced community members and motivate them to contribute.

- Automatically file or recommend new issues based on discussions identified in community sites (e.g., improve documentation on topic X).
- Automatically assign or recommend new issues to the relevant developer community members and users of the software.
- Automatically assign or recommend a person to be the reviewer for a user contribution.

Detect dependencies and manage requirements knowledge:

- Thematically identify and relate feature requests or bug reports to larger wholes or contexts.
- Improve duplicate, irrelevant and non-applicable requests when entering a bug report or a feature request.
- Identify and highlight requirement and issue dependencies.
- Monitor and ensure requirements quality (e.g., by providing pattern/glossary to express them). Recommend relevant stakeholders for quality assurance.
- Recommend relevant reviewers for a new requirement

Support the Qt Company's internal release planning process:

- Identify and highlight "urgent" requirements with precision.
- Identify relevant stakeholders for a requirement, feature, or bug. Help in tracking the availability and load of stakeholders and resources.
- Allow expressing and taking into account priorities of stakeholders and their rationale.
- Ensure that feature and requirement dependencies (requires, also incompatibility) are respected.
- Take into account feature integration deadlines of releases.
- Support stakeholders to prepare a group decision by e.g. highlighting relevant topics and artefacts along with their respective stakeholders to facilitate e.g. release planning decisions.

3.1.3 Evaluation strategy and instruments

In general, the research follows the iterative and incremental paradigm of Design Science. Here, incremental means that a relatively simple solution is developed at first that is then extended or adapted based on the experiences gained resulting in cyclic execution of phases in which solution is provided with more suitable features and evaluations proceeds towards more realistic settings. The phases are roughly based on Peffers et al (2007), in which the subsequent iterations consist of:

1. Problem identification and motivation
2. Defining the objectives and solution
3. Designing and implementing the solution
4. Demonstrating, applying and evaluating the solution

Most of the requirements data for the Qt project is publicly available - and can thus easily be used for testing purposes. The RM system has open interfaces that can be used to query and retrieve data and the data schema is also publicly available allowing a construction of a replicated environment.

Evaluation of the solution relies on local testing environments and the OpenReq infrastructure. Additionally, web-based survey instruments can be utilized, e.g., in the case of opinions. Alternatively, interviews, focus groups or other similar qualitative methods are applied. The research protocol for these is developed individually for each case.

Research following constructive Design Science paradigm does not necessitate predefining methods or data collection techniques. This selection will be made based on the specific needs that are discovered during iterations. At least following qualitative research techniques will likely be used: semi-structured interviews or focus groups, and qualitative analysis of textual documents. Quantitative methods about people's opinions is alternative to qualitative methods and potentially surveys or opinion polls are used. Other quantitative methods focuses on system behavior studies by measuring, e.g., performance in terms of response time.

3.2 Scope

3.2.1 Covered features/services of the OpenReq framework

The main features of OpenReq to be covered in the first phase of this trial are a part of dependence engine developed at WP5. In subsequent runs of the trial, the services from other WPs will be added.

3.2.2 Used data sources (schemata, sizes)

Mainly public sources, the Qt Jira being the most important one. As noted earlier, Jira data as well as data schemas and query interfaces are mostly open. The confidential data follows same schema but is protected from public access. Additionally, the Qt Gerrit and Git instances are good data sources. Finally, the forums, wiki, mailing lists and other services can provide additional data.

Currently the Jira instance in use has over 60000 bugs, issues and topics. A subset of these should be used, as the whole set is quite big. The Jira provides several properties for issues, which can be used in the trial to limit the number of issues. Jira is also divided into several subprojects. For example, so called "QTBUG" is the largest and the main project for the Qt framework itself. Depending on the particular situation, any of the subprojects can be selected for the trial.

For further study the Qt Gerrit instance has all the changes that go into the Qt product. The changes are connected to the issues in the Jira database. This means that combining these data sources is a good venue for further research.

3.2.3 Necessary evaluation activities

Technical feasibility tests and test for preserving data integrity are the elementary forms of test required.

To make sure the results are valid and usable as well have value in practice, it is important to understand developers and managers who are familiar with the current process. In particular, their opinions and suggestions are relevant, these need to be mapped with interviews and possibly focus groups.

Also the data used in the trial should be reviewed by the same experts to make sure that the trial has a good starting point.

Any automatic processing of data needs to be evaluated or validated manually at least partially in order to ensure correctness.

3.2.4 Participants, as preliminary planned

This can be agreed later, but the people involved in the T1.2 interviews are good candidates for information on the trial.

Specifically the trial should get input by interviews and focus groups, or other suitable methods from these groups inside the Qt Company:

- Engineering managers
- Developers
- Product management

The specific participants should be selected based on the phase of the trial. The area is wide with many people contributing, so having different voices at different times is important.

3.3 Metrics

3.3.1 Expected impact

We expect to see the following impacts from the trial:

- Improved quality of community contribution
  - Decrease in issues closed as duplicates by maintainers
  - More dependencies between issues as compared to current state
  - Reduced comments on issues (due to quality improvement)
- Improved speed of issue management process (compare issues that are run through OpenReq with those that are not)
- Improved decision making in release planning
  - Less items dropped from a timeboxed release as compared to previous releases
  - Less items added to a release (means better planning)

Most of these metrics are comparison metrics, due to the nature of the work. Everything goes toward one product that is worked on in a timeboxed model, where releases are done every six months. Some releases are more oriented toward bug fixing, and some bring in more features. To abstract this away, the metrics need to compare releases that are roughly equivalent.

