REACTION

Remote Accessibility to Diabetes Management and Therapy in Operational Healthcare Networks

REACTION (FP7 248590)

D2-8 The requirement engineering process

Date 2012-02-28

Version 3.0

Dissemination Level: Public

Table of Content

1. Executive summary 4

2. Introduction 6

2.1 Purpose, context and scope of this deliverable 6
2.2 Background 6

3. Research and development methodologies 7

3.1 Human-centred software design 7
3.2 Software engineering process 8
3.3 Overview of the iterative approach 9

4. Prototyping cycles 10

4.1 Rapid general ward (in-hospital) closed-loop prototype (M12) 10
4.2 Primary care closed-loop prototype and extended general ward prototype (M24) 11
4.3 Prototype of general ward and primary care applications (M36) 11
4.4 Final prototype with automatic glycaemic control and closed-loop feedback (M48) 11

5. Initial requirements elicitation and collection 12

5.1 Vision scenarios and use cases 12
5.2 Requirements specification and engineering 12
5.3 Architectural specification 15
5.4 System integration and verification 16
5.5 Development of user applications 16
5.6 Validation of prototypes 16
5.7 Field trial usability testing activities 17

6. Re-engineering of requirements 18

6.1 Lessons Learned 18
6.2 Requirements improvement opportunities and change request reports 20

7. Conclusion 21

8. References 22

Document control page

Code	D2-8_2012-02-28_The_Requirement_engineering_process_V30_IN-JET.doc
Version	3.0
Date	2012-02-28
Dissemination level	PU
Category	R
Participant Partner(s)	IN-JET, FORTH-ICS
Author(s)	J. Thestrup (IN-JET) H. Udsen (IN-JET), F. Chiarugi (FORTH-ICS)
Verified and approved by	
Work Package	WP2
Fragment	No
Distribution List	All
Abstract	This deliverable contains a description of the methodology used in the requirement engineering process in the REACTION project.
Comments and modifications	This document has been renamed from *ID2-8-1 The requirement engineering process* and has been recategorised as an external deliverable
Status	□ Draft ☒ Task leader accepted ☒ WP leader accepted ☒ Technical supervisor accepted □ Medical Engineering supervisor accepted □ Medical supervisor accepted □ Quality manager checked □ Project Coordinator accepted
Action requested	□ to be revised by partners involved in the preparation of the deliverable □ for approval of the task leader □ for approval of the WP leader □ for approval of the Technical Manager □ for approval of the Medical Engineering Manager □ for approval of the Medical Manager □ for approval of the Quality Manager □ for approval of the Project Coordinator **Deadline for action**: N/A
Keywords	Requirement engineering
References	
Previous Versions	*ID2-8-1 The requirement engineering process*

	Version	Author(s)	Date	Changes made
Version Notes	1.0	H. Udsen	2011-04-15	First draft
	1.1	J. Thestrup	2011-05-12	Reviewed and updated
	1.2	H. Udsen	2011-05-18	Minor revisions
	2.0	H. Udsen	2011-06-30	Final version – Document has been renamed
	2.1	F. Chiarugi, H. Udsen	2012-01-27	Sections 1, 2.1 and 5.2 now describe revised JIRA workflow
	3.0	H. Udsen	2012-02-28	Reviewer comments addressed. Final version submitted to the European Commission

	Reviewed by	Date	Comments made
Internal review history	Peter Rosengren	2012-02-25	Approved with comments
	Manuel Perez (V1.2)	2011-06-01	Approved with comments

1. Executive summary

This document describes the work performed with the aim of reviewing, analysing and clarifying the iterative requirements engineering process adopted by the REACTION project. Although the process has been carried out as planned in the DOW, it became clear that partners found it difficult to maintain a sufficient overview of the steps in the process because the documentation was fragmented and included in a number of different documents. Hence, there was a need for a comprehensive and detailed description of the iterative requirement engineering process.

In accordance with generally accepted principles, the REACTION project has adopted an evolutionary requirements engineering, specification and design methodology underpinned by a strong user-centric development process.

The methodology calls for comprehensive iterative requirements and stakeholder analysis based on initial requirements gathered from medical and clinical scenario thinking. These requirements would encompass the needs and priorities of the users as well as the wider exploitability and scalability requirements taking into account the technical constraints as well as the safety, socioeconomic and legal acceptance and the deployability of the resulting REACTION platform in real Public Healthcare Systems.

The starting point of the iterative design process is a set of domain-specific vision scenarios delivering end user visions of applications in three different insulin therapy domains: In-hospital, primary care and Automatic Glucose Control (AGC). The scenarios were supplemented by including two interview and focus group sessions with the aim of identifying the present and future clinical workflows in both in-hospital and primary care settings.

The vision scenarios and workflows are used to derive detailed technically and clinically oriented use cases that are discussed in focus groups with stakeholders. The result of this work is an initial set of requirements specifications for the REACTION platform and applications expressed in the Volere template. From the initial set of requirements, the technical experts will specify the initial architectural specification which then drives the research and development work and system integration.

At the end of each iterative cycle (corresponding to one calendar year) a prototype of the REACTION platform is assembled with a view to integrating as many of the existing components as are available at the time and in accordance with the detailed work plan. The prototypes will be developed with the specific purpose of illustrating the following aspects:

1. Rapid prototype of closed-loop system to be used in the general ward
2. Prototype of primary care closed-loop system and extended closed-loop system used in general ward (including sensor prototypes)
3. Partly/fully functional prototypes of general ward and primary care applications including multi-parametric monitoring, risk analysis and full back end interoperability
4. Automatic glycaemic control with closed-loop feedback directly to insulin dosage pumps and field trials with final prototypes

Towards the end of a complete development cycle, the components of the REACTION platform will undergo integration and technical verification of its functionality. System integration and verification will take place in each of the four iterative validation cycles in the research and development phase.

After the successful completion of a prototype cycle, each RTD work package will analyse and report their development results, RTD experiences, Lessons Learned in the development and integration work and other relevant knowledge gained during the development cycle. Lessons Learned constitute both individual and organisational knowledge gained by experience; either negative or positive.

All results from validation and experiences gathered in the process will lead to refined technical scenarios, revised requirements specifications, updated platform architecture and new prototype specifications. The new and updated requirements will be scrutinised and prioritised by the Technical Manager and relevant WP leaders and the re-engineered requirements will be documented in internal deliverables *ID2-8-2 Change request and re-engineering report 1*, *ID2-8-3 Change request and re-engineering report 2* and *ID2-8-4 Change request and re-engineering report 3*.

At the end of the first year a thorough WP2 analysis suggested the possibility of improving the management of the requirements engineering process through the adoption of a revised workflow in the JIRA requirements database.

This revised workflow comprises 6 Statuses and 7 Resolutions for resolved/closed requirements:

- Statuses: Open, Part of Specification, In Progress, Resolved, Closed, Reopened
- Resolutions: Implemented, Cannot be implemented, Validated, Duplicate, Conflicting, Nonsense, Out of Scope

2. Introduction

2.1 Purpose, context and scope of this deliverable

This document describes the work performed with the aim of reviewing, analysing and clarifying the iterative requirements engineering process adopted by the REACTION project. During the first year of the project, the requirements process was implemented by WP2 partners IN-JET and FORTH-ICS.

The iterative requirements engineering process was outlined during the project kick-off meeting and revisited at subsequent project meetings. The planned activities, such as formulation of scenarios and use cases, requirements elicitation and gathering, engineering of the requirements, setting up of a validation framework and collection of Lessons Learned were all undertaken during the course of the year. The necessary online tools supporting the process were put in place. These tools include a JIRA repository for requirements gathering and a TWiki for collecting Lessons Learned,

Although the process has been carried out as planned, it became clear that partners found it difficult to maintain a sufficient overview of the steps in the process because the documentation was fragmented and included in a number of different documents. Hence, there was a need for a comprehensive and detailed description of the iterative requirement engineering process.

