Optique™

| | |
|---|---|
| Project N°: | **FP7-318338** |
| Project Acronym: | **Optique** |
| Project Title: | **Scalable End-user Access to Big Data** |
| Instrument: | **Integrated Project** |
| Scheme: | **Information & Communication Technologies** |

# Deliverable D1.2
# Requirement Analysis and Evaluation Framework

| | |
|---|---|
| Due date of deliverable: | (T0+15) |
| Actual submission date: | Feb 10, 2014 |


---

| | |
|---|---|
| Start date of the project: **1st November 2012** | Duration: **48 months** |
| Lead contractor for this deliverable: | **UoA** |
| Dissemination level: | **PU** – Public |

Final version

# Executive Summary:
## Requirement Analysis and Evaluation Framework

This document summarises deliverable D1.2 of project FP7-318338 (Optique), an Integrated Project supported by the 7th Framework Programme of the EC. Full information on this project, including the contents of this deliverable, is available online at `http://www.optique-project.eu/`.

This document gives a description of the requirements gathering methodology as well as the evaluation framework to be used for the duration of the Optique project.

## List of Authors

Akrivi Katifori (UoA)
Rudolf Schlatte (UiO)

# Contents

# Chapter 1

# Introduction

## 1.1 Purpose of this document

Capturing the requirements for software systems is a problem of communication: "Those who want the new software (either to use or to sell) must communicate with those who build the new software" [6]. There are two main sources of requirements in the Optique project: the end users' needs and the scientific tasks as set forth in the Description of Work (DoW). The main end user input to the project is provided during the annual end user workshops, where the current prototype is evaluated viz. the previous years' requirements and new requirements are found and formulated. Based on the annual reports from Work Package 1 (D1.1, D1.3–D1.5), the scientific project leadership can decide annually on implementation plans and project focus for the next project year.

The purpose of this document is to describe the methodology applied in Optique for structuring the communication process between potential users and stakeholders of the system and the developers, both during the requirements elicitation phase and the evaluation processes.

*A note on terminology.* In the DoW and various project reports, the term "use case" is used to refer to Work Packages 8 and 9 (as in, "The Siemens Use Case", "The Use Case Partners"). In this document, "use case" stands for its common meaning in the wider software architecture community; we refer to Work Packages 8 and/or 9 and the industrial partners directly.

## 1.2 Structure of this document

Chapter 2 gives an overview of the framework for requirements elicitation and evaluation, and of the process of setting up the end user workshops. Chapter 3 explains the requirements elicitation framework in detail, Chapter 4 discusses the evaluation methodology.

# Chapter 2

# Requirements and Evaluation Framework Overview

Requirements gathering and evaluation are crucial activities for the success of the Optique project, both for guaranteeing forward progress in the development of the Optique system and methodology, and for ensuring that the developed artifacts are fit for purpose. This chapter gives an overview of the annual review and feedback process done by work packages 1, 8 and 9.

## 2.1 End user workshop planning

Planning the end user workshops has to start 3–4 months in advance, i.e., during the month of June. To ensure end user availability, the dates have to be announced and participation confirmed well in advance. The preparation work involves mostly the end user and preparation teams of Work Packages 8 and 9. The following table lists the necessary preparation tasks, and their approximate time before the workshop:

**3 months in advance** The end user team decides on a date and begins planning for the venue. End users are invited and asked to reserve the date.

**2 months in advance** The evaluation team decides on the evaluation criteria and designs appropriate tasks for the end users to execute during the workshop. A trial run of the evaluation tasks is done using the current prototype. Any shortcomings and/or missing features are communicated to WP2.

**1 month in advance** The evaluation team confirms workshop dates and end user participation. Travel and accommodation is arranged for all Optique workshop participants. The evaluation tasks are finalized and tested against the current prototype. All necessary material for the workshop (questionnaires, manuals, etc.) is prepared.

**2 weeks in advance** The installation team freezes the prototype and installs a known-good version on the demo machines, either on-premise or using suitable portable machines.

In the Optique project, we are fortunate to have access to expert end users from our industrial project partners. On the other hand, these experts might not be available for a multi-day workshop, hence the project has to plan on having full access to end users for a limited time only, e.g., for one working day.

