DICOMA: DIsaster **CO**ntrol **MA**nagement

Deliverable D1.1

Specification of the User Centred Design (UCD) process

Project Identifier	DICOMA
Project Title	**DIsaster COntrol MAnagement**
Document Version	0.1
Planned Delivery Date	31.10.2012
Actual Delivery Date	<Month Day, Year>
Document Title	Specification of the User Centred Design (UCD) process
Work Package	WP1
Document Type	Word
Abstract	The results of the T1.1 are reported in this deliverable D1.1 – Specification of the UCD process. The issues presented in the deliverable are related to definition, implementation and instrumentation of UCD design.
Keywords	UCD definition, UCD implementation, UCD process.

Function	***Name***	***Entity***
Editor	Mirjami Jutila	VTT
Author	Ville Antila	VTT

List of DiCoMa Task 1.1 participants:

Partner	Country	Type
Mattersoft	Finland	SME
Mobisoft	Finland	SME
VTT	Finland	RES

Contents

1 Executive Summary 6
2 Introduction 7
2.1 Background 7
3 User-centred design process 9
3.1 Personas and scenarios 9
3.2 Use cases 9
3.3 User requirements 10
3.3.1 RTH - Requirements and Testing Hub 10
3.3.2 RTH User guide 13
3.3.2.1 Access to the requirement tool: RTH 13
3.3.2.2 Structure of RTH 13
3.3.2.3 Create requirements 14
3.3.2.4 Create user stories 17
3.3.2.5 Associate user stories/scenarios to requirements 21
3.3.2.6 Associate scenarios to user stories 23
3.3.2.7 Users and password 24
3.4 Existing challenges in the domain 25
3.5 Interaction methods and devices 25
4 Produce design solutions 27
4.1 Forming COP in disaster management 27
4.2 Prototyping design solutions 28
4.2.1 Paper prototypes and low-fidelity mock-ups 28
4.2.2 Functional UI prototypes 28
5 Evaluation of use 29
5.1 Usability metrics 29
5.2 Evaluation methods 30
5.2.1 Usability evaluation 30
5.2.2 User trials with the complete system 32
5.2.3 Summary of evaluation methods 32
6 Conclusions 34
7 References 35

1 Executive Summary

During the crisis management, several authorities and organizations (police, fire brigade, emergency response unit etc.) organize their work using separate policies and procedures. Decision making in crisis situation is a highly complicated procedure, and there is a genuine need to improve the cooperation between different units involved in emergency response activities. Good support in decision making is of critical importance to react accurately, fast and effectively.

Providing the technical solutions and tools for decision makers include overall User Centred Design –UCD process in order to identify the means for providing better services and devices to users.

2 Introduction

Work Package 1 'User Centred Design' defines the overall user centred design (UCD) process for the DiCoMa project in order to ensure the usability and economic perspective of the future solution. Thus WP1 is related to and affects all the other work packages of DiCoMa. The objective of WP1 is to keep the work focused on the users and business stakeholders, their needs and the environment. Task T1.1 – will form the "Definition, implementation and instrumentation of user centred design methodology".

DiCoMa will use UCD from the beginning until the end of the project, using the close collaboration with end users to establish requirements, to examine new functional concepts in data analysis and presentation and to validate the newly developed user interfaces.

In task 1.1 as described in the DiCoMa FPP, state of the art in usability engineering will be taken into account in order to identify proper methods to be applied in the UCD design process.

The aim of this task is to support the definition and management of UCD activities, which share the following characteristics:

- The active involvement of users and a clear understanding of user and task requirements ('context of use')
- An appropriate allocation of function between users and technology ('user requirements')
- The iteration of design solutions ('produce design solutions')
- Multi-disciplinary design ('evaluation of use')

The main tools for UCD Analysis (and Dicoma WP1) phase are:

- **Personas** - fictional characters with all the characteristics of the end-user.
- **Scenarios** - fictional stories about the normal day as well as challenges and goals that user has with personas as the main characters.
- **Use cases** – more in detail description how end-users are system to be developed interact with each other.

2.1 Background

Usability can be seen as a study how to ease the interaction between people and devices. A general definition, as provided by the standard ISO 9241-11, defines the usability and its goals as follows:

"Usability is the extent to which a product can be used by specified users to achieve

specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. This means that usability is not a property of the product itself - it is a property of the entire system, including the product, the user, the user's goals, and the context of use."

As a background example a study of information processing flow and some general user requirements in the emergency sector was carried in the Netherlands in 2006 [1]. They identified *user requirements* for a system providing services in emergency response:

"The first very important principle considered is time. Respondents from emergency services stated that their service requirements are ***time critical*** and in emergency response they demand almost instant and reliable responses from mobilising systems. On the other hand most procedures in risk prevention are not time critical and data response can be acquired over many hours or even days."

Related to this time aspect, respondents involved in crisis response argue that much of the information they request during a crisis can be seen as ***dynamic information***. Mentioned examples are: what's the current magnitude of a toxic cloud and how will this cloud develop over time? What is the current capacity of the nearest hospitals? Which roads are accessible and which not?

Because the circumstances during an emergency may change every moment, continuous monitoring of the developments and a continuous distribution monitored changes is necessary.

As emergency management is a ***multi-disciplinary activity***, it should be possible to exchange information between different partners at different administrative levels. To realise this, a decent spatial data infrastructure is required. Because time forms a critical factor in emergency management, the spatial data infrastructure should be suitable for quick data input and transfer.

3 User-centred design process

3.1 Personas and scenarios

Persona is a fictional character of created to characterize the target user group, representing their hypothesized goals and behaviour. They are short descriptions (1-2 pages) that include behaviour patterns, goals, skills and environment. For each product, more than one persona can be created. Benefits of personas are that it gives the target user group concrete features and thus helps infer real user needs. They can assist in brainstorming, use case specifications and in product feature identification. They also help communication between different stakeholders in the project by concretizing the target user group. Personas are also a common starting point in creating the usage scenarios of the system.

Scenario is a narrative of the interactions between the user and the system (including both hardware and software). Scenario has a functional goal, but can span from single task or transaction to a full day in the operational life of the system. The scenarios are written in plain language without and they should not include technical details of the system. They should be understandable by all the stakeholders of the system. Use Cases are often defined from Scenarios, documenting the way of performing a task with the system.

- Inputs: interviews and surveys with target user group, domain knowledge from domain expert
- Outputs: personas, scenarios/user stories

3.2 Use cases

Use cases are a powerful tool for a designer in early stages of a product design process. They can be used to answer the question: “how is this product actually going to be used?’ It does not answer the question of how exactly is the product going to be built or what specific mechanisms are going to make it work, it only answers the question of how it will be used.

A use case is a description of a task user might want to complete using the service and each use case focuses on describing how the user achieves the goal or task. Different use cases can be created for different types of people that have been identified as the product’s target group. Use case validity can be checked using the identified user group by directly asking them or observing their daily tasks.

Use cases are particularly useful to understand user interaction needs with the product interface. As use cases are technology independent, they are especially useful for finding out user experience requirements when the same service is provided through different interface technologies [Cockburn, A. Use cases, ten years later. *Software Testing and Quality Engineering Magazine*, pp. 37-40, 2002].

How the use cases are generated in DiComa is defined in D1.2 “Context of Use, Scenarios and Use Cases”.

- Inputs: scenarios
- Outputs: use case model (diagram and detailed descriptions)

3.3 User requirements

All the efforts have been focused on the research of the most suitable requirement tool and the definition of the user guide of the selected requirement tool (RTH - Requirements and Testing Hub).

First, in order to find the suitable requirement tool, a detailed research was performed to select the requirement tool that fulfills the needs of the DICOMA project. The results of this performed research by the project members are shown in the deliverable D1.1. Specification of the User Centred Design (UCD) process.

After completing this research, RTH has been chosen as the requirement tool to manage the requirements, user stories and scenarios collection in the project.

