Author version of the paper published in: International Journal of Information and Communication Technology (IJICT), 12 (3-4), pp. 364-392, 2018.

A Framework combining Agile, User-centered design and Service Oriented Architecture approaches for Collaborative Disaster Management system Design

Karima Ait Abdelouhab and Djilali Idoughi
Laboratory of Applied Mathematics – LMA,
Faculty of Sciences Exactes,
University of Bejaia,
Bejaia, 06000, Algeria
Email: aitabdelouhab@gmail.com
Email: djilali.idoughi@gmail.com

Christophe Kolski
LAMIH-UMR CNRS 8201,
University of Valenciennes and Hainaut-Cambrésis,
F-59313 Valenciennes 9, France
Email: Christophe.kolski@univ-valenciennes.fr

***Abstract*— Disaster Management is a special type of human complex organization in which heterogeneous human actors belonging to different authorities collaborate and work together with the shared aim to solve, or at least reduce, the disaster situation. Thus, the collaboration in this case within team members and with other teams operating at the disaster site(s) is very critical and complex; the achievement of the desired goal heavily depends on this collaboration. Interactive and easy to use services in these scenarios are very valuable and necessary as they can improve collaboration, coordination, and communication amongst team(s) to achieve the desired goals. For this purpose, in this paper, we propose a novel design framework for complex disaster management systems. We combine agile characteristics and principles, user-centered techniques and service oriented architecture paradigm. Our aim is to take into account the needs of the disaster managers in an iterative development process, to improve the human actors' involvement in the design projects, to offer the possibility to accept any changes in order to produce highly usable and interactive service based collaborative services.**

Keywords— *User-centered design; agile method; Service oriented architecture; Service design; Disaster management, collaboration.*

INTRODUCTION

Disaster Management (DM) is a complex and multi-disciplinary process of planning and implementing measures that ensure efficient prevention/ response operations whenever an undesirable event occurs (Benssam e*t al*., 2013). Consequently, disaster management is a special type of human complex organization in which the communication and collaboration between several types of actors become a must. All organizational processes, techniques and means used by an organization to prepare for and cope with the occurrence of a disaster and then learn from the event are termed as disaster management activities (Coppola, 2006). These activities can be grouped into four phases: (1) prevention, (2) preparation, (3) response (emergency management) and (4) recovery. The collaboration is essential in

all phases of disaster management. It is very important for dealing with both natural and technological hazards and disasters and underlying consequences (Waugh *et al.*, 2006) (Fig. 1). The effectiveness of disaster management efforts depends mainly on the ability of the participating actors to work together and promote the information sharing practices to enhance decision making along the whole process (Markus *et al.*, 2009). Consequently, the importance of Information and Communication Technology (ICT) lies in facilitating this kind of coordination activities rather than in information processing (Janssen *et al.*, 2010).

Fig. 1 Collaboration in all phases of Disaster Management (adapted from (Borges, 2014)).

However, the development, collaboration and use of Information and Communication Systems (ICS), in the field of disaster management, are very complex and face several and diverse challenges such as:

- First challenge: the complex task environment requires multiple organizations at different levels to transform autonomous actors into interdependent decision-making teams (Maiers *et al.*, 2005).
- Second challenge: the disaster management process involves different scenarios of resolution of the disaster. Multiple and diverse of teams, belonging to different organizations, are needed to collaborate in order to reach a common goal and avoid misinterpretation of situations and consequently, enhance the management process at both levels strategic and operational.
- Third challenge: the interoperability between different systems of disaster management of different organizations is an important issue on the organizational, technical and semantic levels.
- Finally, the social challenge in a situation of disaster is very complex (Benssam e*t al*., 2013).

Many alternatives are possible for designing new disaster management systems supporting multiple organizations at the different levels, involving different teams for crisis response and sustaining the heterogeneous information sources. Thus, for designing and developing such system satisfying these requirements, there are three software development approaches in the literature that share a common objective of efficiency in the resulting software, namely Service Oriented Architecture (SOA) (Groves, 2005; SOA Manifesto, 2009), agile methods (Abrahamsson *et al*., 2002; Agile Alliance, 2001) and User Centered Design (UCD) (ISO, 2010; Loup-Escande et al., 2014). However, these approaches differ in nature, and cover different area.

Following the comparative study between these three approaches given by Ait Abdelouhab et al (2014), agile methods and SOA seem to forget the end user and usability altogether, i.e both methods do not take into account how to ease use for the end users. Besides, agile methods focus on how to organize the delivering working software. Moreover, they focus only on code development. While SOA is a development approach decomposing all in services (software components). In addition, it doesn't stress in end user feedback. Whereas, user centered design (UCD) is an approach that places the end user of the system at the centre of the design (Norman and Draper, 1986). In addition, UCD aims to produce systems or software that are highly usable involving methods and techniques that are oriented towards usability (Blomkvist, 2005). However, this approach takes more time (time-consuming) during the collection of end users needs.

To overcome these drawbacks, in this paper, we propose a novel design framework for developing complex disaster management systems that combines agile characteristics and principles, user-centered techniques and SOA paradigm (This combination is shown in the Fig. 2). Our framework takes the advantages of these three approaches; It identifies ways to apply appropriate user centered design activities throughout the agile development process and using the different best practices of agile methods mainly user stories, unit tests, etc. (Beck 2000) while respecting the SOA paradigm. Our objective is to take into account the user needs in an iterative development process, to improve the human actors' involvement in the design projects, to offer the possibility to accept any changes in order to produce highly usable and interactive SOA based collaborative services.

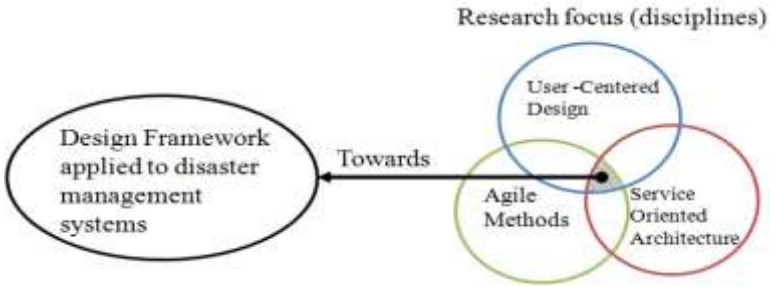


Fig. 2 Combining approaches to design interactive disaster management systems.

In this paper, we extend our paper published in the ICT-DM conference (Ait Abdelouhab et al, 2014) (extended conclusion and state of art, revised design framework, detailed case studied, added discussion). This article is a first step towards bridging the gap between User-centered design and agile principles integration in order to provide service designers/developers with a comprehensive framework for the design, implementation and deployment of SOA based interactive services. In this present paper, the aim of the revised (and extended) design Framework is to provide interactive and usable collaborative services based on SOA for disaster management.

The section II gives an overview of the state of the art on SOA based services, user-centered design (UCD) and agile approaches. In section III, we present current related works concerning methodological aspects of disaster management system design. In section IV, we highlight the main phases of the proposed design framework for disaster management system incorporating UCD principles in an agile process while relying on service-oriented architecture. This framework is applied to a typical disaster management Earthquake Management case study (section V). In Section VI, we dress performance evaluation of the Agile-UCD-SOA Framework. In section VII, we follow with discussion about our framework and the researches practices. This article ends with a conclusion and perspectives for future research.

STATE OF-THE-ART ON SOA, UCD AND AGILE APPROACHES

This section presents a state of art on SOA based services, user-centered design (UCD) and agile approaches.

A. Service Oriented Architecture (SOA)

SOA as a paradigm refers to service creation, interaction, presentation, and integration infrastructure capabilities to build business-level software based on reusable components (Haubrock *et al.*, 2007).

Many definitions have been given to SOA. One of these is given by Erl (2009): "SOA is an architectural model that aims to enhance the efficiency, agility, and productivity of an enterprise by positioning services as the primary means through which solution logic is represented in support of the realization of strategic goals associated with service-oriented computing*"*.

According to Mahmood (2007), SOA offers promising opportunities for enterprise application integration while reducing the cost of application development, improvement in flexibility and scalability. The interoperability is also one of the opportunities provided by SOA in order to offer flexibility to adapt the changing technologies (Liu et al. 2012). SOA allows enterprises and their IT systems to be more agile to the changes in the business and the

environment. SOA are increasingly deployed to achieve distributed systems that are modular, flexible and extensible (Millard, 2009).