3.3.2 Selected metrics

For this trial the following metrics make sense.

- number of duplicates found
- number of improved issues found
- number of issue groups identified
- quality defects found in issues (e.g., bad wording, missing labels)
- number of requirement reviewers correctly identified
- number of items dropped in release planning phase (less is good and shows better planning)
- number of items taken into a release during the feature freeze period in which items can be added by maintainer agreement (less is good, and shows better planning and decision making)

In the case of OpenReq infrastructure, performance measures of the provided inferences shall be made. For example, response time is an essential metrics.

3.3.3 Aggregation of results

The results are best left as primary results, and not aggregated. This is due to the way the development process and release schedule work in the Qt Company.

The process in use in The Qt Company is time boxed with a release twice a year, so creating aggregate measures related to work time is hard, practically impossible.

Also the content for each release changes from release to release. This makes longitudinal comparisons complicated, as it would require multiple releases to be compared.

3.3.4 Success criteria

The minimum success criteria is that the system can show consistent results with the selected impact metrics in section 3.3.2. This would mean that the OpenReq tools can perform on the same level regardless of the set of issues given to it. We plan to measure the consistency by giving the system different parts of the data set and comparing the performance.

Also the experts' opinion from The Qt Company needs to be positive on the value for use of the system.

3.4 Evaluation procedure

3.4.1 Requirements for performing evaluation

The best way is to use publicly available data. Jira and Gerrit provide excellent data sources.

In case release planning is studied in detail, then there is data that will probably need to be kept confidential to some degree. For example, long term release planning may contain the names of customers who do not wish to be known publicly and also some features that are planned are seen as strategically important in competition.

3.4.2 Archiving the input data

All input data in the trial has to be archived by the researchers of the OpenReq project internally, for public re-use

3.4.3 Sequence of the necessary steps

It has been already described in the strategy section that the evaluation is stepwise and incremental. Several different evaluation cases can be run in parallel, but be in different phases. An example is provided below but will be refined for the particular feature in question.

- At first, example data resembling real data can be used to demonstrate different features. This is simplistic and reductionist, but it demonstrates the core features exemplary manner and provides easy communication.
- Second, real data from Jira can be openly obtained to test in more realistic environment. While this shows convincingly practical applications, it is hard to demonstrate all features intuitively and efficiently.
- Third, after minimal functional integration has been achieved, evaluation can be continued in more realistic environment. At first, assessment of practical value as well as directions for future iterations are be done, e.g., by polls or focus group studies. The

objective is to obtain feedback of the current status as well as get advice for directions in the future iterations.

- Finally, tools can be tested with real data and actual users first in more or less isolated and reductionist setting by extending the scope gradually.

3.4.4 Archiving the results

The results need to be archived by the OpenReq project for possible use later, and also to enable later researchers to verify the results. The results need to be available to anyone on request.

The publications from the trial will be in open access for scientific publications and publicly available for any non-scientific publications.

All software from the trial needs to be open source as stated in the OpenReq project goals.

3.4.5 Documentation and dissemination of the results

- Scientific publications co-authored with other consortium partners.
- Open source tools.
- Demonstrations including publicly available videos.
- Tools into use in the Qt project.
- Qt communication channels.

3.5 Time and resource plan

3.5.1 List of tasks, incl. involved persons and estimated duration

The initial task list is as follows:

- Interview with experts from The Qt Company, needs 4-6 experts and interview personnel. Duration in calendar time one month
- Deployment of OpenReq platforms on any available platform, two weeks calendar time
- Pre-working the trial data, 1 month calendar time
- Initial testing, 1 month calendar time
- Verification of initial results, 2 weeks calendar time
- Tests with new data, 1 month
- Verification, 2 weeks
- Interviews with experts, 1 month
- Publications from the trial
- Result dissemination to non-scientific media

3.5.2 Contacts of persons of interest

Within The Qt Company:

- Lars König, product management
- Alex Blasche, R&D manager

- Kai Köhne, R&D manager
- Jukka Jokiniva, IT manager (in case support for data is needed)

3.5.3 Timeline

The trial starts at the beginning of 2018 with simple integrations and proceeds to more complicated use cases. Specifically during the first months, a working integration with the first version of the dependency engine of WP5 will be developed in order to show the feasibility. The integration is then assessed for practical value with experts at Qt by qualitative research methods. This will also explicate directions for the future development of the dependency engine, and define the future integration strategy after the first months' period. The objective is to have the first operational integration demo for the OpenReq mid-term in M18. In addition, detailed strategies with the services of other WPs will be defined that will be released by mid-2018.

Period	Activity	Result at end of period	Type
2018-01	First evaluations for simple integrations	Baseline for controlled experiments	internal
2018-07	Evaluations for first operational integration of dependency engine	Report of results and improvements	internal
2019-01	Evaluations for complicated use cases	Report of results and improvements	internal
2019-04	Final evaluations and documentation of evaluation results	D7.2: Open Source trial report	public

Table 1. Open Source trial - Evaluation timeline

4 TRANSPORTATION TRIAL (SIEMENS)

This trial is expected to run from M18 to M30

4.1 Overview

RFPs (Request For Proposal) for railway safety systems are issued by national railway providers and comprise natural language documents of several hundred pages with requirements of various kind (domain specific, physical, non-functional, references to standards and regulations, etc.) and level of detail. Typically, a complete bid (proposal) comprises several subsystems, such as signaling hardware, track indication, interlocking software, ETCS, SCADA, etc.