On one hand, we detailed RTH User guide. In this section we have explained the access to the requirement tool, the structure of RTH, the creation of requirement and user stories and the relation that can be established between them. The RTH structure is composed of three different element types (scenarios, user stories and requirements) and these can be associated once they have been created. We have included the scenarios of the DICOMA project and its nomenclature. These scenarios are: Forest Fire (FF), Aircraft crash (AC), Heavy Winter Storm (HS), Chemical Good Crash (CC) and Earthquake (EQ). Moreover, the nomenclature of user stories and requirements have been also included in this deliverable.

3.3.1 RTH - Requirements and Testing Hub

RTH, Requirements and Testing Hub is an open source tool used to the manage requirements of any system. This tool was selected to be used in DICOMA after an exhausted analysis of different open source requirement tools. In this analysis, the project members analyzed different tools to find the most suitable for managing requirements.

Moreover, RTH provides great functionalities like the association of requirements to other requirements, generation of traceability matrix, multiuser application, among others.

The results of the research performed by the project members of the different requirement tools are shown in the following table:

Tool features	OSRMT	REM	REMAS	RTH	OpenAdams	OpenCMS
Permits requirements/ subrequirements	Yes	Yes	Yes	Yes	Yes	Yes
Project artifacts:						
* Actor packages	Yes	Yes	Yes	No	No	No
* Use case packages	Yes	Yes	Yes	Yes	Yes	No
* Testing case	Yes	No	No	Yes	Yes	No
* Functionalities	Yes	No	Yes	Yes	Yes	No
* Designs	Yes	No	No	No	Yes	No
* Requirements	Yes	Yes	Yes	Yes	Yes	No
> *Functional*	Do not specify	Yes	Yes	Do not specify	Do not specify	No
> *Informaton*		Yes	Yes			No
> *Not functional*		Yes	Yes			No
> *of Restriction*		Yes	Yes			No
* Organization	No	Yes	Yes	No	No	No
* Participants	No	Yes	Yes	Yes	No	No
* Meeting	No	Yes	Yes	No	No	No
* Aims	No	Yes	Yes	No	No	No
* Traceability matrix	No	Yes	No	Yes	No	No
* Association of artifacts	No	No	Yes	No	No	No
* Images	No	No	Yes	No	Yes	Yes
* Components	No	No	No	Yes	Yes	No
* Global template of system	No	No	Yes		No	No
* Addition of artifacts	Yes	No	No	Yes	Yes	No
* News	No	No	No	Yes	No	No
> *Subject*	No	No		Yes	No	
> *Description*	No	No		Yes	No	
Analysis documents	No	No	No	Yes	No	No
Traceability matrix	Yes: Schema, table, Tree	Yes: Normal	Yes: table	Yes: Screen	No	No
Metrics of function points	No	No	Yes	No	No	No
User management	Yes	No	Yes	Yes	No	No
Exports	Yes: PDF, XTML	No	No	Yes: Excel	No	No
Template package	No	No	Yes	No	No	Yes

Search filters:	Yes: Normal	No	Yes: Tree, Normal	Yes	Yes: Tree	No
* Priority	Yes		No	Yes	Yes	
* Category	Yes		Yes	No	Yes	
* State	Yes		Yes	No	Yes	
* Version	Yes		No	All o last	No	
* ID	No		Yes	No	Yes	
* Complexity	No		No	No	Yes	
* Effort	No		No	No	Yes	
* Risk	No		No	No	Yes	
* Assignment	No		No	Yes	No	
* Document type	No		No	Yes	No	
* Functionality	No		No	Yes	No	
* For words	No		Si	Yes	No	
* Display X searchs	No		No	Yes	No	
Imports	No	No	No	Yes	No	No
Testing	Yes	No	No	Yes	No	No
Relation of requirements	Yes	Yes	Yes	Yes	No	No
Report editor	Yes	No	No	Yes	No	Yes
Installation or web	Installation	Installation	Installation	Web (SQL)	Installation	Apache
File submission	No	No	No	Yes	No	Yes
Requirement features:	Can be personalized	Predefined	Predefined	Predefined/ Personalized	Predefined	Do not specify
* Version control	Predefined	Yes	No	Yes	No	
* Complexity	Predefined	No	No	Yes	Predefined	
* Category	Predefined	No	Yes	Yes	Predefined	
* Priority	Predefined	Predefined	No	Yes	Predefined	
* Assignment	Predefined	No	No	Yes	Yes	
* Description	Yes	Yes	Yes	Yes	Yes	
* Details	Yes	Yes	No	Yes	No	
* Author	Yes	Yes	Yes	Yes	No	
* Origin	Yes	Yes	No	Yes	Yes	
* Component	No	No	No	No	Yes	
* Testing case	No	No	No	Yes	Yes	
* Use cases	Yes	No	Yes	No	Yes	
* Dependences	Yes	Yes	No	No	No	
* History	Yes	No	No	Yes	No	

* Life time	No	Yes	No	Yes	No	
* Importance	No	Predefined	No	Yes	No	
* Concurrence applications	No	Yes	No	No	No	
* State	Predefined	Predefined	No	Yes	Predefined	
* Stability	No	Predefined	No	No	No	
* Comments	No	Yes	Yes	No	No	
* ID	Automatically generated	Automatically generated	Yes	Automatically generated	Automatically generated	
* Base line	No	No	No	Yes	Yes	
* Effort	No	No	No	No	Predefined	
* Attached documents	Yes	No	No	Yes	No	
* Last update	No	No	Si	Yes	No	
* Change justification	No	No	No	Yes	No	
* Subrequirement	No	No	No	Yes	No	
* Block	No	No	No	Yes	No	
* Add features	Yes	No	No	No	No	

3.3.2 RTH User guide

3.3.2.1 Access to the requirement tool: RTH

URL: https://innerisl.dyndns-server.com/rth/login.php

3.3.2.2 Structure of RTH

The RTH structure is composed of three different elements types, there are: scenarios, user stories and requirements. The only difference between these elements is the value of the "DOC_TYPE" field. This field can have the following values: REQ, USER STORY or SCENARIO.

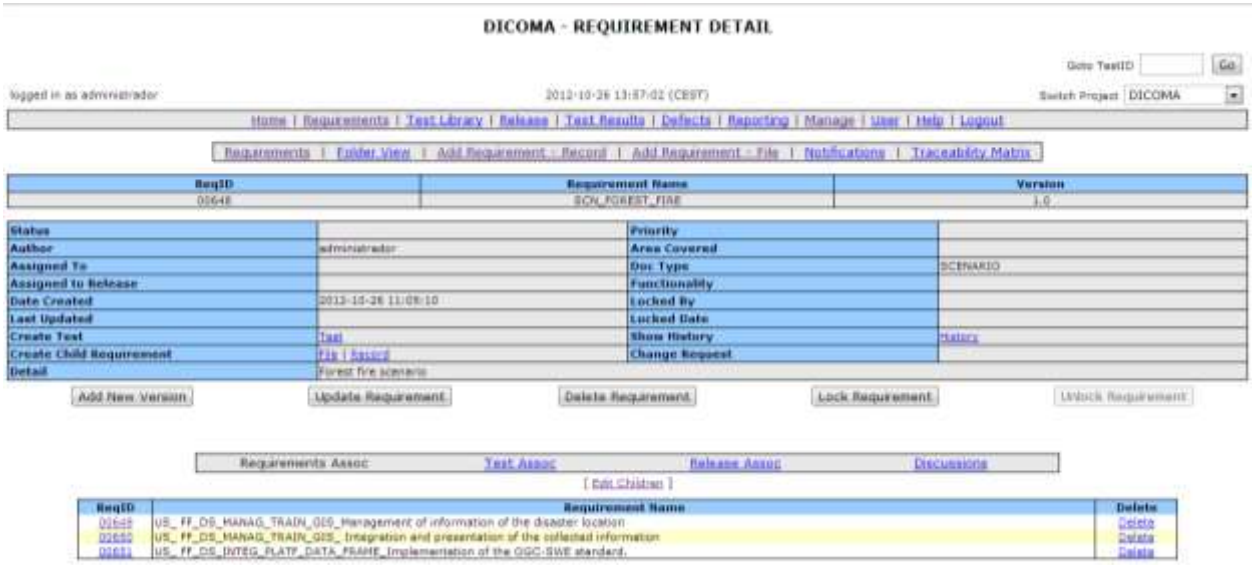
In order to introduce requirements or user stories in the RTH tool, first the different scenarios must be defined. Each requirement must belong to a specific scenario. To DICOMA project, the following scenarios have been defined with its corresponding nomenclature:

- Forest Fire → **FF**
- Aircraft crash → **AC**
- Heavy Winter Storm → **HS**
- Chemical Good Crash → **CC**
- Earthquake → **EQ**
- General Scenario → **GE**

Currently, there are two different ways to relate scenarios, user stories and requirements:

- Scenario → User story → Requirement: Using this association a scenario has user stories as children and these last ones have requirements as children.
- Scenario → Requirement: Using this association a scenario has requirements as children.

Next, a figure is presented in order to show a tab of the Forest fire scenario with several associated user stories.

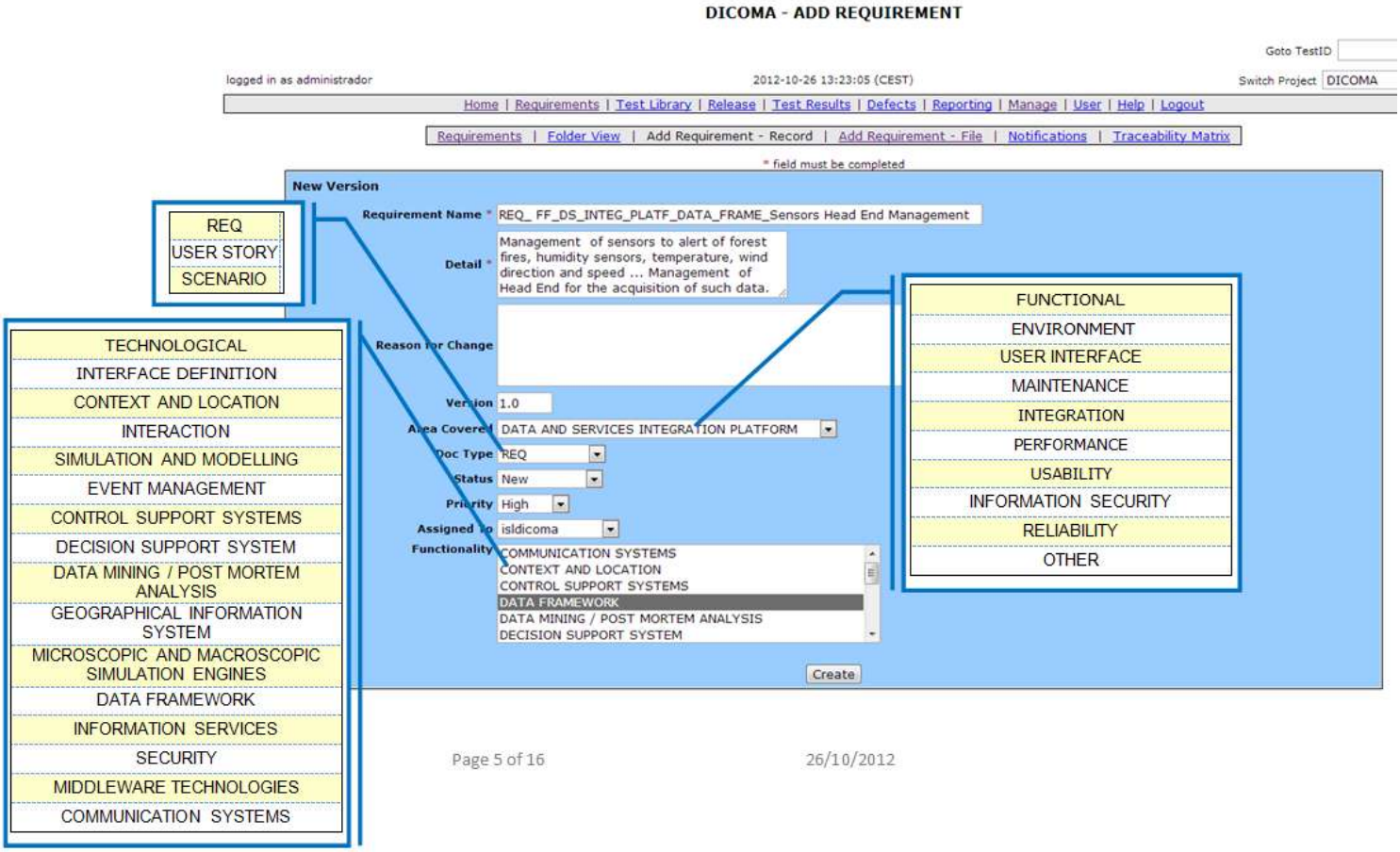


3.3.2.3 Create requirements

1. **Go to tab of "Requirements".**

2. Go to tab of "Add Requirement - Record" and DOC_TYPE = REQ.

3. To insert your requirement, you should complete the attributes as it is explained below :



3.3.2.3.1 Requirement name nomenclature

REQ_Scenario_Category_Functionality _Three main words of the requirement

Where:

1. Scenario: It should be one of the listed below:
   - Forest Fire → **FF**
   - Aircraft crash → **AC**
   - Heavy Winter Storm → **HS**
   - Chemical Good Crash → **CC**
   - Earthquake → **EQ**
   - General Scenario → **GE**
2. Category (Areas covered indicate the category of the requirement) : it should be one of the listed below:
   - Functional → **F**
   - Environment → **E**
   - User interface → **US**
   - Maintenance → **M**
   - Integration → **I**
   - Performance → **P**
   - Usability → **U**
   - Information security → **IS**
   - Reliability → **R**
   - Other →**O**
3. Functionalities: it should be one of the listed below:
   - User Interface Techniques
     - UI_Techniques-Technological → **UI_TECH_TECH**
     - UI_Techniques-Interface definition → **UI_TECH_INTER_DEF**
     - UI_Techniques-Context and location → **UI_TECH_CON_LOC**
     - UI_Techniques-Interaction → **UI_TECH_INTER**
   - Disaster Support Management and Training
     - Ds_Manag_Train-Simulation and modeling → **DS_MANAG_TRAIN_SIM_MODEL**
     - Ds_Manag_Train-Event Management → **DS_MANAG_TRAIN_EVENT_MANAGE**

- Ds_Manag_Train-Control Support Systems → **DS_MANAG_TRAIN_CONTROL_SS**
- Ds_Manag_Train-Decision Support System → **DS_MANAG_TRAIN_DEC_SS**
- Ds_Manag_Train-Data Mining / Post Mortem Analysis →**DS_MANAG_TRAIN_DMPMA**
- Ds_Manag_Train-Geographical Information system → **DS_MANAG_TRAIN_GIS**
- Ds_Manag_Train-Microscopic and Macroscopic Simulation engines →**DS_MANAG_TRAIN_MM_SIM_ENG**

- Data and Services Integration Platform
  - Ds_Integ_Platf-Data Framework → **DS_INTEG_PLATF_DATA_FRAME**
  - Ds_Integ_Platf-Information Services → **DS_INTEG_PLATF_INF_SERV**
  - Ds_Integ_Platf-Security → **DS_INTEG_PLATF_SEC**
  - Ds_Integ_Platf-Middleware Technologies → **DS_INTEG_PLATF_MIDDLE_TEC**
- Communications and Validation
  - Comm_Valida-Communication Systems → **COMM_VALIDA_COM_SYS**