At the end of the first year a thorough WP2 analysis suggested the possibility of improving the management of the requirements engineering process through the adoption of a revised workflow in the JIRA requirements database. It was also found advantageous to suggest some minor changes to the number and structure of the deliverables and internal deliverables in WP2, leading to the decision to rename this deliverable *D2-8 The requirement engineering process*.

The present deliverable thus describes and discusses the various procedures and workflows that the project will undertake in order to make the iterative requirements engineering process transparent, complete and comprehensive.

2.2 Background

The aim of the REACTION project is to develop an integrated ICT platform that supports improved long-term management of diabetes mellitus based on wearable, continuous blood glucose monitoring sensors and automated closed-loop delivery of insulin. Solutions will be developed for in-hospital use in general wards and for monitoring of diabetes patients in primary care.

The REACTION interoperable peer-to-peer communication platform will be based on a service-oriented architecture with all functionalities, including sensors and devices, represented as services and applications consisting of a series of services orchestrated to perform a desired workflow.

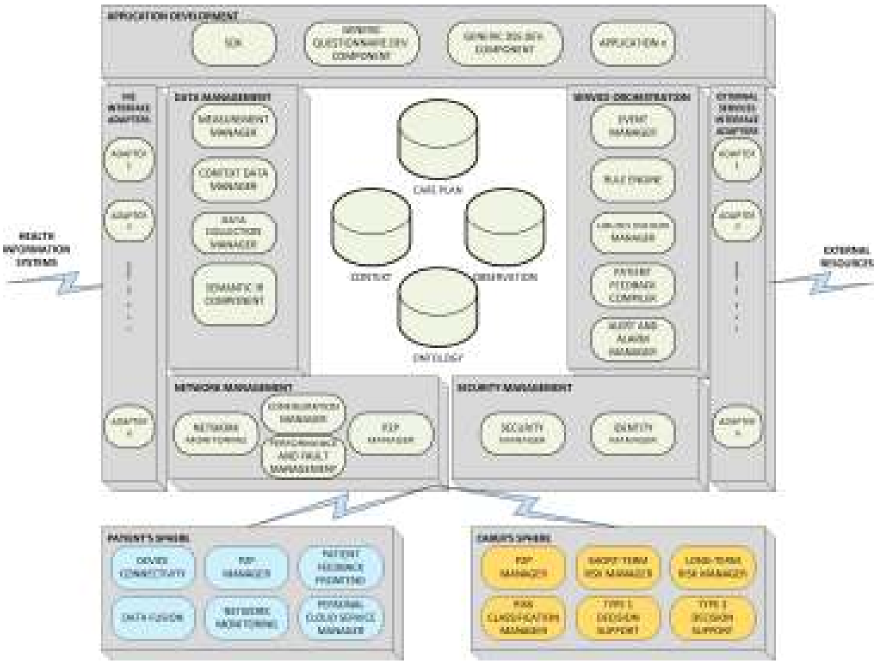


Figure 1 – Overall architecture of the REACTION platform

3. Research and development methodologies

The REACTION project seeks to use technology as a tool to foster human development, in order to use the great potential that new technologies might have for addressing major societal challenges in coping with the increasing number of citizens suffering from diabetes. The success of the new technological applications depends heavily on the acceptance from end users, i.e. patients, relatives and professional carers as well as the acceptance from healthcare commissioners, business stakeholders, and regulatory authorities.

In accordance with generally accepted principles, the REACTION project has adopted an evolutionary requirements engineering, specification and design methodology underpinned by a strong user-centric development, which complies with the following template in each iteration:

- User requirements engineering and refinement
- Architecture design specification and refinement
- Clinical protocol and medical context planning
- Technologies research and development to implement architecture
- Integration and prototype development and field trial preparation
- Field trials in clinical domains
- Conformance testing, usability evaluation and user acceptance testing
- Lessons Learned and change analysis

The methodology calls for comprehensive iterative requirements and stakeholder analysis based on initial requirements gathered from medical and clinical scenario thinking. These requirements would encompass the needs and priorities of the users as well as the wider exploitability and scalability requirements taking into account the technical constraints as well as the safety, socioeconomic and legal acceptance and the deployability of the resulting REACTION platform in real Public Healthcare Systems.

All work packages and subtasks are structured according to this methodology. The project follows a logical flow of analysis and requirements engineering, research and development of prototypes, and implementation and testing.

This section summarises the overall development methodologies adopted to achieve the project goals. The subsequent sections will look more closely into the implementation of the requirement engineering process.

3.1 Human-centred software design

Software engineering is an ongoing and iterative process of user involvement, requirement analysis, specification, implementation, and user evaluation. The REACTION project has adopted an iterative, human-centred design and development approach. The software development process starts with user-defined vision scenarios and future use cases providing the baseline for the initial requirements specifications. It continues with formalistic, iterative, and human-centred requirements engineering phases.

The requirement engineering process follows the principles of the ISO 13407 and its replacement ISO 9241-210 "Human-centred design processes for interactive systems" standard. This standard provides guidance on human-centred design activities throughout the life cycle of computer-based interactive systems. Essential for a human-centred design process is that expertise from a multi-disciplinary team greatly enhances the engineering process. Solutions are implemented in iterations and technology and user functions should be in balance. Users should be actively involved and their tasks should be clearly focussed.

One of the core tasks of user-centred design in REACTION is to negotiate and facilitate the communication across the well-known user-developer gap while acknowledging the different forms of expression and different requirements on each side. The literature has many examples demonstrating that end users have to bridge the large gap in understanding, especially in projects that apply a waterfall model. Clark, Lobsitz & Shields, (1989) show that evolutionary or iterative approaches greatly reduce this gap.

During the initial discussion of the project objectives and the work plan, it was decided, prompted by the arguments of the clinical partners, to align the REACTION platform functionalities with prevailing clinical practice and medical reality in order to close the gap between today's practice and the potential of the REACTION platform.

The user-centred design method is a cyclic process with no sharp start and end points: the context of use phase with intensive user involvement continues for the whole duration of the process with an emphasis on the beginning, and the requirements elicitation likewise extends well into the design proposal phase. There are four essential human-centred activities recommended in ISO 9241:

1. to understand and specify the context of use
2. to specify the organisational and user requirements
3. to produce design solutions
4. to evaluate the design against requirements

The human-centred design approach implies an iterative life cycle in the project. Iterative cycles allow advancing from specification to implemented prototypes, from experience and evaluation to improved specifications and improved prototypes. The design proposals are based on the current understanding of the context of use. These proposals provide an idea on how to meet identified or assumed requirements. The evaluations of the design proposals yield a richer understanding of the context of use and new or modified requirements and thus guide the evolutionary improvement of the design.

For both end users and developer users the design proposals can be understood as design probes to explore the characteristics and usefulness of the proposed system. When a prototype is available, end users can try it and gain personal experience with the system.

3.2 Software engineering process

The applied software engineering process in REACTION constitutes the basis for the future work within the work packages. The software engineering process does not prescribe specific methods to achieve these goals; they are to be chosen according to what state of the art is and what is appropriate under individual project circumstances.

The requirements serve as a reference to measure if the development within the project is always in line with the functionalities and properties desired by them. Based on practical experiences from other R&D projects, we have devised a scenario-based approach, combined with end user interviews and expert analysis, originating in the structure proposed by Robertson & Robertson (Robertson, 2006) for mastering requirements.