Figure 2.1 contains a proposed agenda that can be used for planning a one-day end user workshop. The time before lunch is devoted to introducing the end users to the system and letting them interact with and evaluate the current prototype. Based on their experiences with the prototype, the one-on-one interviews in the afternoon capture their user stories, use cases and requirements. After the interviews, the evaluation team consolidates the findings. The consolidated requirements are then presented and discussed in a plenary session. The outcomes of a one-day end user workshop are:

| | |
|---|---|
| **30min** | Welcome, project presentation and introduction of participants |
| **30min** | Demonstration of current prototype |
| **2h** | Usability evaluation of current prototype by end users |
| | *break* |
| **90min** | 1:1 end user interviews |
| **30min** | Identification of draft requirements by Optique experts |
| **90min** | Plenary discussion + workshop on requirements |

Figure 2.1: One-day workshop agenda (proposal).

- Evaluation results of the current prototype
- End user requirements and user stories based on observing the end users

Durations of the items on the schedule, and the overall agenda, should be adjusted depending on the focus of the workshop (requirements gathering vs. evaluation). A one-day workshop mandates some compromise between the requirements elicitation and evaluation processes. We recommend starting with the evaluation process – during formal and informal evaluation as described in Section 4, the end user attains familiarity with the current prototype and the project goals and possibilities, and is in a better position to formulate requirements and use cases.

In case the end users are available for a longer period of time, the first morning can be devoted to a requirements elicitation workshop, followed by a first round of evaluation in the afternoon. The consolidation and initial write-up of end user requirements can then be done by the team in the evening, and a second plenary requirements elicitation round can be held on the next day. During that second round, the results of the first-day requirements workshops are refined taking the evaluation results and experiences into account. The rest of the day can be used for in-depth interviews and technical discussion between end users and project members. The outcomes of a two-day end user workshop are:

- Evaluation results of the current prototype
- End user requirements and user stories based on discussion and feedback from end users
- Additional end user input into technical work packages

## 2.2 Requirements elicitation framework

The requirements gathering methodology in Optique is partly based on what is known as AGILE methodologies for software development.[1] A basic step of the process is the identification of user roles and recording of user stories. As a result, for effectively capturing user requirements in the Statoil and Siemens Work Packages, the following main steps are implemented:

1. Identification of user roles. (Section 3.1) talks about this process.
2. Requirements elicitation. Chapter 3 identifies the proper combination of methods for requirements elicitation to be used. The end user requirements are documented in terms of scenarios and user stories, taking into account different stakeholder perspectives and requirement category, grouped into different categories.
3. Recasting the end user requirements from the previous step into technical requirements directed at Optique's technical work packages. Technical requirements can be expressed in terms of implementation strategy, but should reference the end user requirements so refinement and disambiguation of technical requirements during implementation can be done taking the context of the original end user need into account.

[1] `http://agilemanifesto.org/`

## 2.3 Usability and performance evaluation framework

Evaluation of each year's Optique system prototype is done according to two criteria: usability and performance.

Usability is evaluated in cooperation with the end users, using standard techniques such as Heuristic Evaluation, Thinking Aloud Method and Structured Evaluation. Usability evaluation takes place during the end user workshops in conjunction with the requirements gathering process. Performance evaluation is done by project members against criteria as set forth in the DoW and/or decided by the scientific project leadership.

The results of the evaluation process are collected and reported in the annual Work Package 1 reports (D1.1 and D1.3–D1.5).

# Chapter 3

# Requirements Elicitation

The requirements elicitation phase of Optique is crucial for setting the directions towards which the project works each year. For capturing user requirements, the objective is to use an optimal combination of methods from the relevant literature and record the requirements in a clear and concise way that will guide the compilation of system specifications.

The detailed methodology to achieve this is explained in this chapter. The following paragraphs focus on the requirements elicitation methods to be used in the context of the Optique project. It has to be noted that the methodology given in this chapter may still be adjusted by the requirements team should new needs of the project be identified.

## 3.1 Identification of User Roles

Optique, as a complex and innovative software system, has multiple types of users with different experiences, backgrounds and goals while using the software. Delivering requirement specifications without a clear reference to the type of user that issued the requirement leads to problems in grouping and especially prioritizing them. Therefore, it is of huge importance to clearly identify the types of users, called user roles in the sequel, and to unambiguously link them to the requirement specifications.

User role identification is performed in the course of requirements analysis, and is used to plan end user workshops and give context to end user requirements. For an overview of user role identification, see, e.g., [1, Chapter 4].

After the main user roles are identified, users should be briefed according to their roles on the project vision and expected accomplishments. User roles in the Statoil and Siemens Work Packages include domain experts, business managers, and technical staff.