In a SOA based system, the business and technical processes are implemented as services. Each service represents a particular functionality that maps explicitly to a step in a business process (Erl, 2005).

Several SOA methodologies are available in the literature such as: SOAD (Services-Oriented Development of Application (Zimmerman *et al.*, 2004), Service Oriented Modeling and Architecture (SOMA) (Arsanjani et al., 2005), Thomas Erl's methodology (Erl, 2005)**,** the methodology for the development of Web services (Papazoglou and van den Heuvel, 2006), the service-based user interface approach (Idoughi, 2008; Idoughi et al., 2010), etc.

The Majority of SOA methodologies propose to divide the SOA development lifecycle into six phases: service-oriented analysis, service-oriented design, service development/construction, service testing, service deployment/transition, service administration/management.

In our work, SOA is an important paradigm, due to its advantageous characteristics namely service reusability, loose coupling, interoperability, etc. In addition, it allows us to transform all business needs into business services. Moreover, the different legacy systems in the civil protection organization can be analyzed and reused.

B. User-Centered Design

Human-Centered Design (HCD) is "an approach to interactive systems development that aims to make systems usable and useful by focusing on the users, their needs and requirements, and by applying human factors/ergonomics, and usability knowledge and techniques*"* (ISO, 2010). The aim of this approach is to enhance effectiveness and efficiency, to improve human well-being, user satisfaction, accessibility and sustainability; and to counteract possible adverse effects of use on human health, safety and performance.

User-centered Design (UCD) is a software design philosophy and a process that puts the users and their needs central to the project life cycle (Loup-Escande et al., 2014). The international standard ISO 9241-210 (ISO, 2010) defines User-centered Design as an approach to software and hardware design that identifies four activities and six principles. The user centered design four activities: (1) understanding and specifying the context, (2) specifying the user needs, (3) produce design solutions to meet user requirements, (4) evaluate the designs against requirements.

Several UCD methods have been proposed in the literature with the aims to make end users and their experiences a focal point of the design process (Loup-Escande et al., 2014) such as: Goal Directed Interaction Design (GDID) (Cooper *et al.*, 2007), Contextual Design (CD) (Beyer and Holtzblatt, 1998), Scenario-Based Design (SBD) (Carroll, 2000; Rosson and Carroll (2000), The Human-Centered Systems Development Life Cycle (HCSDLC) model (Te'eni *et al.*, 2007; Zhang *et al.*, 2005), Persona-Scenario-Goal Methodology (Aoyama, 2007), Persona-based approach (Idoughi et al., 2012), etc.

These methods focus mainly on the utility and usability of an interactive system in order to: reduce errors, satisfy users and facilitate its learning and use, decrease training and support costs. One important aspect of UCD is the collaboration between users and developers to build software solutions, so that each group brings its own experience (Robey *et al*., 2001).

The multiplicity of end user profiles in civil protection organization confirms us that it is very important to tailor user-centered design because the end users by their nature cannot express all their needs at once, and indeed they have tendency to change them often.

C. Agile development process

The Agile manifesto (Agile Alliance, 2001) consists of four values and twelve principles. The Agile manifesto four values are as follows: (1) *individuals and interactions* over processes and tools, (2) *working software* over comprehensive documentation, (3) *customer collaboration* over contract negotiation and (4) *responding to changes* over following a plan.

Agile methods are incremental, cooperative, and adaptive (Abrahamsson *et al*., 2002, Yu and Petter 2014). These methodologies focus on people, communication and the ability to adapt to change rather than the process, tools and predictive planning (Mushtaq and Qureshi, 2012).

The main agile methods are the following: **XP** (Extreme Programming, www.extremeprogramming.org), **Scrum** (www.controlechaos.com), **DSDM** (Dynamic Software Development Method, www.dsdm.org), **ASD (**Adaptive Software Development, www.adaptived.com), **Crystal** (www.crystalmethodologies.org), **FDD (**Feature Driven Development, www.featuredrivendevelopement.com), etc.

The majority of agile methods embrace change in their concept. In fact, the agile methodology encourages rapid and flexible response to changes by emphasizing on user involvement and his/her feedback, and on delivery of several small releases.

In our proposed framework, we rely on the agile development process because they allow us to open a communication between designers, developers and end users (members of civil protection). In addition, agile methods enable us to prioritize needs and begin the development with needs having the highest priority. As a result, agile methods allow us to deliver several small releases for the disaster managers. Consequently, we get frequently feedback.

The next section describes related work concerning methodological aspects of disaster management system design

RELATED WORK ON DISASTER MANAGEMENT SYSTEMS DESIGN

In recent years, several approaches have been proposed in the literature for disaster management; they offer models, guidelines, methods and planning procedures for the development of effective disaster plans/ management. In this section, we present several representative approaches.

Disaster management is characterized by a strong involvement of many and various actors. For this, developing usable and interactive services to be accepted by disaster managers and all the members of an emergency team requires an approach that takes care of different aspects of the decision-making process, from individual and social dimensions to knowledge and activity-centered perspectives.

- Affeltranger *et al*. (2007) in their work describe how to build ERMA (ElectronicRisk Management Architecture). The aim of ERMA is to help authorities to conduct their tasks successfully. The main concept of building of ERMA relies on a user-centered methodology as well as on service oriented architecture.
- Humayoun *et al*. (2009) made a distinction in the collect of users requirements in a situation of crisis while the ERMA project does not make it. Humayoun *et al*. (2009) propose a methodology for designing an interactive system to be used in mobile and pervasive scenarios for emergency management. For this, they combine user-centred design approaches and software engineering approaches tailored for distributed architectures. Moreover, they define a twofold methodology for getting the requirements and feedbacks from real end users: (1) a top-down approach, and (2) a bottom-up approach. The system engineering methodology uses UCD techniques.

We think that it is a good initiative to distinguish between the various needs and groups of users especially to design an interactive system dedicated to disaster management.

Uusually, agility is the ability to react quickly and easily to a stimulus. Applied to disaster management, this is also a useful concept: response organizations have well defined procedures and processes but at the same time should be able to recombine them during extreme conditions in order to solve anticipated events (Hanachi *et al*., 2012).

- Wood *et al*. (2013) focus on agile response for crisis management. They have developed a proposal based on the idea of collaborative agile workflows. Here, agility means people's ability to reconfigure knowledge, skills and resources on the fly at runtime during the occurrence of a crisis.
- Nawaz *and* Zualkernan (2009) questioned on how geographically dispersed agile team could ensure business continuity in the face of a catastrophic event like a terrorist attack or a hurricane. For this, Nawaz and Zualkernan (2009) present an analysis of unpredictable low base rate events in the context of agile software development that plunged a company and its development team into a crisis.

In our work, we focus on agility in the sense of how we can plan and organize the project of development of disaster management system as a series of iterations that will be guided by user needs in an agile process and accept any change of requirement of end users (disaster managers).

The organizations involved in disaster management often use different technological systems and means. Recently, the approaches of Web service-based information and process integration have been receiving much attention.

Moreover, Web services are being adopted as promising technology to support open and distributed decision making and coordination in business applications (Wang *et al*., 2004).

- Jiugang *et al*. (2009) propose the design and implementation of a Web based Geographical Information System (WebGIS) for government comprehensive emergency management, which is based on advanced technologies such as Service-Oriented Architecture (SOA) as well as Web Service.
- Braune *et al*. (2011) identify the core requirements of emergency management systems and present a new generation of modular, service-oriented and semantic-web based architecture for emergency management systems.

These studies show that service oriented architectures have a great deal of interest in the development of emergency systems because they keep the promise to significantly enhance efficiency in performing decision making and information sharing. Consequently, they enable a decision maker in disaster management to access data at runtime, run simulations without being aware of the actual implementation, and generate maps or reports for information aggregation.