The Volere process recommended by Robertson & Robertson ensures that all important aspects of requirements are carefully addressed and that the methods applied have proven their value in practical work. Most importantly, a *fit criterion* is defined which operationalises the requirements and provides a measure to ensure that the requirements are met. The philosophy of Robertson & Robertson is very much in line with ISO 9241:210 "Human-centred design processes for interactive systems" and allows a structured processing of the requirements assuring that they remain always applicable and testable.

Scenarios are part of the system specification; they explicitly deal with the usage of a technical system, the context of use, and the allocation of function between the technical system and human users. Later, when a prototype is available, end users can try it and gain personal experience with the system. Iterative cycles allow advancing from specification to implemented prototypes, from experience and evaluation to improved specifications and improved prototypes. In REACTION four full cycles are planned throughout the project lifetime, aiming at validated prototype specifications, including concepts of usage.

3.3 Overview of the iterative approach

The starting point of the iterative design process is a set of domain-specific vision scenarios developed with the IDON method. The vision scenarios deliver end user visions of applications in three different insulin therapy domains: In-hospital, primary care and Automatic Glucose Control (AGC). The scenarios were supplemented by including two interview and focus group sessions with the aim of identifying the present and future clinical workflows in both in-hospital and primary care settings.

The vision scenarios and future clinical workflows are used to derive detailed technically and clinically oriented use cases that are discussed in focus groups with stakeholders. The result of this work is an initial set of requirements specifications for the REACTION platform and applications expressed in the Volere template.

From the initial set of requirements, the technical experts will specify the initial architectural specification which then drives the research and development work and system integration.

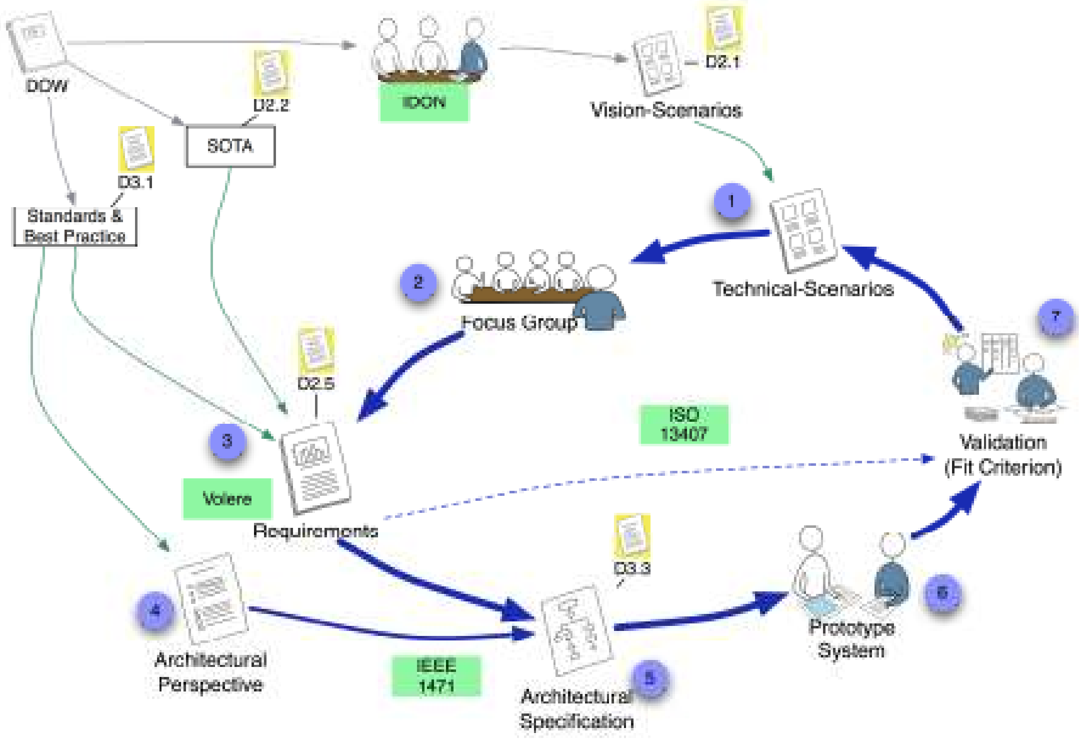


Figure 2 – Overview of the iterative software development approach in REACTION

At the end of each iterative cycle (corresponding to one calendar year) a prototype of the REACTION platform is assembled with a view to integrating as many of the existing components as are available at the time and in accordance with the detailed work plan. The prototype platform is used to implement applications for the two project use cases, including patient and clinician's applications. The prototypes will be demonstrated and validated in domain-specific settings with the aim to demonstrate the outcome of each cycle to end users, project partners, reviewers, the research community, industry leaders, potential customers, etc.

The clinical work will be undertaken in parallel to the technical development. It starts with the establishment of a set of clinical protocols in each of the domains. The protocols define and drive clinical field trials, which will be performed using the increasingly advanced platform prototypes during the course of the project. The field trials will be used for validation of the benefit provided for individual users and healthcare organisations and the usability aspects experienced by the users.

All results from validation and experiences gathered in the process will lead to refined technical scenarios, revised requirements specifications, updated platform architecture and new prototype specifications.

4. Prototyping cycles

The REACTION project aims to develop the platform and applications in a total of four cycles, each lasting one year. For each cycle, the system progressively gets more and more advanced as lessons are learned and more and more end user requirements are fulfilled. Evaluation of experiences from the previous cycles is taken into account. At the end of each annual cycle a prototype will be developed with the specific purpose of illustrating the following aspects:

1. Rapid prototype of closed-loop system to be used in the general ward
2. Prototype of primary care closed-loop system and extended closed-loop system used in general ward (including sensor prototypes)
3. Partly/fully functional prototypes of general ward and primary care applications including multi-parametric monitoring, risk analysis and full back end interoperability
4. Automatic glycaemic control with closed-loop feedback directly to insulin dosage pumps and field trials with final prototypes

At the end of the fourth prototyping cycle, a fully developed version of the REACTION platform, i.e. platform, applications and tools, will be available for demonstration and exploitation.

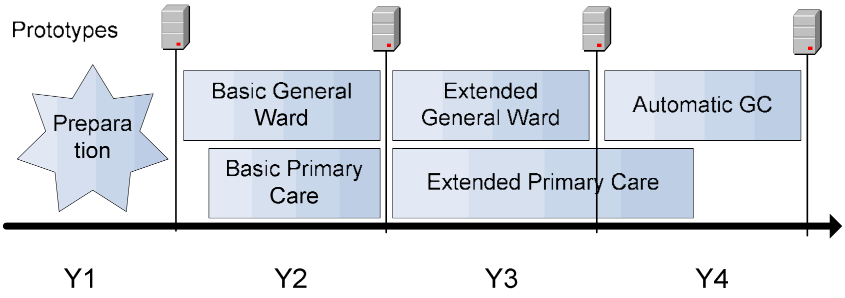


Figure 3 – Prototypes and field trials planning

The guides for developing each prototype are the technical scenarios based on the clinical protocols, the future scenarios and the future workflow descriptions. For each cycle, more and more scenario elements are implemented, thus making the prototype increasingly sophisticated as the project progresses. For each cycle, one user domain is in focus but scenario elements from the other two domains are also included and augmented in the prototype. Finally, in the fourth cycle, all domain scenarios will be fully demonstrated.

In this section the planned prototyping cycles will be explained. It should be noted that this is a premature planning description. The precise content of each annual prototype will be decided at the end of the previous cycle when the validation results and the change request analysis are available.

4.1 Rapid general ward (in-hospital) closed-loop prototype (M12)

The first prototype will be the "proof-of-concept-prototype" of the REACTION platform in an in-hospital setting. It will be based on use cases and scenarios from the general ward domain.