Proposals are delivered by the national sales departments of large enterprises such as Siemens. Several departments and stakeholders (project management, finances, system development, components design, engineering, tools, integration, assembly, safety, etc.) of the proposing company must work together to find a good solution to cover all requirements at a competitive price.

4.1.1 Problem description (as-is state)

After the decision to answer a RFP, a bid project is started. The team comprises a project manager (BPM), a requirements administrator (RMiP), a system architect (SM), and additional experts and stakeholders. Typically, these are 10 - 30 persons, but the number may change during the bid process. The time frame of a bid project is typically 1-3 months.

Requirements engineering is a sub-task within the bid process. Its main purpose is to ensure the technical compliance of the offer.

- As a first step, the requirements administrator analyses the technical documents of the RFP and - after some optional restructuring steps - imports the text paragraphs into IBM DOORS. A specific template for bid projects provides the necessary attributes (e.g. source, type, classification, compliance, risk, approach).
- Next, the requirements administrator decides, which of those DOORS entries are requirements ("DEF") and which are not ("Prose"). This classification is based on domain knowledge and experience from past bid projects.
- All requirements are assigned to one or more domains (ca. 50 are predefined, project-specific domains can be added). The interpretation of domain names is specific to the bid project and relates to the stakeholders (internal departments, external subcontractors) in the bid project.
- Then, the corresponding stakeholders check their assigned requirements for technical compliance (yes, with comment, no) and optionally give additional input to different aspects of the requirement, such as the approach taken to satisfy the requirement, the risk involved in the proposed solution, etc.
- Repeatedly, the requirements administrator checks whether all requirements are set to compliant, which is crucial for submission of a proposal. Group decisions may be necessary concerning comments, solution approaches and risks.
- During the bidding phase, each participating company can ask questions to the railway provider. These questions and answers (Q&A) are communicated to all bidders and may lead to modified requirements, which again must be checked for compliance.

- After the assessment of the requirements is finished, the final state (used for submission of the proposal) is documented as a baseline for subsequent modifications during negotiations with the customer in case the bid is won.

Siemens division Mobility has defined an elaborate RM process with tool support which is followed in bid projects.

4.1.2 Envisioned improvements (by OpenReq services)

Interviews with a requirements manager (i.e. the RM process owner), a requirements administrator, a system architect, and a few stakeholders at Siemens Mobility Management (Rail Automation) revealed potential improvements in the current RM process.

The process could be made more efficient with the following functions (quantitative objectives were not stated by the interviewees):

- The System shall provide an ontology describing basic concepts and a glossary of railway terms and technologies.
- The User shall be able to define bid project specific information like used domains, domain experts, stakeholders, approaches, etc.
- The System shall be able to import requirements from a text document without manual assistance (such as formatting).
- The System shall be able to classify if a given text entry is a requirement or not.
- The User shall be able to restructure requirements in the tool (e.g. add dependencies to other requirements, or cut a requirement into two) without losing the connection to the original text in the source document.
- The System shall be able to suggest the most likely domain(s) for a requirement.
- For a selected requirement the System shall be able to show similar requirements within the bid project or other projects or versions. Similarity can be based on content, technologies, components etc.
- The System shall warn about contradicting requirements within a bid project.
- The System shall be able to compare similar requirements (approach, compliance, risk, etc.).
- The System shall be able to identify approaches (solutions, technologies) for a requirement based on the system ontology.
- The System should be able to learn from existing requirement classifications and domain assignments in existing bid projects.
- The System shall inform Users when their requirement assignment status has changed.
- The System shall support multi-language requirements (English, German, etc.).
- The System shall support group decisions concerning compliance and solution approaches.

4.1.3 Evaluation strategy and instruments

Several case studies are used to validate the results of OpenReq:

- First, the relevant OpenReq services are tested independently on data of previous (completed) bid projects in a lab environment by the Siemens OpenReq team
- Such tests are implemented as batch jobs and can easily be repeated for new versions of the OpenReq services
- Later, field studies are conducted by the requirements administrator and optionally the stakeholders, based on a proof-of-concept integration of the OpenReq services in the DOORS environment
- The RM process for a completed bid project is repeated using that integration
- Its results are compared with the original results
- Its usability is evaluated based on a questionnaire

We use the scientific methods described in section 2, such as quasi-experiments, field studies, usability studies (incl. bid-specific customization), etc. The details are specified in section 4.3.

4.2 Scope

The evaluation covers requirements management (RM) in the bid project. A subsequent realization project is out of scope (although there is a managed process for it as well).

4.2.1 Covered features/services of the OpenReq framework

Although the OpenReq services are not yet defined in detail, we expect that the following services will be implemented and can be evaluated:

- Define domain-specific extensions (e.g. rail automation) to generic ontology
- Customize project-specific (i.e. bid-specific) domains and settings
- Extract requirement candidates from English text
- Classify a requirement candidate as requirement or prose (range of 0..1)
- Suggest one or more ontology concepts (categories) for a requirement, e.g. "security", "financial", domain-specific technologies, etc.
- Rate the quality of a requirement
- Decide whether two requirements are similar, i.e. cover the same contents
- Decide whether two (similar) requirements are contradicting
- Decide whether two (similar) requirements are redundant (or one subsuming the other)
- Suggest one or more solution approaches for a requirement
- Support the group decision about a conjoint (structured) solution for a list of requirements

Other offered services are not subject to the use case and therefore ignored.

4.2.2 Used data sources (schemata, sizes)

Presently, data from 4 bid projects for 3 countries are available. During the project, more data sets will become available.