4. Three main words of the requirement: do a summary of the key words that describe the requirement.

3.3.2.3.2 Example

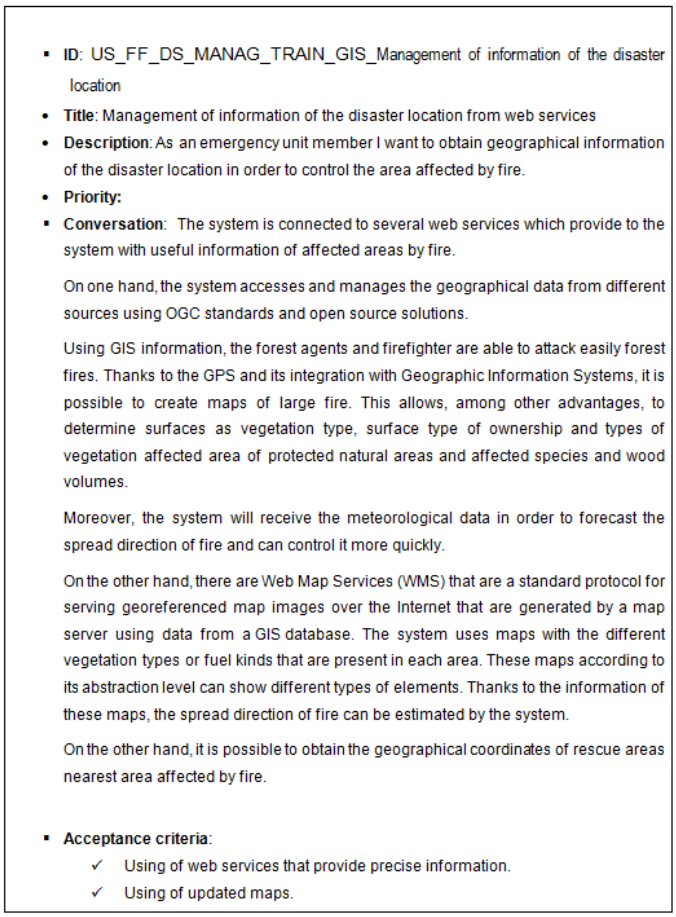
Requirement name: REQ_FF_DS_INTEG_PLATF_DATA_FRAME_Sensors Head End Management.

Detail: Management of sensors to alert of forest fires, humidity sensors, temperature, wind direction and speed ... Management of Head End for the acquisition of such data.

3.3.2.4 Create user stories

1. **Create a user story as a requirement, using the same steps except:**

   a) The doc_type in this case should be "USER STORY".
   b) Instead of an empty file, upload a user story document according to the official templates. Each user story mustn't take up more than one sheet. Example:



- **ID**: US_FF_DS_MANAG_TRAIN_GIS_Management of information of the disaster location
- **Title**: Management of information of the disaster location from web services
- **Description**: As an emergency unit member I want to obtain geographical information of the disaster location in order to control the area affected by fire.
- **Priority:**
- **Conversation**: The system is connected to several web services which provide to the system with useful information of affected areas by fire.

  On one hand, the system accesses and manages the geographical data from different sources using OGC standards and open source solutions.

  Using GIS information, the forest agents and firefighter are able to attack easily forest fires. Thanks to the GPS and its integration with Geographic Information Systems, it is possible to create maps of large fire. This allows, among other advantages, to determine surfaces as vegetation type, surface type of ownership and types of vegetation affected area of protected natural areas and affected species and wood volumes.

  Moreover, the system will receive the meteorological data in order to forecast the spread direction of fire and can control it more quickly.

  On the other hand, there are Web Map Services (WMS) that are a standard protocol for serving georeferenced map images over the Internet that are generated by a map server using data from a GIS database. The system uses maps with the different vegetation types or fuel kinds that are present in each area. These maps according to its abstraction level can show different types of elements. Thanks to the information of these maps, the spread direction of fire can be estimated by the system.

  On the other hand, it is possible to obtain the geographical coordinates of rescue areas nearest area affected by fire.

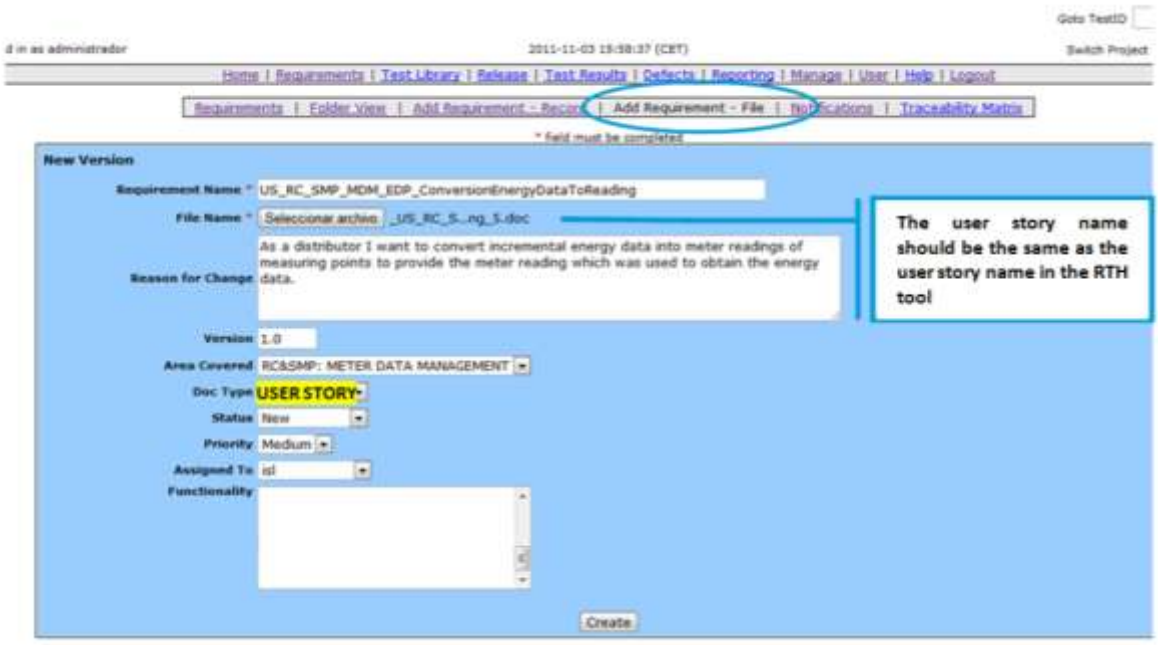
- **Acceptance criteria**:
  - ✓ Using of web services that provide precise information.
  - ✓ Using of updated maps.

This template contains the correct structure that must be used by all the consortium members. Also, it must achieve the following objectives:

- They are written with simple ideas from the user point of view.
- They are focused on WHAT and not on HOW.
- It should be possible to write tests to each user story.

The fields to represent user stories are:

- **Id**: a unique user story identifier.
- **Title**: a short text describing the functionality. It should describe actions.
- **Description**: a brief description of the functionality. It is recommended to write the user story in the format:
  "As a <role>, I want that <system functionality> to <represent business value>"
- **Priority**: the user story priority. It should be fixed by the customer
- **Conversation**: a text describing activities that should be done to support the user story.
- **Acceptance criteria**: those issues that the user story should meet.