The intended application is to support glucose management in the hospital of the University of Graz (MUG) at two wards namely Endocrinology and in a further later step Cardiology. The system will assist glucose management for patients for whom it was decided by the physicians to perform glucose management.

Users of the system will be the medical staff (physicians and nurses) and technicians to maintain the system. Therefore the users can be considered as professional users.

The application will be a software solution which will run on servers but also on portable electronic devices (e.g. Tablet PC). The software enables the management of several patients in parallel in the hospital at the defined wards. The portable device will be carried to the point of care (i.e. to the patient) thus enabling the entering and retrieval of data.

In order to support in-hospital glucose management the system will have the following features:

- Automatic retrieval of data from the hospital information system (HIS) and the laboratory information system (LIS)
- Manual entry of data and information (e.g. comments)
- Visualisation of relevant information like glucose profile, medication (OADs, insulin), nutrition, health status, contextualised comments
- Reminder/alarms of missed measurements or out of range values
- Electronic Decision Support system for dosing of insulin
- Storage and archiving of data

The deliverable *ID2-6-1 Internal Prototype Application Specification 1* provides a technical description of the first prototype application including the scenario and technical solution to be demonstrated.

4.2 Primary care closed-loop prototype and extended general ward prototype (M24)

The second prototype will focus on the primary care sector with the aim of improving long-term management, with its associated impact on reducing the likelihood of developing complications (cardiovascular disease, ulceration, amputation, blindness).

In general, the approach to management of patients with diabetes in the community is different to the acute situation in the hospital. Careful monitoring of multiple parameters may represent a useful integrated basis for achieving strict and sustained glucose control that will provide a better opportunity to reduce complications and improve patients' quality of life. This primary care prototype will focus on simple closed-loop operation and integration with back end systems. The prototype will be installed and validated at the Chorleywood Health Centre (CHC).

New developments in the REACTION platform will also be used to update and extend the general ward prototype with new functionalities to be determined after the validation of the first year development has been performed. The specification of the second year prototype will be provided in the deliverable *ID2-6-2 Prototype application specification 2* due in M18.

4.3 Prototype of general ward and primary care applications (M36)

The third prototype will extend the primary care applications and include multi-parametric monitoring, risk analysis and full back end interoperability. Devices for glucose and physical activity monitoring will be used to determine whether multi-parametric monitoring provides reliable measurements compared to conventional mono-parametric monitoring.

The general ward prototype will be finalised with focus on assessing the sensor development and the accuracy of the CGM sensors in preparation for automatic glycaemic control with closed-loop feedback directly to insulin dosage pumps which is the subject of the final prototype

Based on the outcome of the second prototype cycle a time plan and allocation of responsibilities will be defined at the beginning of the third cycle.

4.4 Final prototype with automatic glycaemic control and closed-loop feedback (M48)

At the end of the project, a complete REACTION platform, including development tools capable of demonstrating closed-loop applications built to support the use cases and scenarios, will be demonstrated.

A reference implementation will be made on the REACTION platform, in which a CGM sensor will continuously measure the glucose level. Other data on treatment response will be collected from EPR and HIS systems and the agglomerated data will be fed to a computer algorithm in the Data Management component, which calculates the adjustments needed to insulin intake. The data are transmitted to a fluidic micro-pump simulating an IV insulin pump.

The entire system will be subject to vigorous risk assessment and security analysis and the evaluation is expected to establish the basis for more appropriate treatment guidance for designing actual AGC systems.

Based on the outcome of the third prototype cycle a time plan and allocation of responsibilities will be defined at the beginning of the fourth cycle.

5. Initial requirements elicitation and collection

5.1 Vision scenarios and use cases

The focal point of the iterative design process in REACTION is a set of domain-specific Vision Scenarios delivering end user visions of the future use of REACTION applications in support of improved long-term management of diabetes mellitus Type 1 and Type 2.

Creating scenarios of end user behaviour and interaction with platform functionality is an extremely useful instrument for identifying key technological, security, socioeconomic and business drivers for future end user requirements. The scenarios will provide the framework for the subsequent iterative requirements engineering phase.

Scenarios are snapshots of possible/alternative futures that help us deal with the inherent uncertainty. Scenarios provide coherent, comprehensive, internally consistent descriptions of plausible futures built on the imagined interaction of key trends. The purpose of Scenario Thinking is to challenge the preconceived notions people have of the future, or their maps, and to afford people the flexibility to change those maps.

The IDON Scenario Thinking technique was selected for development of vision scenarios in REACTION. The core of the IDON technique is to examine a set of wider environmental factors, ambiguities and uncertainties in order to resolve which role they are likely to play in the unfolding of scenarios. The IDON method consists of two parts: *Scenario Development* and *Scenario Deployment*.

The scenarios are developed in the *Scenario Development* part using experts and based on knowledge and systematic analysis. The aim is to develop four mind-challenging scenarios by mixing inevitable trends with creative fiction.

In the *Scenario Deployment* part, technical experts and project decision makers interpret the scenarios and extract a framework for the functional and trust and security requirement specifications.

A one-day workshop was organised to bring together appropriate expertise and experience in the fields of diabetes and healthcare ICT. The activities carried out include identification of uncertainties, grouping and segmenting and flip/flopping (grouping in main directions). After the workshop, scenes, trends and scripts for the scenarios have been defined. The outcome of this Scenario Thinking process is 4 equally plausible scenarios for the future use of the REACTION platform in 2020.

During the initial discussion of the project objectives and the work plan, it was decided to align the REACTION platform functionalities with prevailing clinical practice and medical reality in order to close the gap between today's practice and the potential of the REACTION platform. As a consequence of this decision it was agreed to include two interview and focus group sessions with the aim of identifying the present clinical workflows in both in-hospital and primary care settings.

The results of the interview/focus group sessions and the Vision Scenario workshop have been documented in deliverable *D2-1 Scenarios for usage of the REACTION platform*.

From the vision scenarios, a systematic formalisation of all relevant user requirements and subsystem functional, security and societal requirements will be derived. Functional user requirements specifications will involve the most important aspects of user expectations in the chosen application domains. The security requirements will be based on an analysis of the scenarios and formulation of trust and security perception. Societal requirements arise from correlating socioeconomic, regulatory and policy issues with the deployment and use of monitoring infrastructures.