For each project, data comprise

- a set of tender documents (technical requirements)
  - In average 3 documents with 200 pages each
  - All documents are in English although many of them were translated from the original language of the tender
  - As there are various authors and translators, wording may be different in different documents, even within a project
- an export of all objects (requirements and prose) from the corresponding DOORS project in Microsoft Excel CSV format (or alternatively in ReqIF format), with the following information:
  - text (as imported from the tender document)
  - hierarchical structure of the text (i.e. context of requirements) - optional
  - type ("DEF", "Prose")
  - domains (i.e. roles of assigned stakeholders)
  - for each stakeholder:
    - compliance ("yes", "conditional", "no")
    - complianceComment (natural language text) - optional
    - approach (bid-specific enumeration of offered product/solution types)
    - approachComment (natural language text) - optional
    - risk ("high", "medium", "low", "no")
    - riskComment (natural language text) - optional
- additional information (expert knowledge) may be added as reference data for evaluation:
  - conjoint compliance incl. comment (as a group decision of all assigned stakeholders)
  - conjoint approach incl. comment (as a group decision of all assigned stakeholders in the context of the whole project)
  - conjoint risk incl. comment (as a group decision of all assigned stakeholders in the context of the whole project)

For the first evaluations, such additional information will be added manually by enhancing the available test data before evaluation. Later, when an appropriate UI is available, this may be done in a more interactive way where the system suggests values and the experts confirms or rejects.

Depending on the evaluation procedure, some of the data are used for tuning the algorithms (training data set) and only the remaining data are used for evaluation (test data set). Those sets will be partitioned differently for different evaluation steps. Alternatively, we consider three-fold cross-validation (as the least resource-consuming of the k-fold family).

No data protection measures are required as no private data are stored in DOORS.

4.2.3 Necessary evaluation activities

Post-mortem analyses (using metrics as defined in section 2.3):

- Executed on specified subsets of the data
- Evaluated by Siemens OpenReq team
- Repeated for alternative implementations of the OpenReq services (comparison)
- Repeated for newer versions of the OpenReq services (verify improved performance)

Field studies (following the quasi-experimental design approach of section 2.2):

- Based on local UI or Web UI (not directly integrated in DOORS)
- Repeat the RM process for a completed bid project with the OpenReq framework
- Compare the (new) results with the reference from the completed project
- Executed first by Siemens OpenReq team and later by Siemens Mobility Management (Rail Automation)
- Evaluated by Siemens OpenReq team

Optional (only if time permits): Usability studies
- Based on local UI or Web UI
- Execute the RM process for a new bid project with the OpenReq framework
- Executed by Siemens Mobility Management (Rail Automation)
- Evaluated by Siemens OpenReq team (structured interviews)

Optional (only if time permits): Interviews (according to the description in section 2.1)
- Evaluate the benefits for users (in the field studies)
- Can be carried out during the field studies

4.2.4 Participants, as preliminary planned

Siemens OpenReq team: 3 participants
- Andreas Falkner: senior researcher
- Gottfried Schenner: senior researcher
- Alexander Schörghuber: junior developer

Siemens Mobility Management (Rail Automation): 5 participants

4.3 Metrics

4.3.1 Expected impact

The following goals and impact areas as described in GA.1.3.3 are relevant for the Siemens trial. The default value for the defined impact areas is "strong":
- Time: Reduction of the time to market
- Productivity: Increased productivity
- Quality: Software Quality
- Reuse: Increased Reuse

Goal IDENTIFY: Reduce manual efforts needed to identify requirements from natural language texts (30% from RFPs)
- Potential Metrics: Ratio of the number of automatically identified requirements to the number of manually identified requirements
- Expected impact: Productivity (very strong), Quality

Goal: Reduce number of requirements changes (evolution) by 20%
- Potential Metrics: Number of requirements changes from initial stage (e.g. creation) to its acceptance (e.g., release assignment); Time needed until a requirement is accepted

- Expected impact: Time, Productivity
- Refining original requirements (tender text) is no responsibility of this trial (just a weak relationship could be seen if Q&A were covered or when derived DOORS requirements were changed/extended). Therefore this goal is not covered in the evaluation.

Goal REUSE: Increase requirements reuse
- Potential Metrics: Ratio of the number of automatically identified requirements for reuse to the number of manually identified requirements; Number of requirements reused; Number of requirements identified from natural language texts
- Expected impact: Productivity (very strong), Quality (very strong), Reuse (very strong)

Goal DEPENDENCIES: Increase reuse of requirements dependencies
- Potential Metrics: Ratio of the number of automatically identified requirements dependencies for reuse to the number of manually identified requirements dependencies; Number of requirements dependencies reused; Number of requirements dependencies identified from natural language texts
- Expected impact: Productivity (very strong), Quality (very strong), Reuse (very strong)

Goal EXPENSES: Reduce RE expenses for a bid project by 30%
- Potential Metrics: The difference of actual RE expenses for a bid and averaged expenses over recent similar projects
- Expected impact: Productivity (very strong)

Goal GROUP: Improve group decisions in RE setting
- Potential Metrics: Ratio of the average time needed for a group decision to the average number of stakeholders involved
- Expected impact: Productivity (very strong)

The next section describes which aspects can be actually measured by the planned evaluation process and the available data sources.