It is important to identify early on a group of users from each role and each industrial partner that will be able to follow the progress of the project. These users with a deep understanding of the project requirements, objectives, and limitations will complement the feedback received by the less involved user groups during evaluation cycles with insightful comments and suggestions as well as comparisons with earlier stages of the developed software.

## 3.2 Gathering the requirements

Typical requirements gathering and analysis methods as described in [6] and [5] include:

**Surveys** , both open-ended and focused, conducted electronically or on paper;

**Interviews** , typically conducted face-to-face, to find out what the users' priorities are;

**Focus groups** , which are useful for discussing possible user requirements and brainstorming ideas;

**Field studies** , i.e., observing the end-user situation and the environment in which a new system will be used (useful in understanding user needs);

**Evaluation of existing products** , which reveals and clarifies good and bad aspects of current solutions – valuable input to new design work;

**Task analysis** , giving a deeper analysis of users work with a system (useful for analysing how users' work tasks should be supported by functionality in a system);

**User personas and usage scenarios** , meaning concrete and illustrative data about typical users, their characteristics, usage situation, tasks and goals – particularly useful in supporting early user interface design work;

**Formulation of usability goals and overall design criteria** , which helps focus and steer the design process, supporting the evaluation of early concepts, prototypes and final designs.

In the Optique project, some users have a constant and active participation in all phases of design and development and are available to the requirements elicitation team for interviewing and consulting when it is needed, while more end users are available during the annual requirements workshops. Taking this fact into account, the following most appropriate methods applied for requirements elicitation will include:

- Interviews. For each workshop, the following need to be decided: how many end users; time and location of the interviews; form of the interview (free-form or semi-structured with a prepared questionnaire).
- Focus group meetings: For each workshop to define: time and location; number and profile of participants.

Interviews and focus groups are organized in 1 or 2 day workshops, held at the industrial partners' premises, in order to facilitate communication between the Optique users and domain experts and the Optique technical partners.

The requirements elicitation process is run in multiple iterations, foreseeing one cycle of requirements, development and evaluation for each year of the project. Requirements and evaluation workshops are combined in one event in order to facilitate the smooth transformation of evaluation results to requirements for the next iteration of development.

At the end of each requirement elicitation cycle, the user requirements in the Statoil and Siemens Work Package are recorded in detail. In addition, the technical means of meeting the user requirements are documented for technical partners so that the strictly scientific and technical project work can proceed.

## 3.3 Recording the requirements

The requirements recording methodology applied in Optique is based on several well-known and widely used methodologies such as the AGILE methodologies [2] and the VOLERE templates [1]. In short, the first step is to capture requirements at the level of user stories, known from AGILE software methodologies and described in several works such as [2], followed by an analysis of the stories to produce specific use cases with their corresponding user requirements. This step is done during the annual end user workshops, time permitting.

The stakeholder requirements are then documented in terms of use case scenarios and user stories, taking into account different stakeholder perspectives and requirement categorie, and are grouped into appropriate categories.

The next step is to recast the end user requirements, initially expressed in scenarios and user stories, into technical requirements directed at Optique's technical work packages. In the context of this phase it is also important to establish a user performance baseline for each industrial partner, enabling meaningful identification of improvements due to Optique when the user evaluations have been concluded. For this, a

small number of key performance indicators for each industrial partner are identified, and it is determined what levels these will need to reach in order for us to achieve the project goals. The evaluation process is detailed in Chapter 4.

The following sections provide a description of the proposed methodology including templates for the various recording phases of the process.

## 3.4 User Stories

User stories are short natural language descriptions of the intended functionality of a software system, written in the language of the user or stakeholder of the system. As described in [2], user stories are composed of three aspects:

- a written description of the story used for planning and as a reminder
- conversations about the story that serve to flesh out the details of the story
- tests that convey and document details and that can be used to determine when a story is complete.

As suggested in [2] and proven useful in several other works on requirements, a user story might follow the template:

> "As a <*user*>, I want <*something*> so that <*benefit*>."

Stories following that template usually contain some relevant information that might sometimes be missed, namely:

1. Which user role / actor issued the requirement (<*user*>) and
2. What is the rationale behind the requirement (<*benefit*>).