3.3.2.4.1 User story name nomenclature

US_Scenario_Category_Functionality _Three main words of the user story

Where:

1. Scenario: It should be one of the listed below:
   - Forest Fire → **FF**
   - Aircraft crash → **AC**
   - Heavy Winter Storm → **HS**
   - Chemical Good Crash → **CC**
   - Earthquake → **EQ**
   - General Scenario → **GE**
5. Areas covered (Areas covered indicate the category of the requirement) : it should be one of the listed below:
   - Functional → **F**
   - Environment → **E**
   - User interface → **US**
   - Maintenance → **M**
   - Integration → **I**
   - Performance → **P**
   - Usability → **U**
   - Information security → **IS**

- Reliability → **R**
- Other →**O**

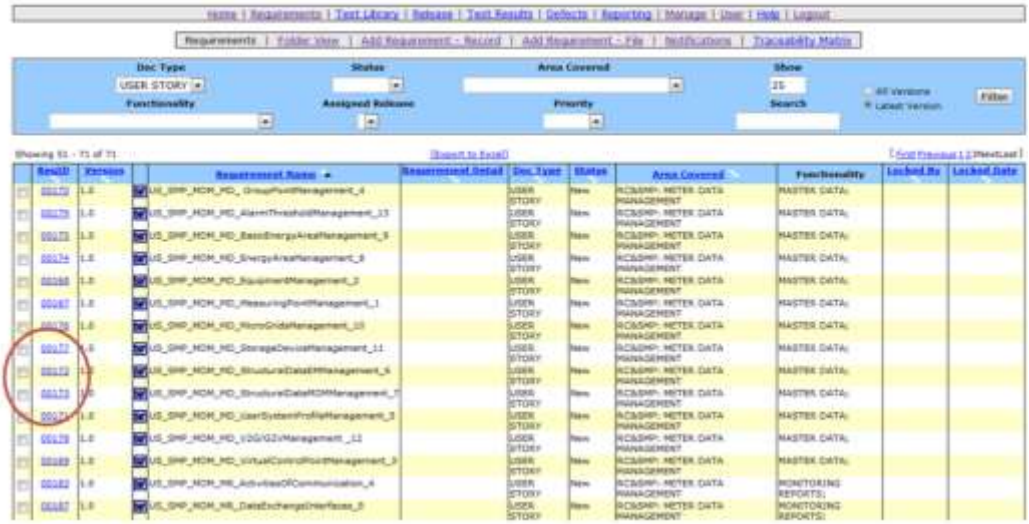
2. Functionalities: it should be one of the listed below:
   - User Interface Techniques
     - UI_Techniques-Technological → **UI_TECH_TECH**
     - UI_Techniques-Interface definition → **UI_TECH_INTER_DEF**
     - UI_Techniques-Context and location → **UI_TECH_CON_LOC**
     - UI_Techniques-Interaction → **UI_TECH_INTER**
   - Disaster Support Management and Training
     - Ds_Manag_Train-Simulation and modeling → **DS_MANAG_TRAIN_SIM_MODEL**
     - Ds_Manag_Train-Event Management → **DS_MANAG_TRAIN_EVENT_MANAGE**
     - Ds_Manag_Train-Control Support Systems → **DS_MANAG_TRAIN_CONTROL_SS**
     - Ds_Manag_Train-Decision Support System → **DS_MANAG_TRAIN_DEC_SS**
     - Ds_Manag_Train-Data Mining / Post Mortem Analysis →**DS_MANAG_TRAIN_DMPMA**
     - Ds_Manag_Train-Geographical Information system → **DS_MANAG_TRAIN_GIS**
     - Ds_Manag_Train-Microscopic and Macroscopic Simulation engines →**DS_MANAG_TRAIN_MM_SIM_ENG**
   - Data and Services Integration Platform
     - Ds_Integ_Platf-Data Framework → **DS_INTEG_PLATF_DATA_FRAME**
     - Ds_Integ_Platf-Information Services → **DS_INTEG_PLATF_INF_SERV**
     - Ds_Integ_Platf-Security → **DS_INTEG_PLATF_SEC**
     - Ds_Integ_Platf-Middleware Technologies → **DS_INTEG_PLATF_MIDDLE_TEC**
   - Communications and Validation

- Comm_Valida-Communication Systems → **COMM_VALIDA_COM_SYS**

3. Three main words of the user story: do a summary of the key words that describe the user story.

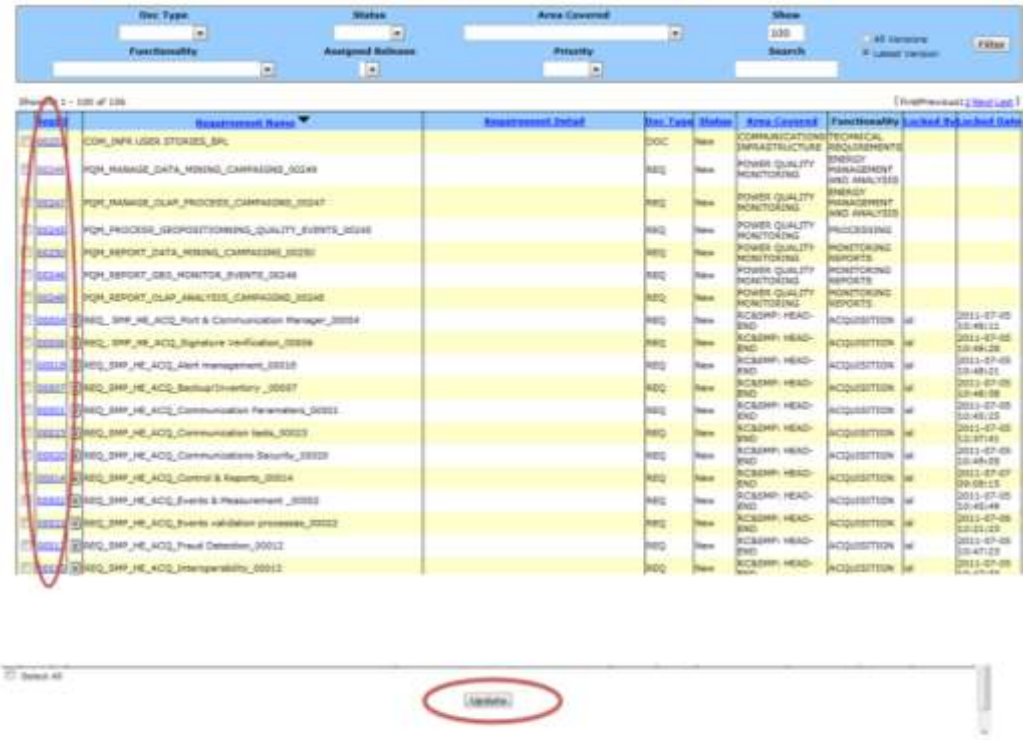
3.3.2.5 Associate user stories/scenarios to requirements

1. **Find the user story in the 'Requirements' tab and edit it (click in user story/scenario ID)**