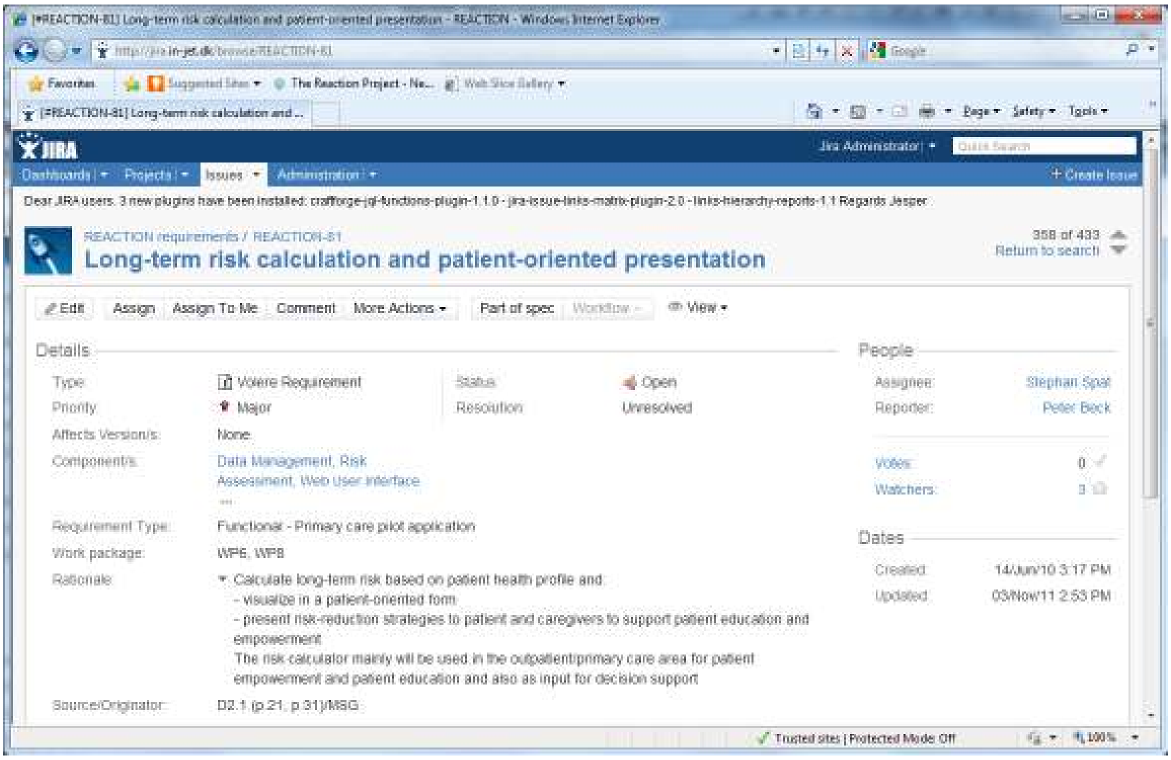
5.2 Requirements specification and engineering

The vision scenarios and the use cases were evaluated and discussed by developers in focus groups consisting of 7 to 10 developers and user representatives. These focus groups provided insight into the work of the targeted users and contributed to a detailed understanding of the emerging problems.

The analysis of scenarios and use cases enables the development of an initial set of requirements. The requirement process is based on the Volere requirements mastering process. The Volere process ensures that all important aspects of requirements are carefully addressed and that the methods applied have proven their value in practical work.

The Volere template ensures that the results are documented in a way that can be communicated efficiently to developers. The condensation was mainly done by abstraction, i.e. by eliminating redundancy in clusters of equivalent statements and phrasing the essential meaning in one statement. Assigning a rationale and an appropriate classification is done in several group discussions within the project team.

During the first months of the project, a JIRA tool has been installed and configured with the Volere template. JIRA is a web-based issue tracker that allows implementing and tracking a collaborative workflow and is used as a tool for gathering and sharing requirements amongst developers. Figure 4 shows a screenshot of JIRA with one open requirement.


[#REACTION-81] Long-term risk calculation and patient-oriented presentation - REACTION - Windows Internet Explorer
http://jira.in-jet.dk/browse/REACTION-81
Google
Favorites
Suggested Sites
The Reaction Project - Ne...
Web Slice Gallery
[#REACTION-81] Long-term risk calculation and ...
Page
Safety
Tools
JIRA
Jira Administrator
Quick Search
Dashboards
Projects
Issues
Administration
+ Create Issue
Edit
Assign
Assign To Me
Comment
More Actions
Part of spec
Workflow
View
Trusted sites | Protected Mode: Off
100%

Dear JIRA users. 3 new plugins have been installed: crafforge-jql-functions-plugin-1.1.0 - jira-issue-links-matrix-plugin-2.0 - links-hierarchy-reports-1.1 Regards Jesper

REACTION requirements / REACTION-81

Long-term risk calculation and patient-oriented presentation

358 of 433
Return to search

Details

Type: Volere Requirement
Status: Open
Priority: Major
Resolution: Unresolved
Affects Version/s: None
Component/s: Data Management, Risk Assessment, Web User Interface
Requirement Type: Functional - Primary care pilot application
Work package: WP6, WP8
Rationale: Calculate long-term risk based on patient health profile and:
- visualize in a patient-oriented form
- present risk-reduction strategies to patient and caregivers to support patient education and empowerment
The risk calculator mainly will be used in the outpatient/primary care area for patient empowerment and patient education and also as input for decision support
Source/Originator: D2.1 (p.21, p.31)/MSG

People

Assignee: Stephan Spat
Reporter: Peter Beck
Votes: 0
Watchers: 3

Dates

Created: 14/Jun/10 3:17 PM
Updated: 03/Nov/11 2:53 PM

Figure 4 – Screenshot if JIRA showing a single requirement based on Volere

The quality control in JIRA is realised by processing requirements along the steps of a workflow. Initially the default workflow of JIRA was adopted for the REACTION project and used during the first year of the project. The experiences and the lessons learned collected during this first year have suggested the introduction of a slightly modified workflow which is described below. Each requirement has a status that changes depending on the current workflow step. JIRA ensures that the quality of the requirements is always controlled by two persons. One person enters the requirement and the other person checks and passes the requirement through the quality gateway.

All requirements have to pass the quality gateway. This means that they are complete and all fields are sensibly filled in. A requirement that passed the quality gateway cannot (should not) be edited any more during the current iteration cycle. The last step is to decide whether a requirement becomes part of the specification or if it is not considered. The latter can be the case for different reasons: A requirement can be a duplicate of another requirement, it may be conflicting with (an)other requirement(s), it may be nonsensical or it may be out of the project's scope. Requirements resolved this way are assigned Resolutions 'Duplicate', ‘Conflicting’, ‘Nonsense’ or ‘Out of Scope’, respectively.

A requirement, which has been quality checked and deemed ‘Part of Specification’ is then assigned to a specific person (the Assignee) for implementation (Status ‘In Progress’) and management. When accomplished, the assignee changes the Status to ‘Resolved’ and applies the appropriate Resolution (‘Implemented’ or ‘Cannot be implemented’), the latter typically with a Comment explaining why. The final step for an implemented requirement can be validation, in general by internal or external end users.

To summarise, the REACTION workflow in JIRA comprises 6 Statuses and 7 Resolutions for resolved/closed requirements:

- Statuses: Open, Part of Specification, In Progress, Resolved, Closed, Reopened
- Resolutions: Implemented, Cannot be implemented, Validated, Duplicate, Conflicting, Nonsense, Out of Scope

The revised workflow is depicted in Figure 5.