4.3.2 Selected metrics

Metrics for service: Extract requirement candidates from English text
- Quantity of automatically identified requirements
- Precision and Recall of automatically identified requirements w.r.t experts' reference
- Serves goal IDENTIFY

Metrics for service: Classify a requirement candidate as requirement or prose
- Quantity of correctly classified requirements (TP)
- Precision of automatically identified requirements w.r.t experts' classification
- Recall of automatically identified requirements w.r.t experts' classification
- Serves goal IDENTIFY

Metrics for service: Suggest one or more ontology concepts (categories) for a requirement
- Quantity of correctly assigned stakeholders in total (TP)
- Quantity of completely correctly assigned requirements
- Precision of automatically assigned requirements w.r.t experts' assignments

- Recall of automatically assigned requirements w.r.t experts' assignments
- Remark: Categories cannot be evaluated directly, as there are no real test data for categories (just for stakeholders which are responsible for such categories), however, test data could be added by an expert
- Serves goals REUSE, EXPENSES

Metrics for service: Rate the quality of a requirement
- Presently out of focus as we cannot influence quality of tender documents
- In future, evaluation could be done after identifying or injecting "bad quality requirements"

Metrics for service: Decide whether two requirements are similar, i.e. cover the same contents (e.g. different contents because of different context despite very similar wording such as "maximal temperature in hardware room …" vs. "maximal temperature of hardware module ...")
- Quantity of similar requirements in a project
- Quantity of similar requirements over all projects
- Precision and Recall of automatically identified similarities (for all pairs) w.r.t experts' reference
- Serves goal REUSE, DEPENDENCIES

Metrics for service: Decide whether two (similar) requirements are contradicting
- Precision and Recall of automatically identified contradictions (for all pairs) w.r.t experts' reference
- Serves goal REUSE, DEPENDENCIES

Metrics for service: Decide whether two (similar) requirements are redundant (same contents or one subsuming the other)
- Precision and Recall of automatically identified equivalences (for all pairs) w.r.t experts' reference
- Precision and Recall of automatically identified subsumptions (for all pairs) w.r.t experts' reference
- Serves goal REUSE, DEPENDENCIES

Metrics for service: Suggest one or more solution approaches for a requirement
- Quantity of correctly assigned approaches in total (TP)
- Quantity of completely correctly assigned requirements
- Precision of automatically assigned approaches w.r.t experts' assignments
- Recall of automatically assigned approaches w.r.t experts' assignments
- Serves goal EXPENSES

Metrics for service: Support the group decision about a conjoint (structured) solution for a list of requirements
- Ratio of the average time needed for a group decision to the average number of stakeholders involved

- Serves goal GROUP

The quantities from above are used to calculate the reduction effect (in %) on RE expenses by comparing the difference of actual RE expenses for a bid and averaged expenses over recent similar projects.

In future versions of this document and depending on progress of implementation of OpenReq framework, other metrics may be added, such as efficiency, satisfaction, usefulness.

4.3.3 Aggregation of results

Quantities (incl. TP, FP, FN) are counted for each evaluation step and data set.

Precision and Recall are calculated based on those quantities and represented as percentages (0-100). F-measures are calculated based on Precision and Recall and represented as percentages (0-100). Averages are calculated over data sets and bid projects.

4.3.4 Success criteria

The following objectives and means for verification as described in GA.1.1.3 are relevant for the Siemens trial.

O3. Increase the productivity of stakeholder and the quality requirements

- Productivity increase of stakeholders (communication overheads, time efforts in decision making, time to understand requirements): Significant improvement compared to current RE tools, 20% less time for each requirement on average
- Accuracy of recommended items: >75% precision and >75% recall
- Efficiency of recommendation algorithm at runtime: Response time < 5 sec
- Novelty of recommendations (surprise factor): >10% of recommended items
- Perceived usefulness of recommendations (questionnaires): > 50% accepted recommendations

O4. Improving group decisions of stakeholders (time, decision quality, and satisfaction)

- Reduction of communication overheads, decision making efforts: Significant improvement compared to current tools
- Efficiency of group decision recommendation algorithm at runtime: Response time < 8 sec
- Acceptability of the recommended tradeoffs: >50% accepted
- Distraction factor due to recommendation functionality (observation): <20% false positive recommendations

O5. Increase requirements reuse, identify and manage requirements dependencies

- Adequacy of reusable knowledge (improved elicitation process by a significant amount of requirements coming from previous projects / releases): 20% less efforts for managing RE models, >80% of patterns reused/accepted in trials
- Quality and adequacy of the ontologies as perceived by different stakeholders and domains: Over 50% of agreement
- Reduced time for inconsistency detection, inconsistencies repair: Significant improvement compared to current RE tools (> 5%)

O6. Full integration into Stakeholder's workflows and tools

- Usability (User/stakeholder satisfaction with GUI concepts and design: >80% user acceptance
- Availability in stakeholder tools: Recommendation features available in >5 legacy tools
- Workflow coverage (different types of tasks, e.g. estimation, elicitation, release planning): >50% support of stakeholders' work time

4.4 Evaluation procedure

Evaluation is done in an agile way. There are several independent steps which can be repeated with newer versions of the OpenReq services. Thus, feedback to services implementation (WP2-6) will be available as early as possible.

4.4.1 Requirements for performing evaluation

OpenReq services are available and installed locally (at the evaluation site). This is ensured by the defined architecture and deployment process. A prototypical evaluation environment was already built on a local computer and will be ported to an internal server in order to do evaluations for Siemens trial in a protected environment (access rights for dedicated Siemens testers only).

Software quality of deployed services must be ensured, e.g. by regression tests before release.

For the field study, the proof-of-concept integration must be available and installed at the Siemens business unit as well. If training is necessary, it must be completed before the evaluation and the corresponding (domain-specific) customization must be installed as well.