Through the studies presented previously, we note that most of these studies do not define a comprehensive methodological framework in their development process of disaster management system. Furthermore, they consider most often only one concept that is user-centered design in the process of collecting user requirements or in the creation of user interfaces, or agility which is treated according to different definitions or service-oriented architecture which is taken into account to improve the sharing and exchange of information between decision makers. However, none of these works have integrated these three concepts into a single design framework. In this paper, we propose a novel framework combining user-centered design and agile method in order to provide collaborative interactive services based on SOA dedicated to disaster management. This framework is presented in the following section. It is illustrated by a case study.

PROPOSITION OF A FRAMEWORK COMBINING AGILE, UCD AND SOA APPROACHES

A. Main characteristics of the proposed framework

We propose a design framework based on the combination between the three approaches: User-centered principal, agile characteristic and service oriented paradigm, we call it Agile-UCD-SOA Framework. The proposed Agile-UCD-SOA framework is applied to an *earthquake management* case study.

The objective of the framework is to transform the conceptual models developed during the different phases of the framework and described in UML diagrams into visual structure (i.e. sketch into mock-ups, animated user-screen-based, etc.) and design structure (i.e. design and develop concrete interactive prototypes).

Our framework begins with the study of the organization (Civil Protection Department- CPD) in order to identify the business objective relative to the disaster management. This is one important characteristic of our framework.

Another characteristic of our framework consists to take into account the user requirements just in time. In other words, this characteristic allows the end users to express their business needs just in time. These needs are expressed by short and simple user stories.

The other important characteristic consists of prioritizing the business services. This prioritization of business services is performed in close collaboration with the end user and the development team in order to rapidly develop a high-level plan for the current iteration. This prioritization leads the development team to develop more rapidly and efficiency.

The last characteristic of Agile-UCD-SOA framework is introducing the evaluation of interactive service by taking into account of user interface in each iteration. For this, usability evaluation aspect is used.

For the modeling our framework, we use UML (Rumbaugh *et al.*, 1999). It is a language that allows models to be represented without defining their development processes. Thus, it can be used with any other software development process. Also, we use BPMN formalism (Business Process Modeling Notation). It is a standard for business process modeling that provides a graphical notation for specifying business processes.

B. Global view of the Agile-UCD-SOA framework

We propose dividing the Framework into four phases (Fig. 3): (1) Study of disaster management organization Business Analysis, (2) Just In Time (JIT) Disaster manager Requirements Analysis and Elicitation, (3) Iterations Prioritization and Planning, (4) Release to Iteration. The main roles associated with the realization of framework are: (1) Project Owner[1], (2) Business Analyst, (3) Development Team (Business Designer, Developer/s and Architects), and (4) Testers and Evaluators.

The main development activities are iterative and incremental. So, the iterativity and the incrementality are related to the fact that the framework is based on a succession of activities. Moreover, many interview sessions with the end users are necessary so they can express their needs. Consequently, we obtain for each iteration a minimum increment.

The framework is user-centered and follows an agile life cycle while respecting also contents of a SOA life cycle.

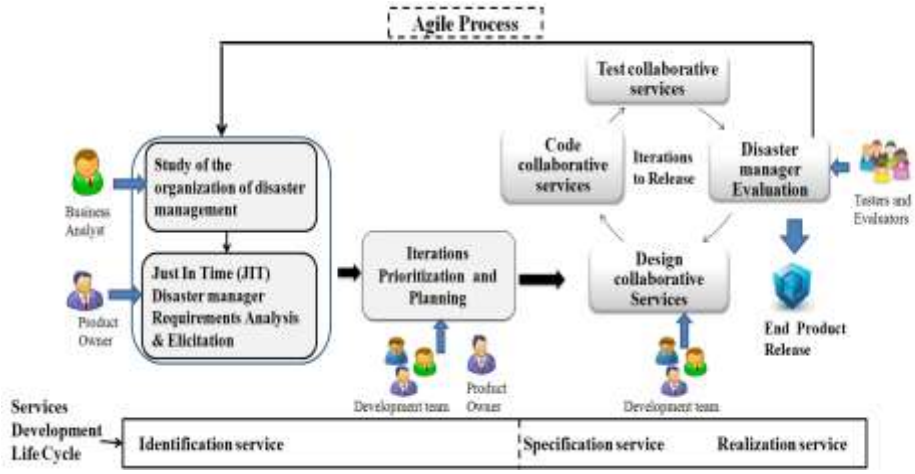


Fig. 3 Global life cycle of the Agile-UCD-SOA framework.

Its constitutive phases are more detailed and illustrated in the following section on a typical case study.

APPLICATION OF THE PROPOSED FRAMEWORK ON A TYPICAL CASE STUDY

A. The case study scope - The Civil Protection Department (CPD) of Bejaia city (Algeria)

The Algerian government has implemented a set of measures of prevention to intervene on the risks and its consequences, but also measures for disaster management, in order to secure the people's life and property and reduce the damage and impact of the occurred event. These measures are largely based on the applicable text in the field of disaster management under the law no. 04-20, of 25 December, 2004, on "**the prevention of major risks and disaster management in the context of sustainable development**"(www.joradp.dz). It is considered as the national act of disaster prevention and management.

In this paper, we focus on the coordination and communication issues occurring within **The Civil Protection Department (CPD) of Bejaia city (Algeria)** as well as the decision-making of the superior authority of the city namely the Wali. According to the law cited above, the civil protection department is organized as a national service, politically coordinated by the premier minister. It consists of central and local authorities, and includes public corps and existing private institutions within the national territory (Fig. 4).

[1] Project Owner: responsible for maximizing the value of the product and the work of the Development Team.

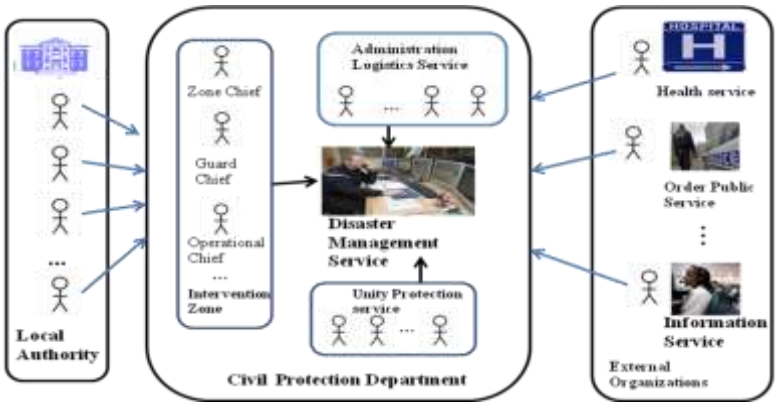


Fig. 4 Disaster Management services and External Partners within Algerian territory.

B. Application of the different phases on the case study

We provide in this section more details of different phases of our proposed Agile-UCD-SOA framework and their application on the case study.

Phase.1 Study of disaster management organization Business Analysis

The aims of this first phase are to study the complex organization of civil protection department by identifying: (1) the disaster's management requirements (2) business objectives relative to the disaster management, and (3) to understand and communicate the business environment context in which the targeted interactive disaster management system is to be developed. This phase is conducted following the three steps below.

Step 1.1 Business analysis

This step is performed by business analyst and designer to elaborate an initial *service model* which includes a first set of candidate services that can support disaster management services, processes and goals of the organization.

At the beginning, the business analyst deals with the hierarchical decomposition of the business domain of disaster management for the purpose of performing service analysis and business use case identification as service candidates.