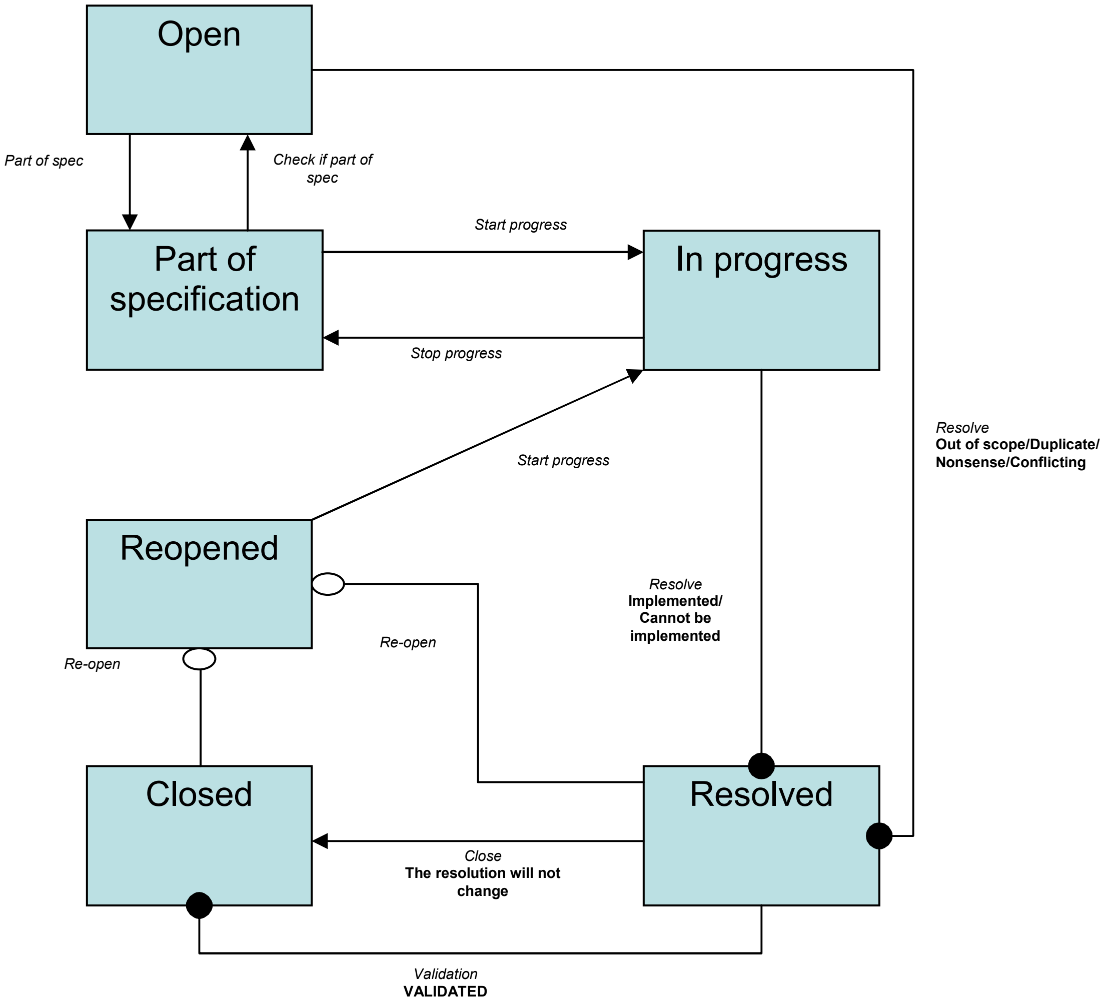


Figure 5 – The revised JIRA workflow

All requirements in the Volere schema fall in one of three categories:

- Functional Requirements
  User requirements that explicitly refer to the functionality of REACTION are referred to as functional user requirements. Functional requirements further classify into general, system, component and human-computer interface (HCI) requirements.
- Non-functional Requirement*s*
  Non-functional requirements address the quality of the future system and are classified to various criteria according the Volere schema (usability, performance, operational requirements, maintainability etc.).
- Constraints

  Restrictions imposed on the platform due to the budget, the time or the way the platform is designed or will work or interact with third-party components. Constraints are the same as other requirements except that they are mandated usually at the beginning of the project.

All requirements are documented and analysed in *D2-5 Initial Requirements Report*. Updated requirements reports will be issued as the iterative process continues.

5.3 Architectural specification

The purpose of the architectural specification is to develop and disseminate a platform architecture which is understood and agreed by all partners and, as such, can serve as the framework for developing all software components in REACTION.

The process is based on the standard IEEE 1471 "Recommended Practice for Architectural Description of Software-Intensive Systems" which defines core elements like viewpoint and view (IEEE, 2000). It also describes that stakeholders need to be involved and how to apply stakeholders' needs to the architecture.

Requirements and architecture influence one another. Requirements provide input for the architectural design process in that they frame the architectural problem and explicitly represent the stakeholders' needs and desires. On the other hand during the architecture design one has to take into consideration what is possible and look at the requirements from a risk/cost perspective.

The architectural design process is based on the following definition:

"Architecture Definition is a process by which stakeholder needs and concerns are captured, an architecture to meet these needs is designed, and the architecture is clearly and unambiguously described via an architectural description." (Rozanski, 2005)

The architectural design process will be supported by the introduction of "architectural perspectives" The architectural perspectives ensure that quality properties are not forgotten in the process because the viewpoint and view approach per se does not explicitly consider those quality properties. But those properties are critical to the success of the project and to reflect them properly one usually needs cross-view considerations while the viewpoints are relatively independent.

The perspectives and the combination possibilities with views that Rozanski and Woods propose are shown in the following diagram:

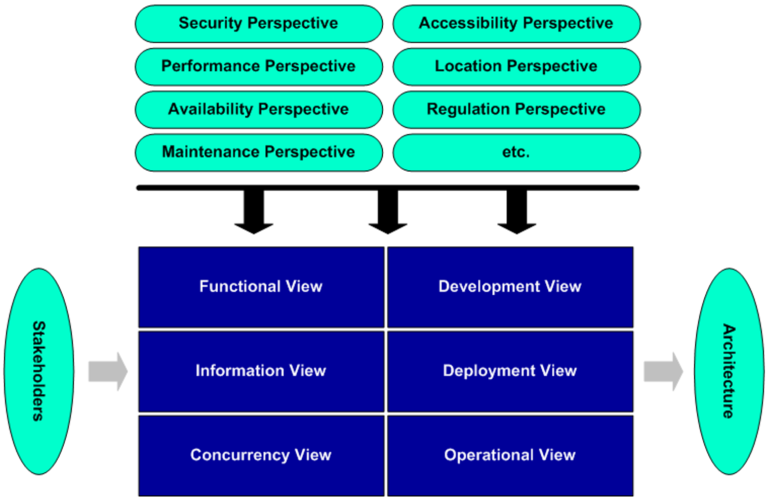


Figure 6 – Architectural perspectives and views

They propose the perspectives security, performance, availability, maintenance, location, regulation etc. but not all combinations of perspectives and views are needed. Artefacts created according to those perspective/view combinations need to be carefully chosen to prevent "analysis paralysis" and getting lost in details. But these perspectives will be a good opportunity to check if all considerations have been taken into account in the process. In later iterations the most appropriate combinations will be chosen and the guidelines and checklists for the different perspectives will be applied.

The work leads to a architectural design specification for three perspectives of the REACTION platform: performance, availability and security. The architectural considerations have been documented in deliverables *D4-1 State of the Art – Concepts & technology for a unified data fusion architecture, ID5-1-1 Initial network architectural analysis, ID6-5-1 Risk assessment engine architectural design* and *D7-2 Concepts of trust and architectural implications.* An additional document called *Internal Document for the REACTION Platform Specification* details the software architecture of the Platform and its individual components.

The architecture will be assessed and validated at each iteration cycle and updated if necessary.

5.4 System integration and verification

Towards the end of a complete development cycle, the components of the REACTION platform will undergo integration and technical verification of their functionality. System integration and verification will take place in each of the four iterative validation cycles in the research and development phase.

System integration encompasses integration of all the parts of the REACTION platform in which the following main objectives will be achieved:

1. To integrate the developed sensor networks and the data management and service orchestration subsystems with back end healthcare information systems.
2. To set up the prototypes of the REACTION platform in the various phases of the evolutionary design with the integration of the various services, seamless connectivity to heterogeneous networks, context awareness, risk assessment, and crisis management.
3. To develop prototype applications based on the clinical protocols, populate the ontologies and perform testing of the platform prior to field trials.
4. To carry out the deployment (installation, configuration and training) of the integrated prototype platform for the field trials.

Software verification (debugging and testing) is a quality control (QC) process that is used to evaluate whether or not a system component complies with regulations, specifications, or conditions imposed at the start of a development phase. It is always performed at the laboratory level by the technical partner(s) responsible for the component.