Test data must be available (see next section).

4.4.2 Archiving the input data

Due to business reasons, input data and reference data cannot be made public. They are stored in a local folder at the Siemens evaluation site: see section 4.2.2 for details.

A restricted set of (partly anonymized) data was uploaded to Tuleap: Project Documentation / DoA / WP7 / Resources / Siemens / Siemens_Example_Tender_Confidential for test use by the university partners only (it must not be made public).

4.4.3 Sequence of the necessary steps

Each regression test (for a service as listed in section 4.3.2) and each field study (see section 4.2.3) can be considered one step. Those steps are independent of each other and can be executed in any order. List of steps:

4.4.4 Archiving the results

Due to business reasons, the raw results cannot be made public. They are stored in a local folder at the Siemens evaluation site next to the input data (see section 4.4.2).

However, aggregated results and improvements w.r.t. the baseline will be stored in the TULEAP project repository.

4.4.5 Documentation and dissemination of the results

The results (all selected metrics and success criteria) will be documented in D7.3: Transportation trial report (Siemens internal).

Aggregated and anonymized results will be made public in D7.5: OpenReq evaluation report. For each evaluated service, it will contain following metrics, aggregated over all projects in the Siemens trial:

- Quantities (TP, FN, FP)
- Precision
- Recall
- F-measure
- Accuracy

4.5 Time and resource plan

4.5.1 List of tasks, incl. involved persons and estimated duration

The following tasks will be executed several times for different versions of the OpenReq platform and for various test data (tender documents from bid projects).

Task	Persons	Hours
Deploy new version of OpenReq platform at evaluation site	Schenner	4
Add a new regression tests (see 4.4.3)	Schenner, Schörghuber	16
Repeat the whole set of regression tests and evaluate the results	Falkner, Schenner, Schörghuber	8
Execute a field study (see 4.4.3)	Falkner, Schenner, experts from business unit	40
Create an (internal) evaluation report	Falkner	16
Optional: Interview a domain expert (user)	Falkner, Schenner	8

Table 2. Transportation trial - Repeated tasks

4.5.2 Contacts of persons of interest

All contacts to Siemens stakeholders are channeled via Andreas Falkner or Gottfried Schenner.

4.5.3 Timeline

Period	Activity	Result at end of period	Type
2018-07	Evaluations for first major OpenReq release	Baseline for controlled experiments	internal
2019-01	Repeated evaluations for patches to OpenReq release	Report of results and improvements	internal
2019-06	Documentation of evaluation results	D7.3: Transportation trial report	internal
2019-07	Repeated evaluations for second major OpenReq release	Report of results and improvements	internal
2019-08 - 2019-12	Final evaluations	D7.5: OpenReq evaluation report	public

Table 3. Transportation trial - Evaluation timeline

5 TELECOM TRIAL (WIND TRE)

This trial is expected to run from M21 to M33

5.1 Overview

The telecom market is more and more competitive.

The advent of social networks and the pervasiveness of connected devices have changed the way enterprises are engaging their customers. In particular, Telecom operators strive to attract and retain this new generation of customers that is socially connected and highly informed. For this, it is increasingly important to offer new innovative software-enabled products and services corresponding to customer expectations, more and more usable and responsive.

Software quality is a value for modern telecom operators: high quality software corresponds to an economic values, churn and IT costs reduction, increased overall company efficiency. Brand management is becoming more and more a social discipline. Making business today is, in other words, a social matter: customers can and want to contribute towards improvements of existing products and services.

A strong involvement of customers is crucial for telecom companies. Telecom companies are not only interested in acquiring and retaining customers, but also to leverage customer's creativity by enabling them to significantly contribute to the evolution of both relevant and innovative requirements.

5.1.1 Problem description (as-is state)

Users are engaged using different channels and different ways. For understanding customer's opinions, social media analysis has been advertised as one of the most promising methods.

Wind Tre needs to foster massive user involvement and automated identification and extraction of requirements from user-generated content (e.g., Twitter, Facebook, and other social networks).

Moreover Wind Tre would reduce interpretation conflicts among the stakeholders through a real-time synchronization capacity between enterprise and users, so that a speed-up of the decision process and overall company's efficiency are achieved.

Identify and extract requirements from user requests and monitor the pulse of the communities to identify acute issues to enable early risk assessment will enable Wind Tre to understands the customer's needs.

Taking collective decisions for enabling innovations on the basis of massive amount of requirements knowledge is a complex and challenging endeavour.

OpenReq in supporting group decisions of stakeholders in requirement evaluation processes in a context where requirements themselves are inferred using intelligent systems technologies for many different input types.

Allows:

- Automatically propose prioritization indicators for requirements derived from the user discussions (e.g., weight a requirement by the number of related user requests).
- Automatically propose prioritization indicators from usage behaviour (e.g., recommend a higher importance for issues related to highly used services.)

- Support stakeholders in the preparation for a group decision

The trial will evaluate the usefulness of OpenReq in supporting group decisions of stakeholders in requirement evaluation processes in a context where requirements themselves are inferred using intelligent systems technologies for many different input types.

5.1.2 Envisioned improvements

The requirement definition and management in Wind Tre is not supported by automatic and technology solutions. Moreover the social media channel is managed only for customer service purpose.