Fig.5 shows a simplified view of the decomposition of the disaster management domain model (or functional mapping system). In addition, we have highlighted, in this model, the business processes related to the resolution of disaster and areas that are directly related to these processes. In this paper we will focus on the emergency management phase. In such scenarios, the process of disaster response is divided into two business processes: *Alert processing* and *Order of Execution.*

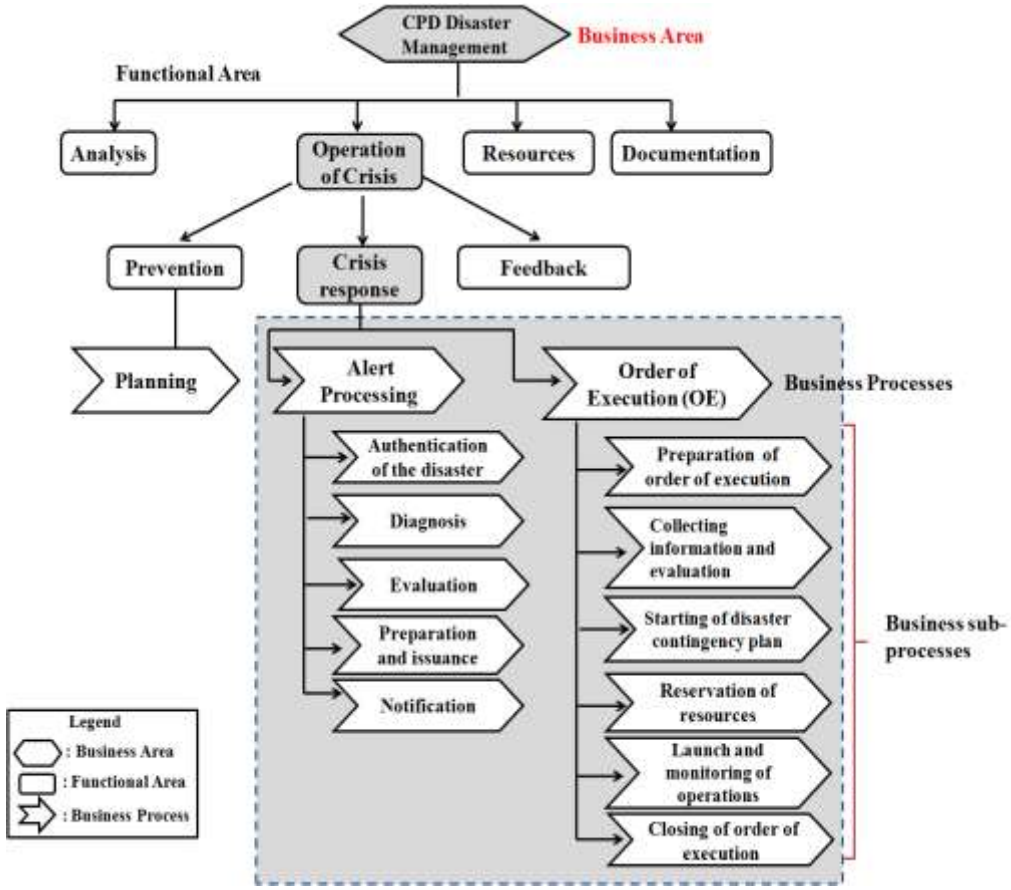


Fig.5 Decomposition of the business domain "Disaster Management".

Subsequently, we highlight the main business use cases supporting the business objectives along with the corresponding UML interaction diagrams. Thus, we identify the business use cases in relation with the business functional and business processes.

Fig. 6 shows the decomposition of business functional and business process into business use-case models. These business use cases gained through this decomposition are considered as good candidates for the high level of disaster management business collaborative services.

The scenario relating to these identified services (Fig. 6) is the following one:

- Upon receiving the alert, the guard chief uses the collecting information and assessment service for sending an evaluation report on the situation to the Wali of Bejaia. On receipt of this report, the Wali proceeds by the authentication service to verify and validate the degree of the disaster. Afterwards, he prepares and creates an execution order through editing/creating service and orders the execution of disaster contingency plan (ORSEC Plan).
- Launch and monitoring of operations service are activated by the operational commandant. This service consists of three services: (1) control on the availability of resources, (2) monitoring of operations, (3) starting of order of execution.
- During the intervention, the reservation service is consulted by the zone chiefs in order to reserve the equipments and / or personals necessary for the intervention operation. At the end of this operation, the director of civil protection closes the order of execution by using closing service. For this, he prepares a closure report and saves it.

.

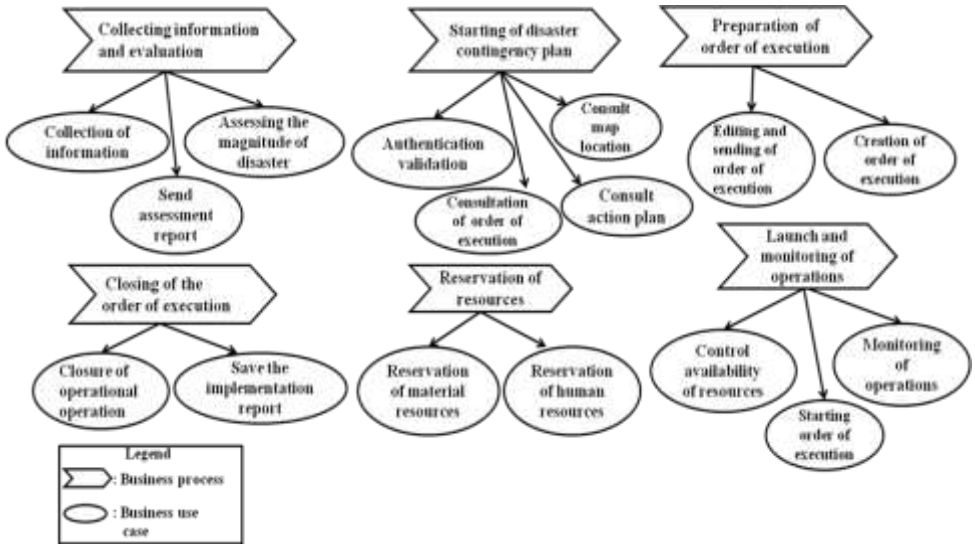


Fig.6 Example of candidate business services of Disaster management.

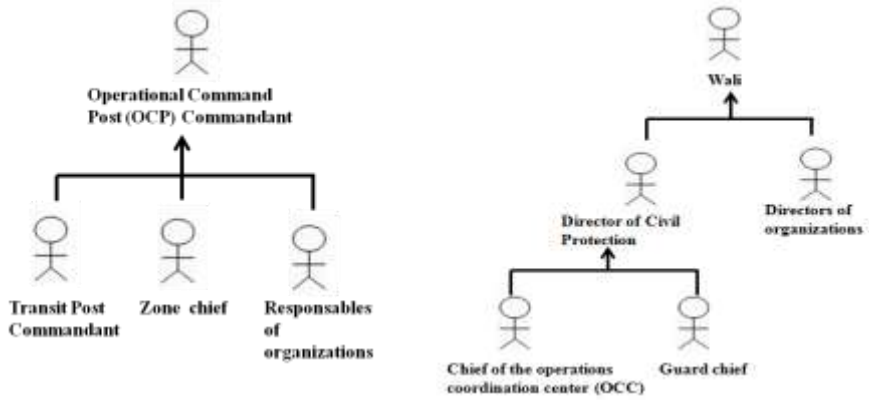
Step 1.2 Users and task Analysis

In order to devise successful interactive and usable services, we follow the user-centered design (UCD) approach.

In this step, we define and identify end user (disaster manager) profiles and their tasks in order to understand how these users are arranged during disastrous happenings, how and which tasks and information are exchanged among teams and with their respective operational centers.

By using User-Centered techniques, in collaboration with the Civil Protection of Béjaia, we interviewed commanders and generic actors of the most important organizations involved in emergency management. From the understanding of how Civil Protection works in Bejaia city during an emergency, we identify the underlying typology of actors (Fig. 7):

1. The actors who are involved directly during the intervention of emergency response as shown in the Fig. 7 (a).
2. The actors who manage the situation of the disaster from fixed command post (crisis cell) and allow for providing instructions/information to operators in the intervention as shown in the Fig. 7 (b).



(a) Typology of actors involved during the intervention.

(b) Typology of actors in fixed command post.

Fig. 7 Typology of actors.

Thereafter, a task analysis of these actors is done during this step to collect the important elements for the specification of *presentation services*. The latter are associated with the presentation layer of SOA. For this, we use a hierarchical model defining the tasks of the actors concerned by disaster management.

The main aims of task analysis is to show an overall structure of the main user tasks; it includes the overall users' responsibilities in processes, goals to achieve and tasks which users intend to perform to achieve goals.

Step.1. 3 Legacy system analysis

We use in this step the decomposition of existing systems in the form of application modules that can provide an implementation for business services previously identified. Therefore, we apply a bottom-up approach, *i.e.* starting from the existing system to the business services and business processes (Fig. 8).