An example of a user story captured for Optique is:

> "As a Statoil *exploration expert*, I would like to *discover stratigraphy data for a particular geographic region over certain time periods* so that I can *bring it into my ArcGIS tool to work with it.*"

Furthermore, there can be different levels of details for user stories. Although, in general, user stories should be kept small in terms of the functionality to implement, there can be generic user stories that capture whole use cases in a single sentence. Such stories are often called epics. For example, a generic user story or epic derived from the Optique use cases is "As an end user, I want to search for stratigraphy data." Obviously, this is a rather generic user story capturing a whole use cases and therefore, a huge set of details is missing such as what are the search criteria (thematic, spatial or temporal or all of them), is it possible to store the queries, etc. Such details can be captured in additional low-level detailed user stories derived from the epic.

In literature on user stories and AGILE software development methodologies such as [2] or [7], it is suggested that good user stories should follow the INVEST model proposed by Wake [4]. The INVEST acronym stands for independent, negotiable, valuable, estimable, small and testable user stories.

### 3.4.1 User story templates

User stories are recorded using templates provided by the Information Workbench installation hosted by fluidOps. The template comprises a number of fixed elements and a longer, free-form requirements description:

The main part of a user story is recorded in free text format and can be as extensive as needed to adequately describe the story situation and rationale. A summary table like the one in Figure 3.2 is provided by the Information Workbench to provide an overview of the gathered stories.

| | |
|---|---|
| ID | An automatically generated identifier, for cross-referencing between user stories and use cases |
| Title | A user story title, to be presented in summary views |
| Source | The industrial partner where this user story originates, i.e., Siemens or Statoil |
| Tags | A set of tags for categorizing the user story |
| Description | A free-form description of the user story |
| Use Cases | A set of references to use cases connected with this user story |
| Requirements | A set of references to requirements connected with this user story |

Figure 3.1: User Story Template

| User Story Title | User Story Description | User Story ID | Actors involved |
|---|---|---|---|
| | | US_01 | |
| | | US_02 | |

Figure 3.2: User Stories Summary List

## 3.5 From User Stories to Use Cases

As described in [6] and extensively discussed e.g., at several websites such as [5], user stories differ from use cases. While user stories describe desired functionality of a system in natural language, use cases often follow a formal structure (template) and cover details such as preconditions, steps in a success scenario, or processed data. However, the relationship between user stories and use cases is not simply a generic matter of detail but highly depends on the scope of a user story. In some cases, a single user story might be transferred to a whole use case specification. For example, the user story or epic "As an employer, I want to post a job offering" covers a whole use case. Instead, more detailed user stories such as "As an employer, I want to put details on the job offering such as start date, payment and qualification" do not cover a whole use case but only a single step in the main success scenario.

Therefore, the process of transferring user stories to use cases depends on the scope of the stories. A successful approach typically involves the following steps:

- Identify user stories that cover functional requirements on the system
- Group the user stories and identify related user stories and the nature of the relationship such as generalization, specialization.
- Identify user stories that can be transferred to whole use cases and those that correspond to parts of use cases.

In Optique, the use cases and requirements are captured in the annual deliverables for Work Packages 1, 8 and 9. The use case descriptions will be derived from the user stories collected at the user community workshop. The use case specifications roughly follow the formal template based in [1], as given in Figure 3.3 and the methodology for deriving use cases from user stories given in this section.

## 3.6 Requirement Categories

This section introduces the requirement categories for Optique. The categories for non-functional requirements serve as a basis for categorizing and identifying the non-functional requirements for the industrial partners in Work Packages 8 and 9.

| | |
|---|---|
| Name | A natural language identifier of the use case |
| Identifier | The unique identifier for the use case in the form UC<*number*>, for example UC1 for the Optique Use Case 1. |
| Version | Specified in the form <*Version*>, <*Date*>.<br>Example: 1.0, 23.02.2007<br>Each minor change (changing descriptions within the use case) will be reflected by an increase by 0.1 of the version number. |
| Description | A short natural language description of the use case. |
| Primary Actor | A role name or description for the primary actor. |
| Stakeholders & Interests | A list of stakeholders and key interests in the use case |
| Pre-conditions | A description of any relevant precondition that must hold before the use case can be executed. |
| Success End Condition | A description of the result if the use case is successfully executed |
| Failure End Condition | A description of the result if the use case is not successfully executed |
| Trigger | The condition(s) that start the use case |
| Main Success Scenario | A numbered list of steps from trigger to goal delivery, and any cleanup afterwards |
| Scenario Extensions | Any alternative steps that might occur in the main success scenario |
| Exceptional situations | Any steps for handling exceptional situation that might occur durcing the main success scenario |
| Processed data | A description of the data types relevant / processes in this use case. |
| Generated data | A description of the data generated by executing this use case and, e.g., whether it is persistent or not. |
| Related User Stories | A reference to related user stories using the user story IDs |