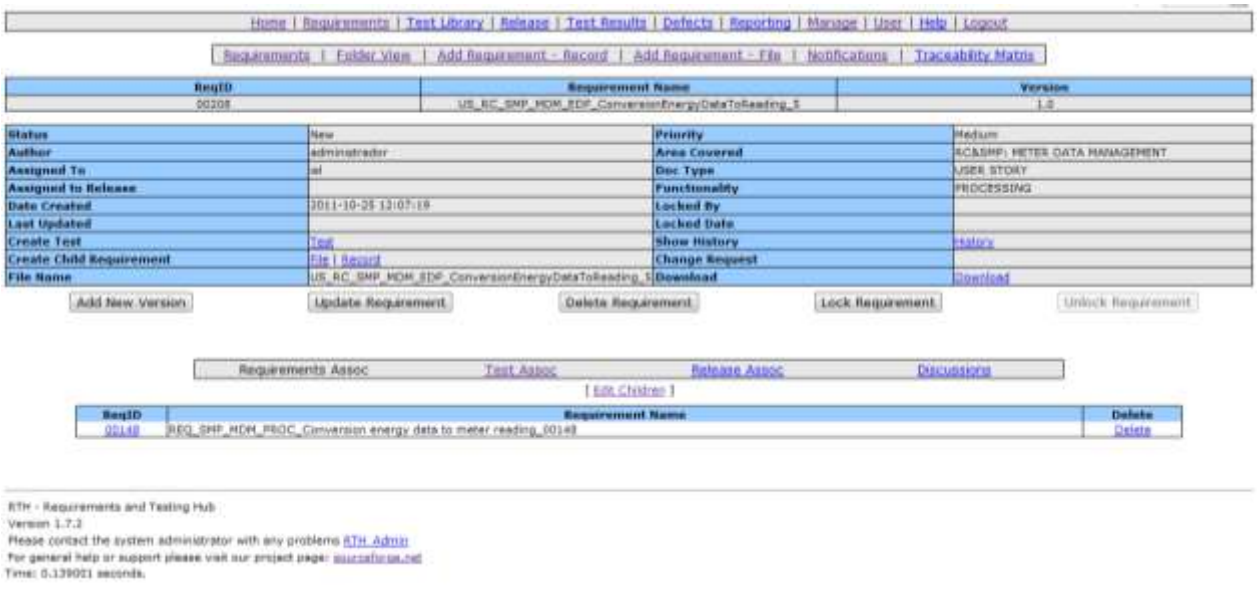


2. **Go to 'Edit children'.**

3. **Select the requirement you want to associate and click in 'Update'.**

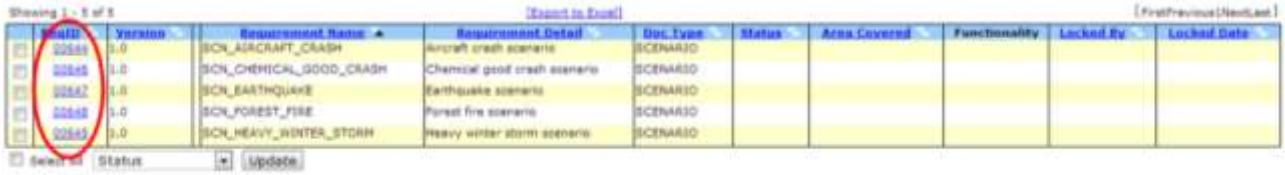


Your user story/scenario is associated to the differents requirements that were previously selected:



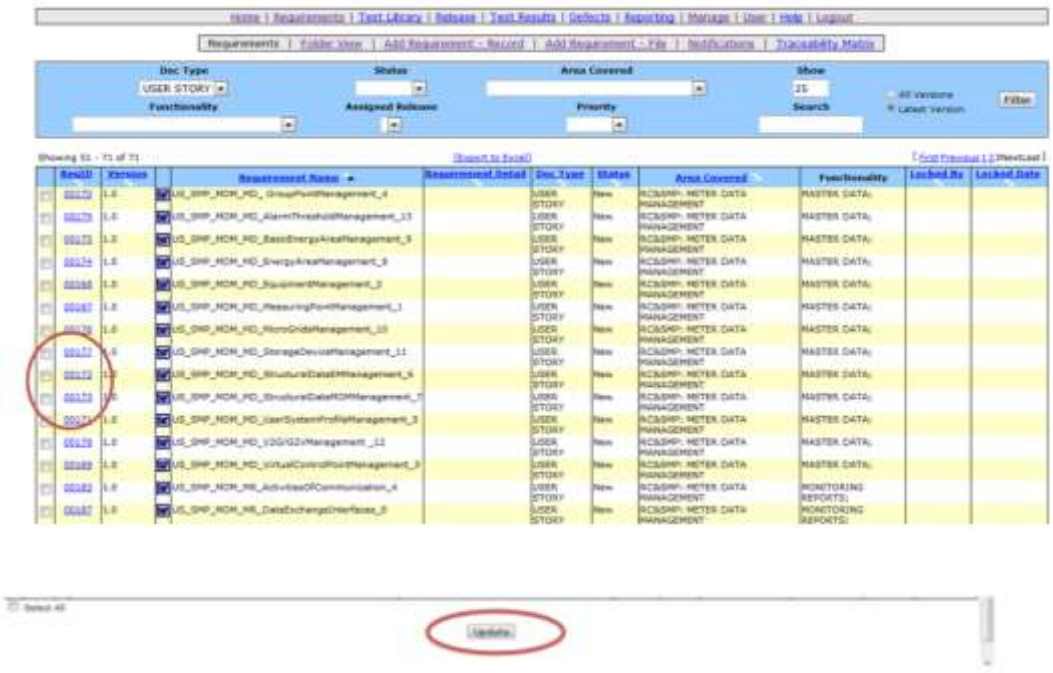
3.3.2.6 Associate scenarios to user stories

1. **Find the user story in the 'Requirements' tab and edit it (click in scenario ID)**

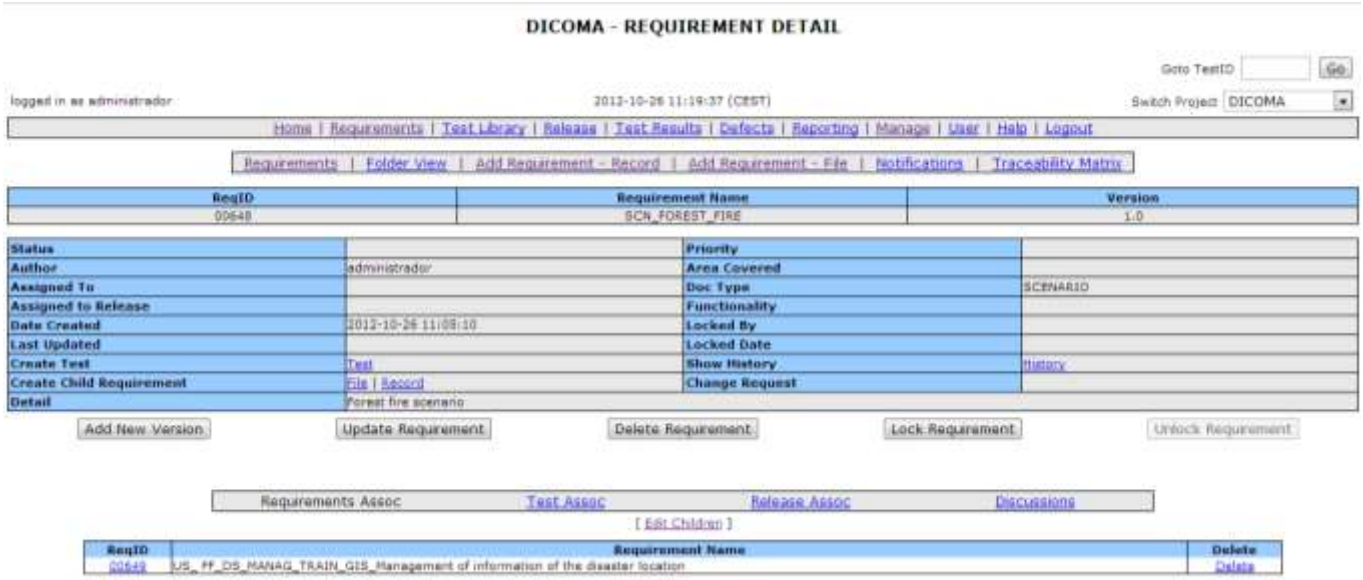


2. **Go to 'Edit children'.**

3. **Select the user story you want to associate and click in 'Update'.**



Your scenario is associated to the differents user stories that were previously selected:



3.3.2.7 Users and password

COMPANY	USERNAME	FIRST NAME	LAST NAME	E-MAIL	PASSWORD
Answare Tech	answaredicoma	Tonny	Velin	tvelin@answare-tech.com	dicoma
Athena GS3 Security Implementations Ltd.	athena	Alon	Moss	alonm@athenaiss.com	dicoma
Centre de Visio per computador – Image Secuence Evaluation Lab	Cvc	Xavier	Roca	xavir@cvc.uab.es	dicoma
DeustoTech – Compunting	Deustodicoma	Pablo	García Bringas	pablo.garcia.bringas@deusto.es	dicoma
Finnish Meteorological Institute/Arctic Research and Meteorogical Research Groups	Fmi	Timo	Sukuvaara	timo.sukuvaara@fmi.fi	dicoma
Indra Software Labs	Isldicoma	Eloy	Gonzalez Ortega	cmurphy@indra.es	dicoma
Mantis	Mantis	Aydin	Can Polatkan	aydincanpolatkan@mantis.com.tr	dicoma
Mattersoft Oy	mattersoft	Ms. Laura	Niittymäki	laura.niittymaki@mattersoft.fi	dicoma
Mobisoft Oy	Mobisoft	Pekka	Eloranta	pekka.eloranta@mobisoft.fi	dicoma
Oulu University of Applied Sciences	Oulu	Kirsi	Koivunen	Kirsi.koivunen@oamk.fi	dicoma
Savox Communications Oy	Savox	Tomi	Kankainen	tomi.kankainen@savox.com	dicoma

Universitat de Girona / eXit Group from the Institute Informàtica i Aplications	Udgdicoma	Roberto	Petite	roberto.petite@udg.edu	dicoma
Universitat Politècnica de Catalunya	Upc	Juan	Hernández Serrano	jserrano@entel.upc.edu	dicoma
University of Seville / Electronic Engineering Department (DIE)	Uds	Clara	Lujan	clara.lujan@gie.esi.us.es	dicoma
VTT Technical Research Centre of Finland	Vttdicoma	Jukka-Pekka	Laulajainen	Jukka-pekka.laulajainen@vtt.fi	dicoma
Infotripla	Infotripla	Kimmo	Ylisiurunen	kimmo.ylisiurunen@infotripla.fi	dicoma
Netcad	Netcad	Serkan	Gazel	serkan.gazel@netcad.com.tr	dicoma

3.4 Existing challenges in the domain

Different kind of challenges exist currently with the state-of-the-art systems in the disaster control management domain. In this section, existing challenges in the disaster systems and methods of each participating country will be listed here.

In Finland, an interview was circulated concerning : The transmission of important information of the operational higher officers from different authories (police, fire and resque, medical emergency) in disaster and crisis situations. This interview included such parts as: situation today, future visions and needs and wishes.

- Inputs: interviews and surveys with target user group, domain knowledge from

System in Use/Country	Challenge # 1	Challenge # 2	Challenge # 3	Challenge # 4
Authority network called 'VIRVE'/Finland	Hard to get enough information	Different authorities syncronize poorly	Many different persons involved	Information is fragmented and defective

3.5 Interaction methods and devices

In this section, relevant interaction methods and devices per country are presented, for example methods (& devices) which are already used, and new methods (& devices) which could be beneficial to be introduced.

The development of new innovative products is many times driven by technological advancement while leaving behind the identification of real user needs. Real use cases are necessary to form

and determine the potential and directions of the new products. Technical details and interaction between different systems can be evaluated based on use cases.

There exist various new techniques for the emergency response personal depending on the devices that they are aiming to use. The interaction methods are many times based on [2]:

- voice
- multitouch devices
- tangible and pen-based interaction interfaces

Voice interaction is the most common and natural way to disseminate the crisis information between different participants. However, crisis management requires planning, coordination and collaboration where the use of plain voice interaction and other traditional methods (pen and paper, paper maps etc.) are not the smoothest and most efficient choices. IT tools have to be integrated with a dedicated way in order not to violate existing specifications.

Well-known single- and multitouch techniques for the translation and rotation of virtual objects have been developed and evaluated. These techniques already present the state-of-the art and are used by many people Multitouch technology includes devices, from smart phones, tabletops, tablet computers to large interactive walls. Multitouch devices enable multiple interactions through a touch-based interface. Especially tabletops [3] and interactive walls allow multiple users to work at the same time with the same system.

Pen-based interaction is one of the most typical ways of interaction in everyday life. The benefits of using pen and paper are diverse: natural for humans, fast, precise and usable. Therefore, digitals pens are utilized in many tasks, and nowadays even large touch displays can be equipped with digital pens for precise input. Classical methods in disaster informatics nowadays also include paper maps and magnetic labels.

New Required Interaction Methods and Devices/Country	**+++**	**----**
Coherent management system between authorities/Finland	All the leaders are in the same physical or virtual place	How to the maintenance, training, system development etc. organized?
Helmet cameras/Finland	Real-time video	Available communication platform limits the video

4 Produce design solutions

New technical solutions to the disaster management domain will be created by combining the user centred design factors with the technological development. The designed solutions should be developed together with the people working in the special field and gaining the first-hand knowledge of the user experience and expectations. Requirements of the working environment will bring forth the important aspects regarding the software and hardware interaction.

The information flow between different participating authorities has to support forming the common operational picture (COP) of the disaster. COP systems will differ greatly depending on the applications area. An earthquake disaster management and warning system requires a different COP than the control centre operations in a chemical disaster. Implementing too many features in one system will be neither economic nor would it be operationally applicable. The evaluation for the suitable COP has to regard the following features:

i) Identify appropriate properties and technologies etc. which have to be implemented and also the less relevant components
ii) Evaluate and compare to the current state-of-the-art situation

4.1 Forming COP in disaster management

The common operational picture (COP) is an integrated result of various needs of the first responders and people operating with the Command, Control and Communications Centre (C4). COP handles the information that is needed to create and maintain a picture of the emergency situation. The common picture window must be shared among multiple operating agents so that they can co-operate in a coordinated way horizontally and vertically. Different agents also need separate picture windows that correspond to their particular responsibilities.

In EU FP7 COPE (Common Operational Picture Exploitation) project during 2008-2011, the goal was to achieve significant improvement in emergency response management. New solutions were created by combining a user oriented human factors with technological aspects. They defined the Common Operational Picture [4]:

- COP is an outcome of a joint functioning of human actors and technology
- COP is related to awareness of:
  - situation-specific operational goals and resources and their relationship to overall purpose of emergency response
  - actor's (and other authorities') own activity and effect on the system
  - constraints/limitations on action (e.g. rules, laws, limited resources)

4.2 Prototyping design solutions

4.2.1 Paper prototypes and low-fidelity mock-ups

Paper prototyping and other low fidelity prototyping methods can be used to validate design decisions early on in the product development. The strength of paper prototyping is that it allows testing basic product and interface concepts without writing any software code. With paper prototyping, product developers can also more easily create mock-ups for novel control device interfaces, such as table-top computers.

Other low fidelity prototyping methods include:

- *Wireframes*, specifying information architecture and interaction logic - mainly used to communicate design within the development team
- *Simple interactive "click-through" prototypes* - used to validate information model and task structures with end-users

The low fidelity prototypes are usually validated via a controlled user evaluation done in a lab environment. Usable methods include: *participatory design review* with end-users, developers and usability engineers, *think aloud protocol* with end-users and *heuristic evaluation* with a usability expert. These methods are explained in more detailed in section 5.2.1.

4.2.2 Functional UI prototypes

A natural evolvement from low fidelity mock-ups is to test the actual visual components used in the final product. Functional UI prototypes include a limited functionality of the system and show a simulated scenario. The purpose of a functional UI prototype is to validate:

- *knowledge continuity* across the system (workflow)
- *distribution of functionality* (efficiency of use)
- *consistency of the system*, from different point of views:
    - perceptual consistency (look and feel)
    - lexical consistency (used terminology)
    - syntactic consistency (interaction logic) and
    - semantic consistency (understandable functionality)

Functional prototypes can be validated in a controlled user evaluation similarly to the low fidelity prototypes; however such evaluations do not always capture all possible scenarios. Once the design has evolved to an alpha release stage the product should be validated using a user trial in either a real field environment or at least simulated, realistic usage scenario. These methods are explained in more detailed in section 5.2.2.