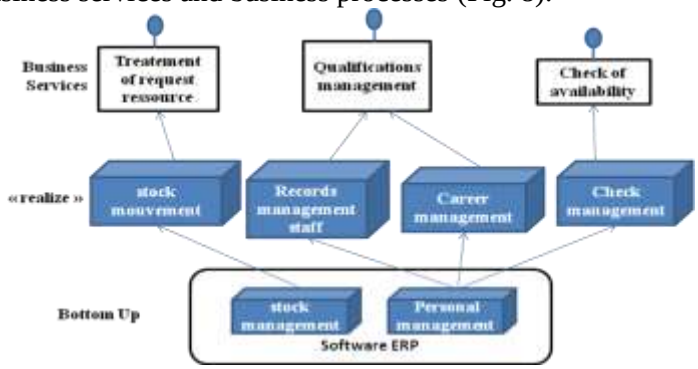


Fig. 8 Example of a decomposition of Legacy systems "Disaster Management".

Phase.2 Just In Time Disaster manager Requirements Analysis and Elicitation

This phase aims to capture, analyze and define requirements of disaster manager Just-in-Time when they are needed.

By using agile methods and XP in particular, in collaboration with the Civil Protection Department, the requirements are identified and expressed in terms of user stories by the product owner. Therefore, two steps are defined as follows.

Step 2.1 Identifying and creating user stories

In this sub-stage, it is about identifying and creating the user stories of disaster managers. User stories (Cohn, 2004) are collaborative design tools which help the disaster team to think through what the project needs to deliver from the perspective of those who will use it (Rivero et al., 2014). Cohn (2004) proposes the following template for requirement modeling (Fig. 9).

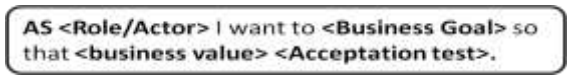


Fig. 9 A Template of User story.

Therefore, each user story encapsulates the whole knowledge (role, business goals, business value, etc.) about the potential user of the service. It serves as a communication tool between the project team members. For this, we need to extend this first description towards the second level of the description which uses the scenario concept. We use a technique of elicitation of requirements like an interview; brainstorming. As we can see from the template of a user story, we can easily extract some pertinent modeling elements as shown in the following process:

(1) From the Actor component, we can highlight the Actor types;
(2) From User goals, we can identify goals related to the application;
(3) From the Business goals, we can identify business use cases.

We give an example in (Fig. 10).

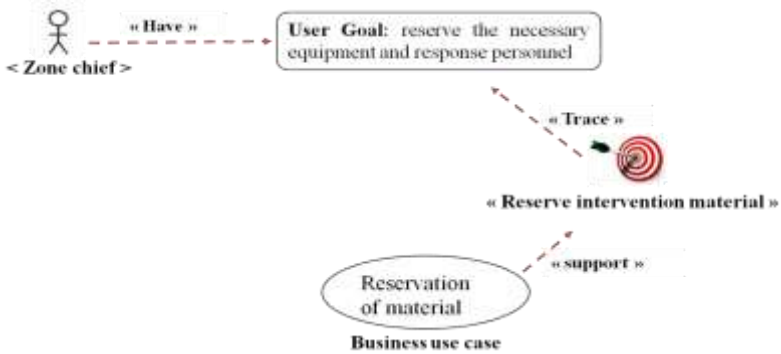


Fig. 10 Example of User story representation.

Step 2.2 Description of business scenarios

From the Business Goal (supported by business use case(s)) identified in the previous step, we describe relevant business scenarios relative to the end users (disaster manager) and their stories. The different interactions from these scenarios are represented with a UML sequence diagram. As a result, we can obtain realized activities which can define new web services, while the exchange of messages can then match the operations of these services. Fig. 11 shows a set of new web services obtained during the interaction with UML sequence diagram by specifying their operations such as a web service reservation and web service stock materials management.

Phase.3 Iteration Prioritization and Planning

Prioritization is a process whereby the end users and developer team cooperate together in order to place a number of business services in rank order based on their perceived or measured importance or significance.

This phase aims to prioritize the first business services identified in the previous phase, whose purpose is to rapidly develop a high-level plan for the next iteration.

In this phase, we borrow the principle given by Shahrbanoo *et al* (2012) in which business service prioritization must be done in the manner in which all kinds of stakeholders give different viewpoints and give their importance.

We start to prioritize the business services according the degree of importance. Before selecting business services for current iteration, we need to check the criteria of dependence between business services. This is an important point to fit the SOA project. So, to achieve this, business services which depend on each other must be grouped so that the groups of business services are independent. Then in each group the most dependent business services must be combined as a new business service. Business services in each group must be placed in two categories: 1) Business services which have high priority, 2) Business processes which have low priority (Fig. 12).

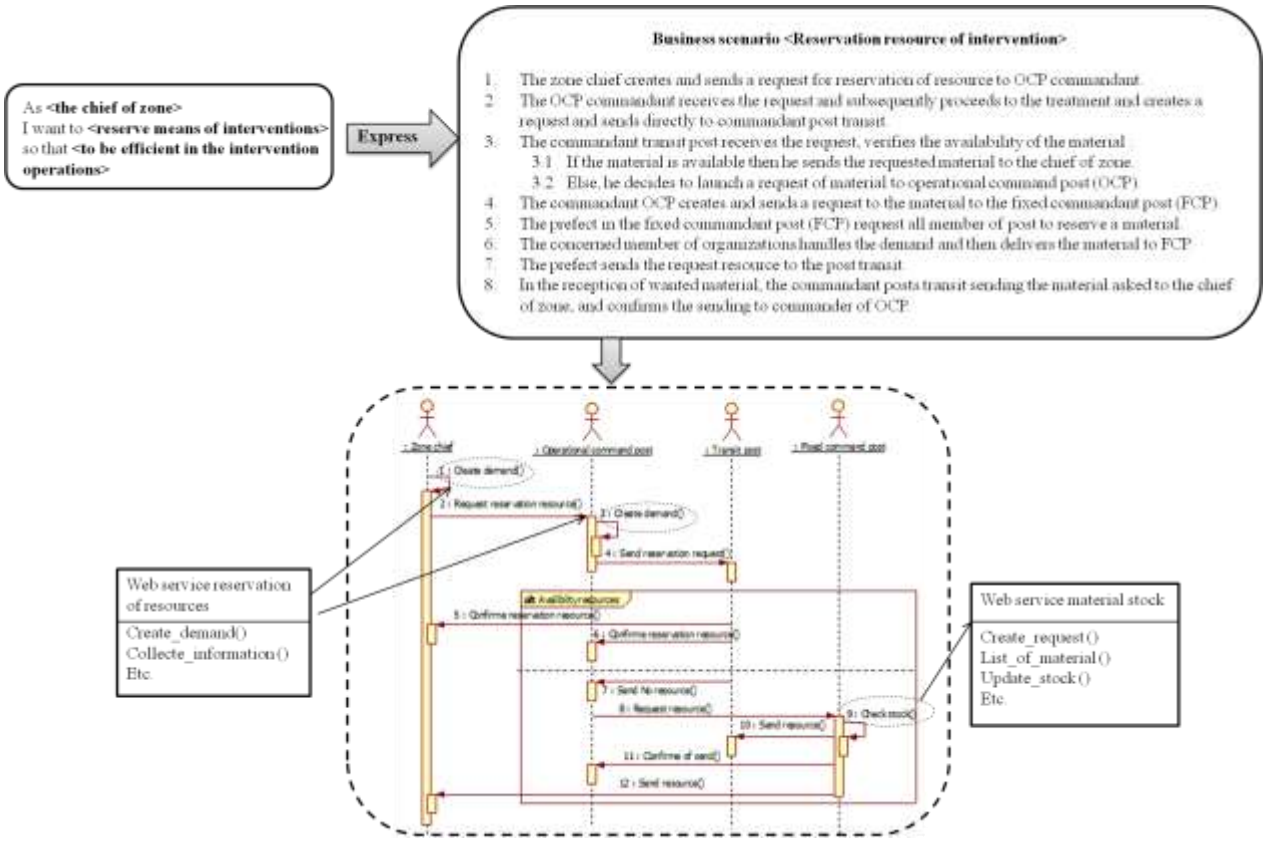


Fig. 11 Transformation Process: user story into business services “Reservation resources of intervention”