Figure 3.3: Use Case Template

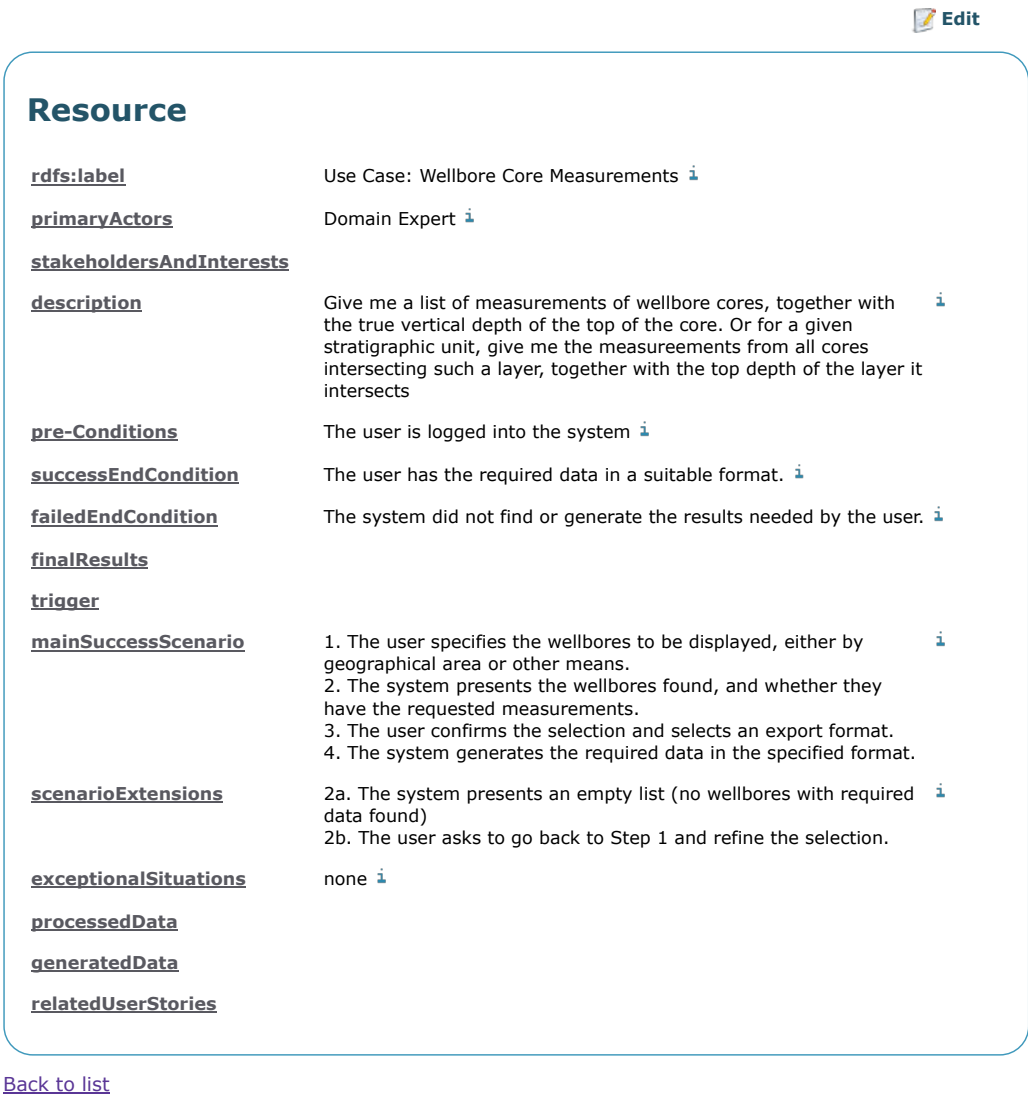


Figure 3.4: Use Case Example

### 3.6.1 Functional requirements

Functional requirements (FU) refer to requirements on what the system is expected to do. Functional requirements "present a complete description of how the system will function from the users perspective. They should allow both business stakeholders and technical people to walk through the system and see every aspect of how it should work – before it is built." [3].