There are lot of improvements in requirement management process; the following functionalities will be useful:

- The System shall provide an ontology describing basic concepts, a glossary of telecom terms and technologies.
- The System shall be able to import social network data (at least 6 months of data).
- The System shall be able to understand if the social network data is relevant for a telecom company or not.
- The System shall be able to classify (tag) the social network data.
- The System shall be able to notify to the Stakeholder if a social network data is a request of new functionality (social network data without a related tag).
- The System shall be able to import requirements from a plain text document (not structured) without manual assistance.
- The System shall be able to match the tags of social network data with tag/basic concept of requirements.
- The System shall support Italian language.
- The System shall support group decisions and suggest prioritized requirements.

5.1.3 Evaluation strategy and instruments

OpenReq can be seen as a kind of black-box with 2 explicit inputs:

- Social network data
- Example of requirements

And an implicit input: customer usage from antennas.

The OpenReq features will be verified at different level through the score model approach.

The first level will check the tags within the social network data, the second will check the correspondence between social network data (tags) and tag/basic concept of requirements, the third level will check the analysis of customer usage and the correspondence with phone network capabilities requirement, the last will check the requirements prioritization.

5.2 Scope

The evaluation covers social network analysis and requirements prioritization.

5.2.1 Covered features/services of the OpenReq framework

We expect that the following services will be implemented and can be evaluated:

- Acquisition of social network data in Italian language
- Remove duplicate data (e.g. retweet)
- Classify (tag) the social network data
- Classify (tag) requirements or suggest one or more ontology concepts (categories) for a requirement.
- Extract technical requirement candidate from customer usage, e.g. increase network capability in specific area
- Prioritize requirements.

5.2.2 Used data sources (schemata, sizes)

In the telecom use-case we will use different data sources:

- Tweet from Twitter, creation date, hashtag (#) and mentioning (@), emoticon
- Post from Facebook, creation date, telecom operator object, emoticon
- Customer usage from network antennas
- Business requirements

Tweet, post and business requirements are in Italian plain text; the business requirements contain, in some cases, structured data (for example bullet points for privacy, security requirements).

Tweets and post are public data downloadable from Twitter and Facebook with registered API.

Data about customer usage are structured:

- Msisdn (phone number)
- Start date
- End date
- Antenna location
- Other data

For privacy Italian regulation all personal data will be anonymized.

5.2.3 Necessary evaluation activities

The process of evaluation shall be:

- OpenReq captures data from social network
- OpenReq removes duplicate data
- OpenReq classifies (tag) the social network data
- Evaluation: Apply score model approach on a subset of data classified
- OpenReq captures business requirements
- OpenReq classifies (tag) the business requirements
- Evaluation: Apply score model approach on a subset of business requirements classified (TBC)

- OpenReq matches the social network tags with tag/basic concept of requirements.
- Evaluation: Apply score model approach on a subset of matches
- OpenReq extract technical requirement candidate from customer usage report
- Evaluation: Manual check
- Openreq prioritize requirements
- Evaluation: Manual check through OpenReq UI.

5.2.4 Participants, as preliminary planned

Wind Tre OpenReq team

Wind Tre Architecture: 1 participant

Engineering OpenReq team

5.3 Metrics

5.3.1 Expected impact

The following goals and impact areas as described in GA.1.3.3 are relevant for the Telecom trial. The default value for the defined impact areas is "strong":

- Time: Reduction of the time to market
- Productivity: Increased productivity
- Quality: Software Quality
- Reuse: Increased Reuse

Goal: Reduce manual efforts needed to identify requirements from natural language texts (40% from user requests)

- Potential Metrics: Ratio of the number of automatically identified requirements to the number of manually identified requirements
- Expected impact: Productivity (very strong), Increased Reuse

Goal: Reduce number of duplicate comments and requests by 40%

- Potential Metrics: Ratio of the number of duplicate comments and requests found automatically to the number of duplicate comments and requests flagged manually
- Expected impact: Productivity (very strong), Increased Reuse

Goal: Filter irrelevant comments and requests from social network by 30%

- Potential Metrics: Ratio of the number of irrelevant comments and requests found automatically to the number of irrelevant comments and requests flagged manually
- Expected impact: Productivity (very strong)

Goal: Reduce time needed for release planning

- Potential Metrics: Ratio of the average time of meetings needed for release planning to the average number of stakeholders involved. Number of changes to the release schedule
- Expected impact: Time, Productivity (very strong)

Goal: Improve group decisions in RE setting

- Potential Metrics: Ratio of the average time needed for a group decision to the average number of stakeholders involved
- Expected impact: Productivity (very strong)

5.3.2 Selected metrics

In general, following types of metrics are based on Score Model Approach:

- Quantity: natural number
- True positives (TP): natural number (number of correct "yes"-decisions of the tool w.r.t. expert decision)
- False positives (FP): natural number (number of tool's "yes"-decisions where experts decided "no")
- False negatives (FN): natural number (number of tool's "no"-decisions which should have been "yes")
- Precision: percentage (ratio of correct "yes"-decisions to all "yes"-decisions of the tool = TP / (TP+FP))
- Recall: percentage (ratio of correct "yes"-decisions to all "yes"-decisions of the experts = TP / (TP+FN))
- Averages of such values

Metrics for service: Extract requirement candidates from Italian text

- Quantity of automatically identified requirements
- Precision and Recall of automatically identified requirements w.r.t experts' reference

Metrics for service: Classify a requirement candidate as requirement or prose

- Quantity of correctly classified requirements (TP)
- Precision of automatically identified requirements w.r.t experts' classification
- Recall of automatically identified requirements w.r.t experts' classification

Metrics for service: Suggest one or more ontology concepts (categories) for a requirement