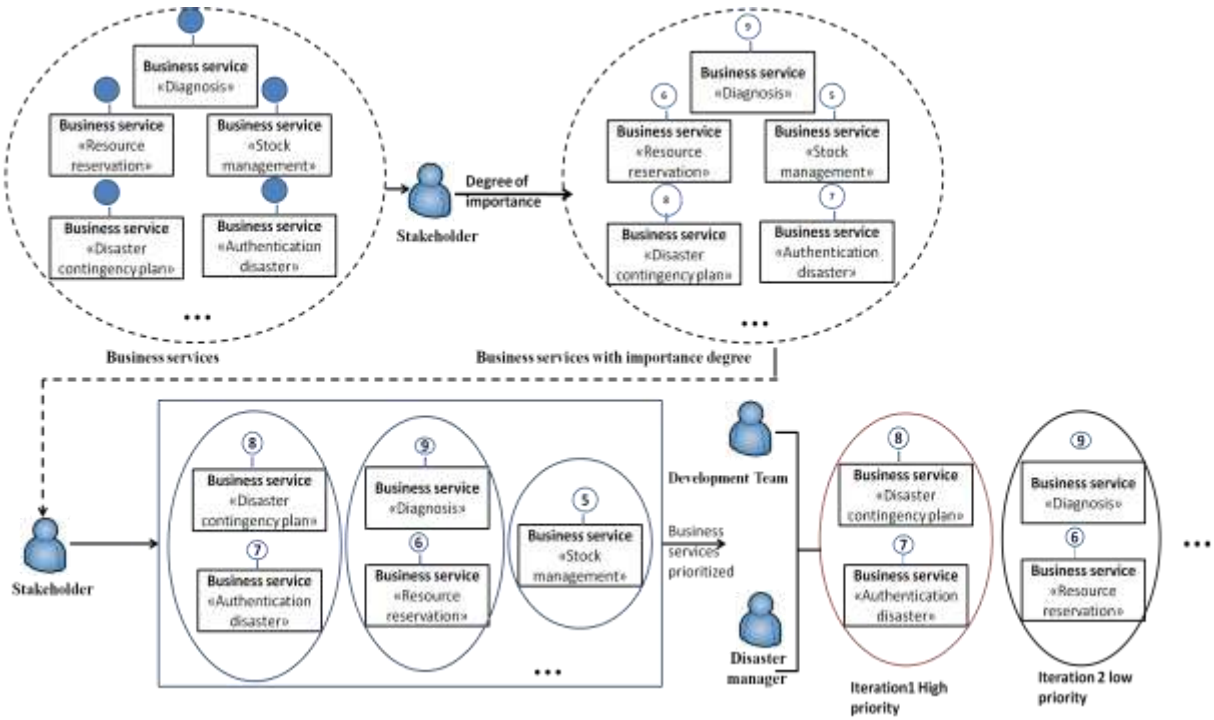


Fig.12 Example of prioritization of business service.

Phase.4 Iterations to Release

This phase is realized by development team (Business Designer, Developer/s and Architects teams). It consists to include several iterations of the management disaster system before the first release. In this phase, we find four essential steps.

Step 4.1 Design collaborative services

In this step, we proceed to design the service-oriented architecture components of the management disaster system. This step consists to transform the models developed during business analysis phase (domain decomposition, users analysis, etc.) described in UML diagrams (sequence diagrams) into a set of conceptual models describing the sequence of activities, business rules, etc.

For that purpose, we specify business processes identified in the previous phase by the use of the BPMN (Business Process Modeling Notation) formalism.

Fig. 13 illustrates BPMN model of the business process "Alert processing". This model highlights a set of elements such as: user tasks (e.g. diagnosis analysis), internal operations process (invocation of a service partner, returning results, etc.), connections, that allow to define abstractions and choose the granularity of the information.

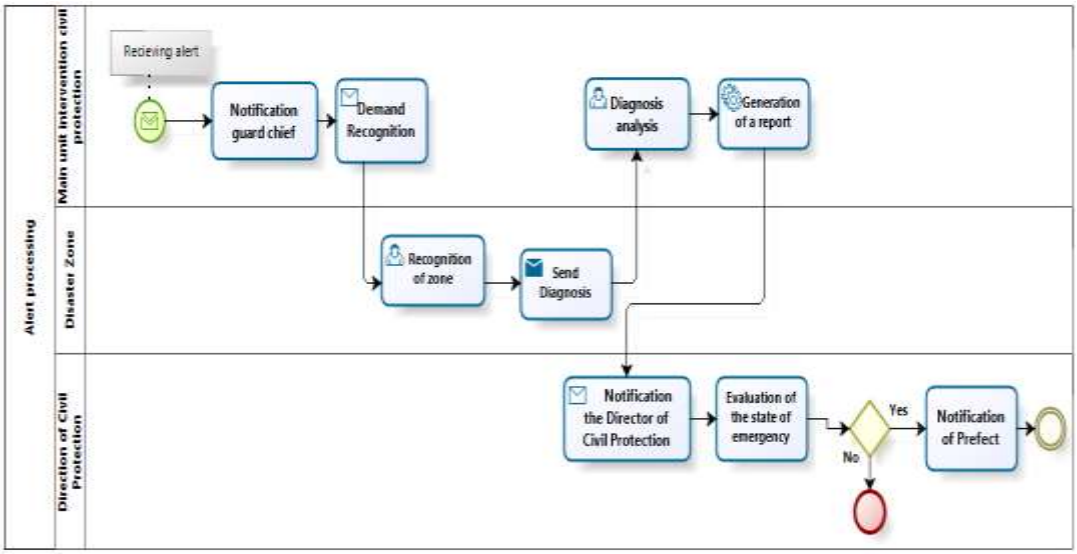


Fig. 13 Example of modelling of the “Alert Processing” process.

Step 4.2 Code collaborative services

In this phase, the developer uses several best practices of agile development such as: coding standards, code ownership, continuous integration, continuous testing and refactoring, etc. Coding process needs continuous testing and refactoring. The continuous integration practice is very important in the process of orchestration and choreography of new services in disaster management. Moreover, the code refactoring technique is required in order (1) to restructure a code without changing the functionality of the program and (2) to add flexibility in the system and communication improvement.

In this step, the implementation is based on web technologies and languages such as J2EE, JavaServer Pages (JSP), HTML, Javasript, and XML. As a result of this phase, the development team obtains a prototype implementing the main earthquake crisis management services previously identified.

Step 4.3 Test collaborative services

The aim of this step is to make sure that the users are able to use services successfully. This test will highlight the errors in the code. For this purpose, the developers use unit tests. This process aids the programmers to understand all the coding problems by evaluating some pieces (file, program, module, component, etc.) in isolation.

Testing early the disaster management system reduces risks such as schedule delays or cost overruns due to incomplete or unacceptable components.

Step 4.4 Evaluations by disaster managers

In this step, we proceed to evaluate the disaster management system by disaster managers.

Based on the UCD approach adapted from ISO 13407 (ISO, 2010), which means end-users are actively involved in the usability evaluations that are performed throughout the whole project lifecycle to ensure an interactive and easy-to-use system.

For this, we use several evaluation techniques, particularly interviews and thinking aloud method (Baccino et al., 2002; Lewis, 1982), in order to study how the managers use this system, the problems encountered and their opinion concerning the system.

The usability testing (Hurtado *et al.*, 2015) was performed by eight disaster managers during preparatory training. They are all male and below of age 30. Among disaster managers, 2 were much experienced, 2 were less experienced, and the remaining four was inexperienced in using disaster management system. We have used video recordings that are particularly suitable in order to collect more information. Thus, each user performed several tasks with the different disaster management system components. Table 1 shows the results of this test.

Table 1 Disaster management prototype testing with disaster manager

	Fully Agree	Partially Agree	Partially Disagree	Disagree
Components system are easy to use	6	2	X	X
Components run without any interruptions	2	6	X	X
Screen text is easy to read	6	2	X	X
Disaster management system supports in performing tasks in the case of an emergency	3	5	X	X

More, useful comments and feedbacks were received from disaster managers as recommendations for improvement of the user interface. The main recommendations are the following:

- Give more description of the commands with icons;
- Add predefined inputs in the list of choices (e.g., drop-down lists);
- Improve toolbars with frequently used functions by icons, help guides (e.g., tool-tips);
- Use meaningful colors of the real situation (e.g., red for emergency, green for emergency exit and so on);
- Give a capture and share of the actual emergency situation picture.