5 Evaluation of use

Each stage of development (a prototype) should be evaluated in order to validate the current design and refine the design for the next iteration. In this section we briefly describe the common metrics used in measuring the usability of a product as well as the common methods of evaluation (in both controlled environment and in field conditions).

5.1 Usability metrics

The usability is not a single, one-dimensional property of a user interface but has multiple components. Traditionally these components have included at least: *learnability*, *efficiency, memorability, errors* and *satisfaction* [Nielsen, J. *Usability engineering*, Morgan Kaufmann Publishers Inc., 362. ISBN 0-12-518406-9, 1993 and Shneiderman, B. *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1997]. These main usability goals are listed and described more below:

- *Learnability*: How easy is it for users to accomplish basic tasks the first time they encounter the design?
    - The system should be easy to learn so that the user can rapidly start getting some work done with the system
- *Efficiency*: Once users have learned the design, how quickly can they perform tasks?
    - The system should be efficient to use, so that the once the user has learned the system, a high level of productivity is possible
- *Memorability*: When users return to the design after a period of not using it, how easily can they re-establish proficiency?
    - The system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again
- *Errors/ effectiveness*: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
    - The system should have a low error rate, so that the users make few errors during the use of the system, and so that if they do make errors they can easily recover from them. Further, catastrophic errors must not occur
- *Satisfaction*: How pleasant is it to use the design?
    - The system should be pleasant to use, so that the users are subjectively satisfied in using it; they like it

Table 1 Usability metrics

Usability metric	Evaluation measurement
Satisfaction	Anonymous questionnaire (about how pleasant the system is to use)
Efficiency	Number of steps required to perform certain task(s)
Learnability	Time difference between performing certain task(s) for the first time and after that
Memorability	Field observations by a usability professional
Errors/ effectiveness	Number of errors when using a system, in certain task(s)

5.2 Evaluation methods

The above mentioned usability metrics can be evaluated using a number of methods. Namely there are two different approaches that can be applied: a controlled user evaluation in a lab environment and a field trial. Controlled evaluations are usually applied if the evaluated system is not yet finished or if the focus is evaluating a sub-set of functionality. In a controlled evaluation, exact measurement of efficiency, learnability and effectiveness is possible. On the other hand, validation of the whole system in a realistic usage scenario is only possible through a field trial (or a simulated field trial). The evaluation methods are briefly explained in the following sub-sections.

5.2.1 Usability evaluation

Think aloud protocol is used in usability evaluations where a prototype user interface is tested with an end-user in a laboratory environment. In the test setting the user is performing a set of tasks and told to think aloud while doing the tasks. The test is usually recorded using video other equipment such as eye-tracking to capture possible usability issues.

Heuristic evaluation is an evaluation done by a usability professional assessing a user interface according to a set of heuristics. Nielsen's ten usability heuristics are the most commonly used framework for heuristic evaluation: Nielsen, J. Enhancing the explanatory power of usability heuristics. *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*. ACM, 1994. Pp. 152]. This method is not limited to only these heuristics though and can be refined for a specific domain. The Nielsen's ten heuristics for user interface design are listed below

- **Visibility of system status -** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
- **Match between system and the real world -** The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

- **User control and freedom -** Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
- **Consistency and standards -** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. Users recognize a standardized environment better and this makes the use of it easier (e.g. user interface of a car).
- **Error prevention -** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
- **Recognition rather than recall -** Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
- **Flexibility and efficiency of use -** Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
- **Aesthetic and minimalist design -** Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
- **Help users recognize, diagnose, and recover from errors -** Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- **Help and documentation -** Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Participatory design review or user-centred walkthrough is a method where a group of people evaluates a user interface together. The group consists of *end-users*, *product developers* and *usability professionals*. The end-users should be representative of the target user group and are considered primary participants in the usability evaluation. Product developers answer questions about the design suggest solutions for the problems. Usability professionals are usually facilitating the discussions and can provide feedback on the design and suggest improvements.

The walkthrough method consists of printed screens and task description. The whole group goes through each scenario writing down their view on how they would accomplish the task using the system. For each scenario the team discusses the findings, analyses usability problems and decides future actions. The presence of variety of stakeholders allows potential synergy to develop that can lead to collaborative solutions. Using the walkthrough method the project personnel gains an early systematic look at the product features and satisfaction data from end-users. This method is most useful to evaluate the *paper prototypes* of the system.

5.2.2 User trials with the complete system

Here we describe the methods evaluating usability in user trials:

Observation can be used along with the trial of the complete system. Observations can reveal tasks that were misunderstood or tasks which were not specified in the product development.

Logging actual use can reveal bugs and problems with the efficiency of the system. Usage logging can be used in a field trial or in a controlled user test to find out the efficiency, learnability and effectiveness of the system.

User feedback can be acquired after a trial using a follow-up questionnaire. The feedback will give a holistic perspective about the perceived performance of the system and how well it functions in a realistic usage scenario. User feedback can also unveil changes in user requirements. User feedback is an important measurement of validity of the product as well as way to find out unexpected problems or errors, which otherwise would not be detected.

5.2.3 Summary of evaluation methods

In the following table we summarize the most important methods for evaluating the system from the user's perspective. The table summarizes the stage of applicability, the need for users in the test and the main advantages and disadvantages of the method. The table is adapted from [add reference to: Nielsen, J. *Usability engineering*, Morgan Kaufmann Publishers Inc., 362. ISBN 0-12-518406-9, 1993].

Table 2 Summary of evaluation methods

Method name	Lifecycle stage	User needs	Main advantage	Main disadvantage
Heuristic evaluation	Early design, "inner cycle" of iterative design	None	Finds individual usability problems. Can address expert user issues	Does not involve real users, so does not find "surprises" relating their needs
Performance measures	Competitive analysis, final testing	At least 10	Hard numbers. Results easy to compare	Does not find individual usability problems
Thinking aloud	Iterative design, formative evaluation	3-5	Pinpoints user misconceptions, cheap test	Unnatural for users. Hard for expert users to verbalize
Observation	Task analysis, follow-up studies	3 or more	Ecological validity; reveals users' real tasks. Suggests functions and features	Appointments hard to set up. No experimenter control.
Questionnaires	Task analysis, follow-up	At least 30	Finds subjective user preferences.	Pilot work needed (to prevent

	studies		Easy to repeat.	misunderstandings)
Interviews	Task analysis	5	Flexible, in-depth attitude and experience probing	Time consuming. Hard to analyse and compare
Focus groups	Task analysis, user involvement	6-9 per group	Spontaneous reactions and group dynamics	Hard to analyse. Low validity
Logging actual use	Final testing, follow-up testing	At least 20	Finds highly used (or unused) features. Can run continuously	Analysis programs needed for huge mass of data. Violation of users' privacy
User feedback	Follow-up studies	Hundreds	Tracks changes in user requirements and views	Special organization needed to handle

6 Conclusions

UCD is an established methodology that focuses on the users' needs and requirements to create efficient technological solutions. UCD approach is essential when developing and designing new user interfaces. In order to create user-friendly solutions, it is necessary that the involved users are taken into the design phase in an early stage. UCD provides different applicable methods at different development stages including contextual and behavioural analysis. There exist several useful methods how to observe the user behaviour and form UCM models that guide the development and how to benefit from available technologies in real-world applications.

7 References

[1] Diehl, S., Neuvel, J.M.M., Zlatanova, S. and Scholten, H.J. 2006. “Investigation of user requirements in the emergency response sector”: the Dutch case.

[2] K. Nebe, F. Klompmaker, H. Jung et al. (2011) Exploiting New Interaction Techniques for Disaster Control Management using Multitouch- , Tangible- and. In *HCII 2011*.

[3] http://www.microsoft.com/surface

[4] EU FP7 COPE (Common Operational Picture Exploitation), http://cope.vtt.fi/