Verification is the answer to the following question: Have we built the system right? (i.e., does it match the requirement specification?). Thus, verification is the process of evaluating a subsystem or system in order to check whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

In the REACTION project, the internal verification procedures will be done at the technical partners' premises and will have the main purpose, without involving real end users and stakeholders, of performing the necessary tests in order to check whether the products of a given development phase satisfy the conditions imposed at the start of that phase or that the starting specifications have been correctly implemented.

Detailed procedures for the software verification process including test environment, structure of tests and system test is provided in deliverable *D2-7 Validation framework*.

5.5 Development of user applications

System integration also encompasses the setting up of prototypes to be used in the clinical field trials, including population of user domain data. This work is also a preparation for the validation of performance and applicability to user requirements.

A suitable development environment for rapid application development will be implemented and the prototypes will be used to build prototype scenarios in the different domains. The parts of the REACTION platform will be installed in a "smart" space test bed located at the premises of FORTH-ICS before being rolled out for field trials. In each cycle the previous prototypes will be enhanced.

5.6 Validation of prototypes

The purpose of user validation is to assure that the implemented result is in agreement with the needs and requirements of users. The user validation activities will focus on impact on patients, their relatives, healthcare personnel and other individual users as well as on organisational processes (e.g. in primary and secondary care as well as nursing care), with appropriate weight given to either aspect according to the phase of project progress. Hence, the validation will mostly centre on organisational workflows and stakeholder interaction as observed during the field trials. Traditional clinical research and validation of the clinical protocols is outside the scope of the project.

User validation is the answer to the question: Have we built the right system? (i.e., is this what the end users need and want?). Thus, validation is the process of evaluating a subsystem or system at the end of the development process in order to establish whether it satisfies specified user needs.

The process followed is similar in all validation cycles and foresees fixed steps to follow: an initial preparation part, an internal verification activity and/or a validation activity with (expert) end users, the collection and analysis of the outcomes and feedback of the results into the loop for the next step in the process.

The results of the validation are documented in a user validation report, which contains a description of the experience with the use of the platform at the clinical site, report the results of the usability test, the clinical workflow validation and the performance evaluation. Specific problems, inconsistencies or bugs at any level will be reported in order to be properly addressed in the next iteration and also new functionalities addressing specific user needs not yet included in the current requirement specifications will be listed.

Detailed procedures for the validation process including analysis of user needs, requirements, and preferences are provided in deliverable *D2-7 Validation framework*.

5.7 Field trial usability testing activities

The purpose of the Field Trial Usability Testing is to perform usability tests of prototypes used in the field trials. *Usability testing* is thus the assessment of the quality of use of the REACTION applications.

The overall aim of the field trials is to assess the effectiveness of the REACTION platform (i) within a hospital environment, (ii) with patients under therapeutic control in primary care and (iii) for patients who are self-managing their disease. The goal is to conclusively prove the validity of the applications, demonstrate the benefit for healthcare providers and provisioning authorities, gain acceptance by patients and other users and to assess the impact at the organisational level.

In order to evaluate the overall user experience in using the REACTION platform, usability tests will be conducted once or twice in each envisaged environment, in order to collect feedback from the users of the platform, and give retrofit to the developers for redesigning, if necessary, user interfaces or functionality of the REACTION platform.

Usability will be tested in two field trials (in-hospital and primary care) with a small number of users to detect user problems and deficiencies of the prototypes.

Validations will be performed for the applications that have been built using the platform, and to a lesser extent for the platform used by developers to create the applications. It is possible that an application validates poorly because the application developer made an inferior job of using the platform. The opposite may also happen that the application validates positively, but in reality the platform was inadequate or its capabilities not exploited. Whatever the outcome, the results will be fed back to the development teams.

The end user usability testing report will contain a description of the experience with the use of the platform, including the performance evaluation. Specific problems, inconsistencies or bugs at any level will be reported in order to be properly addressed in the next release and also new functionalities addressing specific user needs not yet included in the current requirement specifications will be listed.

Detailed procedures for the usability testing process including the establishment of quality criteria and testing procedures are provided in deliverable *D2-7 Validation framework*.

6. Re-engineering of requirements

After the successful completion of a prototype cycle, each RTD work package will analyse and report their development results, RTD experiences, Lessons Learned in the development and integration work and other relevant knowledge gained during the development cycle. Moreover, knowledge gained from formal testing and system integration will be collected together with latest developments in technology, regulatory affairs and markets, which influence the REACTION platform and its exploitability.

6.1 Lessons Learned

Lessons Learned are a principal component of a project culture committed to Knowledge Management. Lessons Learned help to support project goals in the RTD work of:

- Promoting recurrence of successful outcomes
- Precluding the recurrence of unsuccessful outcomes

As part of the continuous improvement program adopted by the REACTION Project Board a systematic and continuous collection, indexing and dissemination of Lessons Learned will be undertaken in WP2.

This section will establish criteria for the Lessons Learned process in REACTION and discuss how to turn Lessons Learned into Lessons Applied.

Lessons are learned during project RTD work, during testing and integration, as a part of the validation of project prototypes and during literature search and technology watch reports. Lessons can thus be learned throughout the project work. As such, Lessons Learned constitute both individual and organisational knowledge and understanding gained by experience, either negative (missed targets, solutions that do not work as expected, wrong choice of technology) as well as positive (easier implementation than expected, faster response time, more interoperable devices than expected).

In order to implement a workable Lessons Learned process, we need first to define what we understand with the term “lesson”. We use the following characterisation for a lesson:

– It must be significant in terms of the project progress and ability to meet its goal.

– It must be valid, i.e. the experience gained must be repeatable.

– It must be applicable to the REACTION project

– It may contain or address pertinent info

– It may provide information of interest

Not all experiences will qualify as being Lessons Learned and it is important that reported Lessons Learned not merely restate existing information and existing experiences not related to REACTION work.

The REACTION Lesson Learned process has 6 steps:

- Collection
- Verification
- Storage
- Dissemination
- Reuse
- Identification of improvement opportunity

Collection

The collection process focuses on collecting Lessons Learned from many sources internal and external to the project. The collection will be undertaken in practically all work packages.

WP2 will collect Lessons Learned from the iterative requirements engineering process which can be reused to improve the performance and efficiency of future iterations.

The RTD work undertaken in WP3-7 will provide a large amount of Lessons Learned, by virtue of the many researchers participating in this work and the many small and large experiences gained individually and as teams. The challenge here is to identify and properly describe the Lessons Learned and filter them according to significance, validity, and applicability to the REACTION project.

The field trials in WP8 will obviously provide a range of experiences that can be classified as Lessons Learned. So will the implementation and integration work done in WP10 as well as the endeavours of WP9 of identifying sustainable business ecosystems for deployment. Finally, the supporting work undertaken in the form of demonstrations, training and dissemination will, at least in principle, contribute Lessons Learned to the project.

Verification

Verifying the collected Lessons according to established standards is the second step in the process. All Lessons Learned must be verified for correctness, significance, validity, and applicability. The verification will be performed by the WP2 team together with the Technical Manager and relevant WP leaders. The Technical Manager will decide to add and remove Lessons Learned as necessary.