Different templates for expressing functional requirements exist. Although for the functional requirements in some projects the use case recording method is considered sufficient, for Optique we propose additionally a more formal listing of the functional requirements (IEEE-styled requirements like "The SYSTEM shall. . . ") to be listed along the non-functional ones.

### 3.6.2 Non-functional requirements

Non-functional requirements can be grouped into the following categories:

**Performance requirements (PR)** Performance requirements typically relate to the response time of a system and the amount of data the system is expected to deal with. For Optique, initial performance requirements are already indicated in the technical annex of the description of work.

**Reliability requirements (RR)** Reliability requirements usually relate to the availability and reliability of the system. Typical examples are: "The system should be up at least 90% of the time."

**System interface requirements (SIR)** Interface requirements refer to the software interfaces offered by the system. A typical system interface requirement is "The system should offer all mandatory operations of the OGC Web Feature Service standard version 1.1.0."

**Security requirements (SR)** Security requirements relate to access restriction to the system.

**Standard requirements (STR)** Any requirement related to standards (OGC, ISO, etc.) can be grouped into this category.

**Human-Machine interface requirements (HMR)** Human-Machine Interface requirements relate to requirements on the user interface offered by the system.

**Documentation requirements (DR)** Documentation requirements capture those requirements that are related to the software documentation such as "The user manual should be delivered in English and Norwegian."

## 3.7 Listing the user requirements

It is important to establish and document the user requirements so that they lead into the process of designing the system itself. Numerous resources exist with general guidance on how to best specify user and organisational requirements and objectives [5]. For requirements to remain consistent in form, to enhance readability, and to facilitate further processing, we list requirements separately rather than in the commonly used tabular form. Each requirement has been marked with an ID, the name of the requirement, a priority designation, a description of the requirement, the source of the requirement and the requirement's rationale. Specifically,

**Requirement ID** takes the form of Ryyyy.nn, where yyyy is the project year (2013–2016) and nn an ascending number.

**Requirement Name** is the short title of the requirement.

**Requirement Priority** is the initial desired priority defined for each requirement, so as to facilitate further project development decisions by ranking importance. We define priority as low, medium or high.

| R2013.01 | Search stratigraphy in specific regions | High |
|---|---|---|
| Requirement ID | Requirement Name – Title | Requirement Priority |

**Description:** The exploration expert would like to be able to select multiple regions on a map when defining the search criteria for a stratigraphy data search. (FU)

**Source:** Statoil first focus group meeting.

**Rationale:** ...

| R2013.02 | ... | High |
|---|---|---|
| Requirement ID | Requirement Name – Title | Requirement Priority |

**Description:** ...

**Source:** ...

**Rationale:** ...

Figure 3.5: User requirements list

**Requirement Description** is a text describing the requirement, including its category.

**Requirement Source** indicates the source from which the requirement was derived (examples).

**Requirement Rationale** includes any other comments related to the requirement.

## 3.8 Traceability of requirements

An important aspect is the traceability of the requirements. Different common characterizations of traceability exist, such as

> "Requirements traceability refers to the ability to describe and follow the life of a requirement, in both forwards and backwards direction (i.e. from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases.)"[1]

or

> "Requirements traceability refers to the ability to define, capture and follow the traces left by requirements on other elements of the software development environment and the trace left by those elements on requirements." [2]

Traceability in Optique is enabled by the use of supporting systems (an Optique installation or version-controlled documents) and by providing clear references between all the different requirement related issues such as user roles, user stories, use cases and requirements: Within this system, user roles are linked to the user stories. Further, user stories will be linked within the formal use case and non-functional requirements specifications.

[1] http://en.wikipedia.org/wiki/Requirements_traceability#cite_note-5
[2] ibid.

# Chapter 4

# Usability and Performance Evaluation

This chapter describes a general framework for carrying out usability and performance evaluation of the Optique software. The framework defines a detailed process according to which requirements will be used to evaluate technical project results.

*System performance assessment.* Quantitative performance measurements are foreseen in the framework to meet user requirements for data retrieval tasks. General expectations are already anticipated in the Description of Work (see the description of Task 1.4 for key performance indicators), but might be refined by end-users.

*Usability assessment.* The objective of this phase is to perform user evaluation, collect and analyse the results, compare to stated requirements and document the degree to which the results meet the requirements. This will be combined with reflection on the results to assess how the solution has affected individual and organisational behaviour, and on the applicability of the solution to other applications.