- Quantity of completely correctly assigned requirements
- Precision of automatically assigned requirements w.r.t experts' assignments
- Recall of automatically assigned requirements w.r.t experts' assignments

Metrics for service: Decide whether two requirements are similar, i.e. cover the same contents

- Quantity of similar requirements from Social Network Data
- Precision and Recall of automatically identified similarities w.r.t experts' reference

Metrics for service: Decide whether two (similar) requirements are contradicting

- Precision and Recall of automatically identified contradictions (for all pairs) w.r.t experts' reference

Metrics for service: Decide whether two (similar) requirements are redundant (same contents or one subsuming the other)

- Precision and Recall of automatically identified equivalences (for all pairs) w.r.t experts' reference

- Precision and Recall of automatically identified subsumptions (for all pairs) w.r.t experts' reference

5.3.3 Aggregation of results

Quantities (incl. TP, FP, FN) are counted for each evaluation step and data set.

Precision and Recall are calculated based on those quantities and represented as percentages (0-100).

5.3.4 Success criteria

The following objectives and means for verification as described in GA.1.1.3 are relevant for the Telecom trial.

O2. Derive actionable RE insights from large amount of user feedback

- Availability of requirements intelligence and analytics component. Component provides useful aggregation of user feedback to derive requirements decisions.
- Visualization of explicit and implicit feedback and combination of both for RE related tasks. Reduced time to process explicit and implicit user feedback by increased processing coverage and quality. 20% less time to process change reports, 50% less time to process online reviews

O4. Improving group decisions of stakeholders (time, decision quality, and satisfaction)

- Increased satisfaction with decision process/outcome, level of trust. Significant improvement compared to current state

5.4 Evaluation procedure

The evaluation procedure will done by applying the Score Model approach.

Each OpenReq results will be verify comparing a subset of data with manual analysis as described in 5.2.3.

5.4.1 Requirements for performing evaluation

To implement the Score Model approach there will be available

- Subset of social network data for manual analysis
- OpenReq results about social network data analysis on the same subset of social network data
- Subset of business requirements
- OpenReq results about the same subset of business requirements
- Subset of correspondence (match) social network tags with tag/basic concept of requirements.
- OpenReq requirements prioritization with number of related social network data (though UI)

These data could be available through UI or as report-format.

5.4.2 Archiving the input data

The input data will be available and archived only for trial phase.

5.4.3 Sequence of the necessary steps

Applying the Score Model approach the necessary steps are described in 5.2.3.

5.4.4 Archiving the results

The result should be available as report-format with the metrics details.

5.4.5 Documentation and dissemination of the results

According to this Wind Tre plans to disseminate OpenReq's results by means of internal and external communications. In particular results will be disseminated through the company website and social media and other channels to be defined along the project time.

5.5 Time and resource plan

5.5.1 List of tasks, incl. involved persons and estimated duration

As described in 5.2.3 the list of activities shall be:

- OpenReq captures data from social network
- OpenReq removes duplicate data
- OpenReq classifies (tag) the social network data
- Evaluation: Apply score model approach on a subset of data classified
- OpenReq captures business requirements
- OpenReq classifies (tag) the business requirements
- Evaluation: Apply score model approach on a subset of business requirements classified
- OpenReq matches the social network tags with tag/basic concept of requirements.
- Evaluation: Apply score model approach on a subset of matches
- OpenReq extract technical requirement candidate from customer usage report
- Evaluation: Manual check
- Openreq prioritize requirements
- Evaluation: Manual check through OpenReq UI.

For performing evaluation the resources involved shall be:

- Engineer and Wind Tre Architect to install OpenReq in a private environment
- Engineer to manage OpenReq and its functionalities
- Wind Tre and Engineering OpenReq team to manage the evaluation stage

The estimate duration of the trial is 12 months.

5.5.2 Contacts of persons of interest

All contacts to Wind Tre stakeholders are channeled over Fabrizio Brasca.

5.5.3 Timeline

Period	Activity	Result at end of period	Type
2018-09	Evaluations for first major OpenReq release	Baseline for controlled experiments	internal
2019-03	Repeated evaluations for minor OpenReq releases	Report of results and improvements	internal
2019-07	Repeated evaluations for second major OpenReq release	Report of results and improvements	internal
2019-09	Documentation of evaluation results	D7.4: Telecom trial report	internal

Table 4. Telecom trial - Evaluation timeline

6 OVERALL TIMELINE AND PROJECT COVERAGE

The timeline table is compiled from all trials (sections 3.5, 4.5, and 5.5).

Period	Activity	Result at end of period	Type
2018-01	First evaluations (Qt, Siemens, Wind Tre) for preliminary OpenReq services	Baseline for controlled experiments	internal
2018-07	Repeated evaluations (Qt, Siemens, Wind Tre) for first major OpenReq release	Report of results and improvements	internal
2019-01	Repeated evaluations (Qt, Siemens, Wind Tre) for first integrated OpenReq release	Report of results and improvements	internal
2019-04	Documentation of evaluation results (Qt)	D7.2: Open Source trial report	public
2019-06	Documentation of evaluation results (Siemens)	D7.3: Transportation trial report	internal
2019-07	Repeated evaluations (Qt, Siemens, Wind Tre) for second major OpenReq release	Report of results and improvements	internal
2019-09	Documentation of evaluation results (Wind Tre)	D7.4: Telecom trial report	internal
2019-08 - 2019-12	Final evaluations (Qt, Siemens, Wind Tre) and compilation of results	D7.5: OpenReq evaluation report	public

Table 5. Overall evaluation timeline