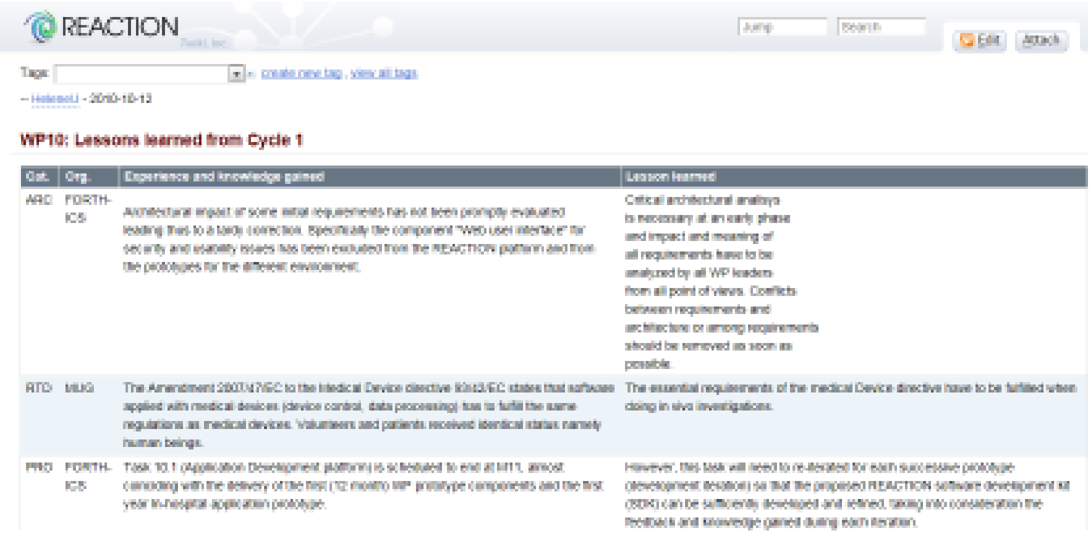
Some of the criteria that may be used for verification are:

- Relationship with the project flow
- Relevance to the project outcome
- Significance in terms of quality parameters such as robustness, ease of use, functionality
- Research aids used
- Systemic process issues
- Credibility or reputation of the originator

Storage

In the first instance, the Lessons Learned will be entered into a reserved area of the REACTION TWiki. The area has been created and is maintained by WP2. It contains a simple categorisation tagging for filtering purposes. For the sake of simplicity, a very simple template will be provided with no special structure or format needed.

The Lessons Learned repository will act as an organisational memory for experiences incurred by all project members during the cause of the project.



-- HelenaU - 2010-10-13

WP10: Lessons learned from Cycle 1

Cat.	Org.	Experience and knowledge gained	Lesson learned
ARC	FORTH-ICS	Architectural impact of some initial requirements has not been promptly evaluated leading thus to a tardy correction. Specifically the component "Web user interface" for security and usability issues has been excluded from the REACTION platform and from the prototypes for the different environment.	Critical architectural analysys is necessary at an early phase and impact and meaning of all requirements have to be analyzed by all WP leaders from all point of views. Conflicts between requirements and architecture or among requirements should be removed as soon as possible.
RTD	MUG	The Amendment 2007/47/EC to the Medical Device directive 93/42/EC states that software applied with medical devices (device control, data processing) has to fulfil the same regulations as medical devices. Volunteers and patients received identical status namely human beings.	The essential requirements of the medical Device directive have to be fulfilled when doing in vivo investigations.
PRO	FORTH-ICS	Task 10.1 (Application Development platform) is scheduled to end at M11, almost coinciding with the delivery of the first (12 month) WP prototype components and the first year in-hospital application prototype.	However, this task will need to re-iterated for each successive prototype (development iteration) so that the proposed REACTION software development kit (SDK) can be sufficiently developed and refined, taking into consideration the feedback and knowledge gained during each iteration.

Figure 7 – TWiki for collection and storage of Lessons Learned

Dissemination

A very important part of the process is of course to inform other users in the feedback cycle. All project workers are encouraged to continuously consult the Lessons Learned repository, not only with the purpose of reporting, but also to continuously follow Lessons Learned by other project partners.

Once in every iteration cycle, the Lessons Learned will be documented in internal deliverables *ID2-8-2 Change request and re-engineering report 1* (M15), *ID2-8-3 Change request and re-engineering report* 2 (M25) and *ID2-8-4 Change request and re-engineering report 3* (M37).

Reuse

The REACTION project encourages and promotes lessons to be used by other than the submitter. The WP leaders have a responsibility to consult the Lessons Learned repository regularly and at least before any major decision affecting the scientific work and project outcome is to be made. The WP leaders are obliged to take part in the engineering process of requirements, which is based on a timely assessment of the reported Lessons Learned.

6.2 Requirements improvement opportunities and change request reports

The last step in the process relates to the identifying of incremental and innovative improvements that will measurably improve the project's requirement specification. Input from the Lessons Learned from the validation reports will be analysed and used for updating of requirements.

Further, an environmental screening process will be undertaken to identify trends which may influence the requirements during the course of the project. These trends are documented in the deliverables *ID2-2 Clinical watch report, ID2-3 Technology watch report and D2-4 Market and regulatory standards watch report.*

From the relevant analysis new and/or updated requirements will be extracted. The identification, formulation and validation of requirements will be performed by a team together with the Technical Manager and relevant WP leaders. The JIRA tool will be used for gathering and sharing requirements changes.

The new and updated requirements will be scrutinised and prioritised by the Technical Manager and relevant WP leaders. A handful of the requirement changes with the highest priority for each WP will be selected. New user requirements that conflict with existing requirements will be resolved by the team. This process will be part of the quality gateway process described above. Conflict resolution will be solved by evaluating the relevant requirements against the overall project vision, the exploitability of the final project results, and the cost of implementation. Important conflicts with substantial impact on the architecture description shall be referred to the Technical Board for discussion and resolution.

The Technical Manager and the WP2 leader will evaluate and describe the impact on the future development work arising from the re-engineered requirements and report this in a Change Request Report, which also will contain the relevant Validation Reports. The evaluation will take place both in the context of scientific and technological progress, adherence to the overall vision of the project and impact on the overall project plan.

Once in every iteration cycle, the re-engineered requirements will be documented in deliverables *ID2-8-2 Change request and re-engineering report 1*, *ID2-8-3 Change request and re-engineering report 2* and *ID2-8-4 Change request and re-engineering report 3*.

7. Conclusion

WP2 will manage and undertake the work of carrying out the iterative engineering of requirements, with special focus on the engineering process after the end of each iteration cycle and continuous monitoring of key technological opportunities.

The Lessons Learned from the development process as well as the results of prototypes evaluated with users during the project progress will be used to re-formulate the initial requirements and enable the necessary modification of the design specification and possible re-engineering of the affected modules.

The objectives of this work package are as follows:

- Elicit the generic and specific domain requirements for the full technical and business realisation of the project results to support the developer users as well as to ensure the enhancement of the business potential for all stakeholders (completed in period 1).
- Formulate the initial user requirements in terms of the functionalities expected from each of the subsystems (completed in period 1).
- Maintain a continuous study of the clinical, technological, regulatory standards and market developments affecting the REACTION platform and their impact on project requirements
- Collect comments, analysis and verification results from development work, software testing, Lessons Learned in prototyping, results from prototype validation.
- Structure, study and analyse the feedback into design re-specifications to enable the requirements re-engineering to enhance the resulting project deliverables for best technical, usability fit and market potential.
- Add to and update the JIRA requirements database with new and modified requirements and prioritise a limited number of requirements for each WP, which should be in focus in the subsequent development iteration phase.

The basis for engineering of requirements is a structured dialogue with the developers and representatives of the user domains.

8. References

(IEEE, 2000) IEEE Standard 1471-2000: Recommended Practice for Architectural Descriptions of Software-Intensive Systems, IEEE Computer Society

(Robertson, 2006) Robertson S. & Robertson J. (2006) Mastering the Requirements Process. Upper Saddle River, NJ, Addison-Wesley

(Rozanski, 2005) Rozanski N. & Woods E. (2005) Software Systems Architecture: working with Stakeholders Using Viewpoints and Perspectives. Upper Saddle River, NJ, Addison-Wesley