## 4.1 Usability assessment

The usability testing framework ensures a thorough user evaluation that provides meaningful feedback in every stage of the project. This framework will be adjusted for the two industrial partners as user evaluation is carried out in parallel for Siemens and Statoil within Task 1.4. The usability assessment process produces joint user evaluation reports for the annual evaluation cycle and provides feedback to requirement analysis Task 1.3.

An important overall objective of the usability evaluation phases to be carried out each year is to achieve a significant degree of co-constructed learning of the end-user and the IT-experts for each use case. By tracing changes in end-user behaviour over time, the end-users' growing knowledge of technology opportunities will be reflected in renewed requirements. The learning is related to an increased understanding of the situation and opportunities of different user groups, ideally resulting in increased collaboration between these user groups. We will evaluate this by tracing changes in the behaviour of end-user expert groups. To this end, we will try to keep the the group of experts that test the Optique solutions in the course of the project relatively constant. The results of the evaluation will be thoroughly analysed in order to provide a list of problems and suggestions to the requirements phase. The usability evaluation report will be included in the Joint Report deliverables (D1.3–D1.5).

The usability tests are conducted by a team of evaluators in the premises of both industrial partners. Afterwards, the team prepares the analysis and reports on the conclusions that feed back to the requirements phase of the next cycle. We shall employ a combination of methods that can be applied during the evaluation phases of the project. This approach follows two main directions:

1. User performance measurement for interactive tasks combined with the Think Aloud Protocol and interviews

2. Heuristic evaluation and focus groups for usability analysis

### 4.1.1 User performance measurement

User performance measurement takes place in a controlled setting where the users, after having received an appropriate amount of training, are asked to perform specific tasks and their performance and reactions are recorded. The tasks to be used during the evaluation are defined beforehand, taking into account the end user needs and interests as discovered during previous requirement elicitation rounds.

End users will be briefly trained in the use of the system and then asked to complete the predefined tasks. To aid with the subsequent interview, we combine this structured testing approach with the Thinking Aloud Protocol method, which means that the user will be asked to verbalise their thoughts, comments, emotions while working on the predefined tasks. The process is observed and recorded by one or more evaluators.

After the completion of the tasks during the user performance measurement, project staff conduct a semi-structured interview.[1] The discussion will be directed by a questionnaire designed for this purpose where the user is asked to rate several aspects of the experience and discuss positive issues, problems and suggest solutions. The interviewer will also follow up on issues expressed by the end user via the Thinking Aloud Protocol during the evaluation process.

The users are also asked to compare how they performed the task without Optique. To have a baseline for comparison of the Optique user performance measurement, users will be asked to work on specific queries with their current method of work.[2]

Task completion times, usability issues, user comments as well as evaluator observations, combined with interview data collected during the requirements phase are assessed and compared to the results of the performance measurement sessions of the Optique system.

### 4.1.2 Heuristic evaluation and focus groups for usability analysis

In addition to the laboratory-style performance measurement evaluation described in Section 4.1.1, we will collect additional feedback via a heuristic evaluation approach. Heuristic evaluation involves a group of users who are asked to freely experiment with the system, to define their own queries and explore the limits of the system. The users are given certain guidelines as to which aspects and features of the system they should evaluate. Evaluators may be present during this phase; however their role will not be to interact with the users but rather to observe them as they are going about naturally using Optique to perform their tasks.

After completing the heuristic evaluation stage, users are asked to participate in one or more focus group sessions. With the evaluators as coordinators, the participants discuss the experience using Optique and identify possible positive as well as negative issues. Special emphasis is given to the comparison with working practices without Optique. Again, users are asked to provide ratings of the various usability aspects of the system, through questionnaires, which include the comparative ratings to the performance of the same tasks before Optique.

Heuristic Evaluation works best with users that are sufficiently trained and familiar with the Optique tools so as to be able to use the system without the guidance or intervention of the evaluators. Hence, this evaluation method will be given more emphasis towards the end of the project.

As already noted for the requirements elicitation section, evaluation may be combined with the requirements elicitation in joint focus group sessions or workshops.

### 4.1.3 Key usability indicators

The following key performance indicators, taken from the task description of T1.4, are to be taken into account for the evaluation of the integrated Optique functionality and services:

**User satisfaction** User satisfaction determines a subjective estimate for ease of use and result quality as given by the expert.

[1] This interview doubles as part of the requirements gathering process during one-day end user workshops.

[2] Note that this may not be possible in all cases, as the goal of Optique is to enable users to write queries that they are unable to write at present.