Through these recommendations of disaster managers, we have improved the usability level of disaster management system user interface. Fig. 14 shows the user interfaces relating to resource reservation. As a consequence, the disaster managers are satisfied by these improvements.

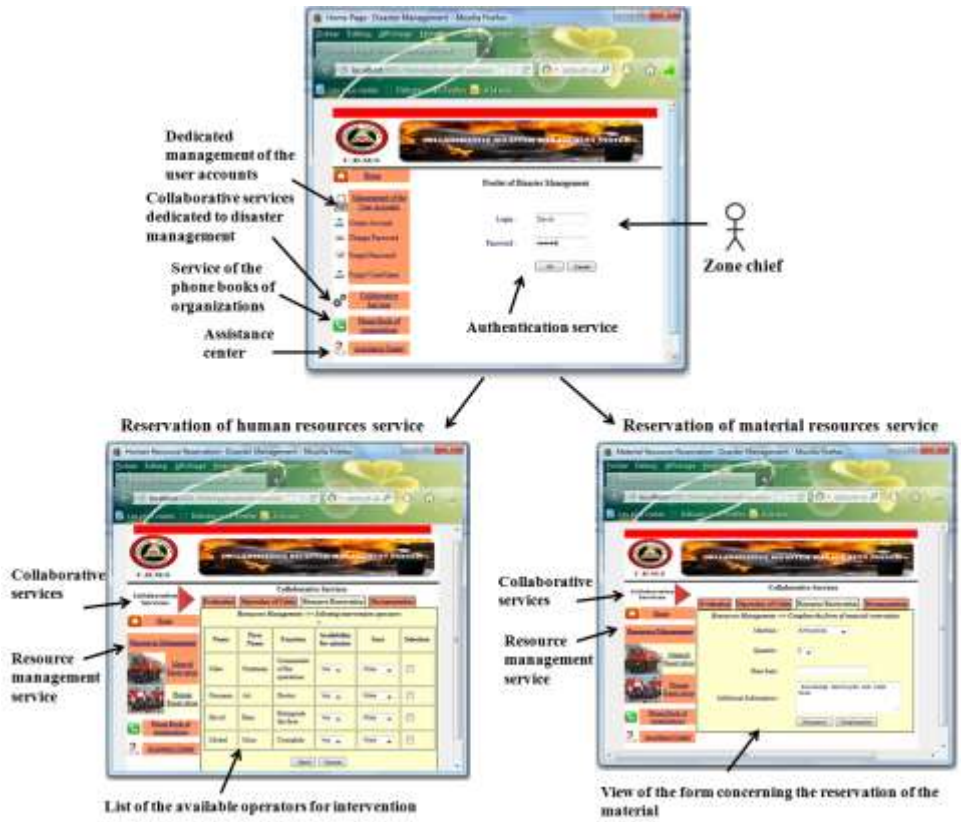


Fig. 14 Screen shot of the user interface prototypes relating to reservation resource services

PERFORMANCE EVALUATION OF THE AGILE-UCD-SOA FRAMEWORK

In this section, we propose an empirical evaluation of the proposed framework, which is applied to disaster management case study. This evaluation concerns the methodological aspects of Agile-UCD-SOA framework in terms of Human–computer interaction (user-centered approach) and Software Engineering (agile approach). Recall that the main objective of the proposed framework is to design and develop user-centered services.

The evaluation of Agile-UCD-SOA framework is based on four dimensions: (1) Human–computer interaction (user involvement, user analysis, task analysis, and user evaluation), (2) the iterative aspect of development, (3) agility (prioritization, culture of collaboration), and (4) the service oriented architecture (Business analysis, business process modeling, and legacy system analysis). These are the dimensions on which was based the proposal of Agile-UCD-SOA framework. The conclusions and the results of this evaluation are presented in Table 2.

The evaluation result through the case study shows the global degree of validation of the criterion (0 = not validated; 1 = partially validated; 2 = validated).

Table 2 Framework evaluation report

Criteria	Conclusions	Result
Human–computer interaction		
1. User involvement	The users (the members of civil protection) participated in a continuous way in the majority of the activities of the Agile-UCD-SOA framework. During the development of the interactive services, they gave their remarks and wishes regardless of the development stage.	2
2. User analysis	The users of interactive services are the members of civil protection. Their behavior as decision makers was precisely described. The most appropriate intervention strategies and / or tactics during a crisis have been well defined from the beginning of the business process development 'response to the crisis'.	2
3. Task analysis	The tasks of end users (member of civil protection) were analyzed with CTT (Concur Task Trees) , which is one of the most known task models in HCI (Human –Computer Interaction). CTT provides a graphical notation and ease of understanding by end users.	2
4. User Evaluation	The evaluations were planned at the end of each iteration. They allowed to validate the developed mockups and prototypes.	2
Iterativity and prototyping		
5. Iterativity and prototyping	At the end of each iteration, we obtained a prototype tested by developers and members of civil protection. This iterative character of the framework allowed to refine gradually the specifications, the Human-Computer Interaction, to take into account needs for the end users, then to integrate the modifications until the obtaining of the final prototype considered globally satisfactory by the users.	2
Agility		
6. Prioritization	The prioritization was realized in collaboration between end users and the development team. It aimed to take into account the priority needs of the users' point of view. Therefore, members of civil protection were main decision-makers during development, which is in the spirit of the agile methods.	2
7. Culture of collaboration	During all the activities related to the framework, the end users (members of civil protection) were present, motivated and involved in the activities (as well under the angle of the design as the evaluation).	2
Service oriented architecture (SOA)		
8. Business Analysis	The business analyst began with an analysis of the business domain. Thus, a decomposition of the business domain was performed (resulting in a	2

	"business model"). At the same time, the designers and the end users have analyzed and defined the user needs. A set of business services was able to be identified later.	
9. Business process modeling	The members of the Civil Protection and designers have modeled together all the different business processes with BPMN formalism (with the help of the analyst).	2
10. Legacy system analysis	An analysis of existing systems was conducted during the framework activities. This activity allowed to take into account several existing systems operated by the civil protection department of Béjaia city (Algeria). However, some systems of this organization were not able to be analyzed, the authorization not having been able to be obtained (by security measure).	1

The overall assessment of Agile-UCD-SOA framework (see Table 2) shows that: nine out of 10 criteria have been validated (corresponding to 90% of all criteria), a criterion was only partially validated (10%). Thanks in large part to the availability and the strong involvement of actors, this assessment have yielded very satisfactory results (Fig.15). This will confirm in other case studies.

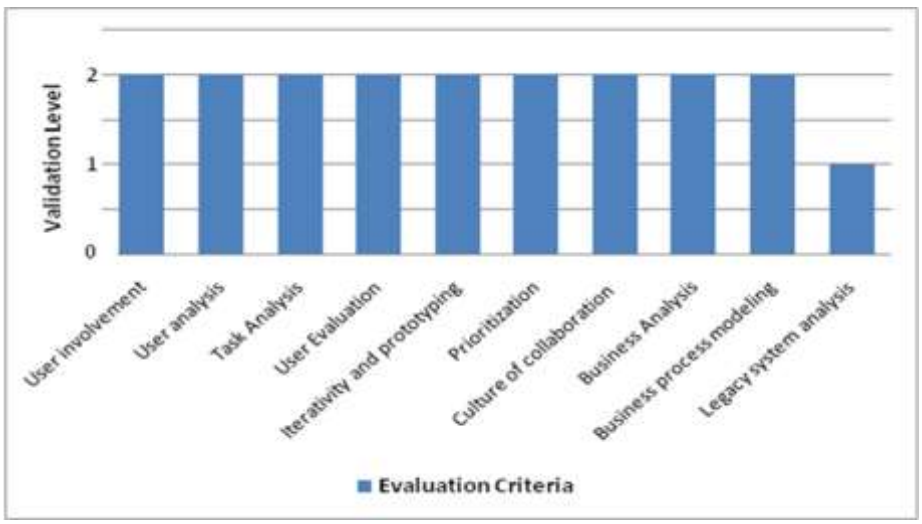


Fig.15 overall evaluation of the Agile-UCD-SOA framework

DISCUSSION

In this work, we propose a novel Framework Agile-UCD-SOA combining a set of agile characteristics and user-centered design principles in order to develop interactive, collaborative and usable services based on SOA paradigm.