**Query definition time** The level of automation that Optique will offer is expected to reduce this time from weeks to hours or minutes.

**Query answering time** For retrieval times this is expected to be in the range of minutes, or (for complex queries) not significantly larger than pre-existing handcrafted queries.

**Number of continuous queries the system can handle for stream-based data** This is expected to be in the low thousands with a system running queries that refer to both temporal and static data.

**System response time for interactive tasks (query formulation support, ontology augmentation, etc)** This time is expected to be under 1–2 seconds. For verification tasks, such as ontology consistency checks, however, longer times, even at the range of tens of seconds, are acceptable.

**Query language adequacy** Whether the query language includes all the features necessary to capture the users data access requirements is evaluated by means of user studies that compare expressivity w.r.t. query execution times and query formulation times to be systematically acquired.

**Query formulation adequacy** It expresses whether support for end-user (domain expert) query formulation is sufficient for realistic real world tasks required in the existing business work flows.

**System administration adequacy** It investigates whether there is an adequate range and efficacy of system administration tools for use by the IT experts. This includes ontology and mapping management tools as well as tools to tune and optimise the Optique platform.

Additional indicators may be defined during user workshops or focus groups meetings if the need arises. Issues to be also investigated include how query answering can benefit from new ontology axioms and the impact of better data (generated with less effort by Optique) for statistical analysis tools being part of the workflow of business processes.

## 4.2 System performance measurement

System performance evaluation within Optique will be based for both use cases on the query catalog established in Task 8.2 with respect to temporal and continuous queries and on the query catalog established in Task 9.2 for static queries. The aspects to be tested are:

1. Correctness (soundness) of the implementation of the algorithms
2. Correctness of the integrated system.
3. Completeness of the query results
4. Performance

The system performance measurement evaluation will be performed by Optique project members in the context of Work Packages 8 and 9.

### 4.2.1 Key performance indicators

The aspects to be tested are:

**Query answering time** For Work packages 8 and 9, an environment is set up such that values for various optimisation parameters can be automatically selected in a batch evaluation setting. This procedure allows for automatic and repetitive testing without errors being counted and reported (regression testing). Expected results are defined, and hence correctness and completeness are checked automatically as well for various modes of operation. Query answering time is expected to be in the range of minutes, or (for complex queries) not significantly larger than pre-existing queries.

**Number of continuous queries** An evaluation system is designed such that the number of registered queries can be gradually increased, while values for various optimisation parameters are automatically and systematically selected. Changes in data stream characteristics are automatically generated from pre-recorded streams such that repetitive evaluation of the same situation under various settings is possible. The number of continuous queries a system can handle while keeping pace with data input rates should be at least in the order of low thousands of registered queries.

**Memory consumption for indices and temporary data generated by query answering** Similar techniques as used for answering times and number of continuous queries apply. Automated comparison with previously generated reports ensures measurement of project progress, and, as a consequence, development efforts can be effectively controlled. The effect of statistical analysis algorithms for detecting events to be materialised (selective materialisation) by inspecting data and query logs can be investigated as part of the regression testing evaluation technique.

# Bibliography


[1] Alistair Cockburn. *Writing Effective Use Cases*. Addison-Wesley, 2001.

[2] Mike Cohn. *User Stories Applied: For Agile Software Development*. Addison-Wesley, 2004.

[3] IEEE Computer Society. Software Engineering Standards Committee and IEEE-SA Standards Board. IEEE recommended practice for software requirements specifications. *IEEE Std. 830-1998*, 1998.

[4] Judy Hammond, Tom Gross, and Janet Wesson, editors. *Usability: Gaining a Competitive Edge*, volume 226 of *IFIP Conference Proceedings*. Kluwer, 2002.

[5] Scott McEwen. Requirements: An introduction. `http://www.ibm.com/developerworks/rational/library/4166.html`, April 2004. Accessed: Jan 15, 2013.

[6] Jacob Nielsen. *Usability Engineering*. Academic Press, 1993.

[7] Bill Wake. INVEST in good stories, and SMART tasks. `http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/`, August 2003. Accessed Jan 15, 2013.

# Glossary

**API** Application Programming Interface

**DoW** Description of Work

**DR** Documentation Requirements

**HMR** Human-Machine Interface Requirements

**PR** Performance Requirements

**RR** Reliability Requirements

**SIR** Security Requirements

**STR** Standard Requirements

**UC** Use Case