We showed that the Framework Agile-UCD-SOA is a process that helps the developer team to get requirements that provide such services for disaster management.

The most motivation of this work is to propose a framework where the end users (disaster managers in civil protection) are the principal actors all along the development process. In addition, to allow delivering usable and interactive services in short delay. These services are supported by service oriented architecture (SOA).

In the literature, several studies for the development of disaster mangement systems, such as (Affeltranger *et al.*, 2007; Nawaz and Zualkernan, 2009; Wood *et al.*, 2013; etc.) have been reported. Nevertheless, none of these works has considered the aspect of changing user needs that may arise during the design project of disaster management

system. We believe that the concept of agility in (Wood *et al*., 2013) is considered as disturbance that can happen to a development team during a crisis such as a terrorist attack, earthquake, etc. Our work addresses the agility according to the changing needs of the end user and how we can plan and organize these requirements. In addition, we note that these studies also use either user-centered design or agile concept or service oriented architecture in their development of disaster management system but never the three in one unified approach.

Our proposed framework should be a model that guides the developer team to build up interactive services, while respecting in the one hand usability, interactivity and iteratively and on the other hand respecting acceptability of changing business needs of users. Often, the developer team asks about how to develop these services or product, and when can be delivered the product for the end users. But unfortunately, they do not think about how to ease use for the end users. So, the usability of the services is often not enough taken into by the development teams. For this, our framework incites the development team to build up user-centered disaster management system in which end users' needs are considered early in the design process of services.

In addition, the multiplicity of end user profiles confirms that is very important to tailor user-centered design because the end users by their nature cannot express all their needs at once, and indeed they have tendency to change them every time. For this, the development team and the end users must collaborate and work together in order to: (1) accept any changes of their needs (new user needs) and integrate them as soon as possible in the development process, (2) implicate end users all along the process.

Moreover, our proposed framework adopts an agile process development. The aim of agile process is to encourage rapid and flexible responses to changes by emphasizing on user involvement and his/her feedback and delivery of several small releases. Our framework targets end user satisfaction rather than satisfying the contract negotiation.

In the literature, many agile methods do not cover all steps of process and lack of formality. In our proposed we follow with agile process with give all step of process. In addition, in the traditional project the prioritization is usually performed once and before the implementation phase, while in agile process is an ongoing process, performed in the beginning of each iteration, or even during the iteration; this reflects the dynamics of the development process during the project.

To make this choice we had to perform a comparative study between user-centered design, agile approaches and service oriented architecture (Ait Abdelouhab et al., 2014). This comparison allows us to note that no model combining the advantage of those approaches exists in the literature.

Our purpose of interaction design is to provide an interactive service that is, from a user perspective, easy to use, more effective and more usable.

As a practical implication, we can note the development of interactive, collaborative services using our Agile-UCD-SOA framework in a real context which is the disaster management. During this project we were able to involve disaster managers throughout the development of interactive, collaborative services. Indeed, as we have detailed in different phases of our framework, disaster managers were involved from the beginning to the end of the development process. The interactive and collaborative services are the final product of our application. These services may now be used by disaster managers in case of emergency situations.

CONCLUSION

Disaster management can raise new challenges for the HCI design. Indeed, the disaster management is a cooperative activity; so the organization, the coordination and the communication dimensions must be taken into account in the representation of this type of situation. In this context, heterogeneous actors belonging to different organizations need to collaborate and work together with the shared aim to solve, or at least reduce, the crisis situation. Each actor is often equipped with different devices and communication technologies, and should carry on specific tasks. In addition, the multiplicity and diversity of actors involved the volume and the heterogeneity of information sources, the critical dependencies between actions and the dynamics of the situation make the management more complex.

In this paper, we have proposed a novel Agile-UCD-SOA framework for design disaster management system. For this, we have combined an agile process, user-centered approach and the service oriented paradigm for the development of interactive and collaborative services for disaster management. The Agile-UCD-SOA based framework described is aimed at incorporating the users' perspectives in the service-oriented development with an agile process. The three approaches mainly differ in their perspectives on system design. We believe that an

integration of agile process, user-centered design approach with service-oriented software design approach is an important step for the development of interactive services to be accepted by end users in such complex organizations.

Our Agile-UCD-SOA framework focuses on UCD principles and adapts techniques to get the requirements and feedbacks from real end users. This approach continuously involves disaster managers (end users) for getting their requirements and needs. Thus, the definition of user requirements is the first step that the development team has to perform in order to provide interactive and collaborative services. Once identified, user requirements effectively lay the foundation for development team and testers to begin determining the functionalities, responsiveness and interoperability required by the system. This framework allows the gathering and validation of the requirements incrementally.

In addition, our Agile-UCD-SOA framework is based on agile process. The major importance of these approaches is that the work is organized in a series of iterations in which the user goals to be dealt with are developed. Moreover, it concentrates only on the functions needed immediately, delivering them fast, collecting feedback and reacting rapidly to business and technology changes.

Therefore we have tailored the SOA life cycle to identify and design relevant services of disaster management. SOA is an approach for software design which breaks everything down into agile services dealing with one specific need. The interest of the service-oriented architecture (SOA) is to allow the implementation of sequences of tasks performing a business process. In addition, it provides a methodology to organize and utilize resources of the computer system in a distributed manner. SOA also encourage the reuse of services.

A major benefit of Agile-UCD-SOA framework is that it leads to highly flexible and agile software that should be able to meet rapidly changing business needs. We applied this approach to an Earthquake management case study relative to the Civil Protection Department of Béjaïa (Algeria).

Further work and research perspectives need to be considered. Some research perspectives may concern the following issues. First we plan to propose a methodology for validation of Agile-UCD-SOA Framework by taking into account evaluation methods and techniques. We envisage also proposing a prioritization methodology for agile-UCD-SOA framework, taking as our starting point the prioritization criteria, methods and techniques used in the software engineering field. We hope to undertake a comparative study between our solution (Agile-UCD-SOA Framework) and the related work available in the literature while integrating the disaster managers of the Civil Protection of Béjaia. Finally, we envisage studying the main organizations (e.g. health services, order public service, information service) involved in the process of resolving the crisis and provide other interactive services.

Acknowledgment

The authors gratefully acknowledge and express their warm thanks to the Direction Générale de la Protection Civile de la wilaya de Béjaia, and the University A. Mira of Bejaia, Algeria.

References

Abrahamsson P., Salo O., Ronkainen J., and Warsta J. (2002) 'Agile software development methods: Review and Analysis', Espoo, Finland, Technical Research Centre of Finland, VTT Publications 478, (Accessible at: http://www.inf.vtt.fi/pdf/publications/2002/P478.pdf).

Affeltranger, B., Berger, E., Peinel, G., Rose, T. and R.A. Mlesnita (2007) 'IT for risk and emergency management: Consolidating methodologies for user -centered design', In *Proceedings of the 6th international symposium on environmental software systems*.

Agile Alliance. (2001) 'Manifesto for Agile Software Development', Technical report. (Accessible at: http://www.agilealliance.org).

Ait abdelouhab K, Idoughi D and Kolski C (2014) 'Agile and user centric SOA based service design framework applied in disaster management' ICT-DM'2014, 1st IEEE International Conference on Information and Communication Technologies for Disaster Management (March 24-25, 2014), 1-8, Algiers, Algeria.

Aoyama, M. (2007) 'Persona-scenario-goal methodology for user-centered requirements engineering', In Requirements Engineering Conference, RE'07. 15th IEEE International. pp. 185-194, IEEE.

Arsanjani A., Booch G., Boubez T., Brown P.C., Chappell D., Vadoss J., Erl T., Josuttis N., Krafzig D., Little M., Loesgen B., Manes A.T., McKendrick J., Talbot S. R., Tilkov S., Utschig-Utschig C., and Wilhelmsen H. (2009) 'The SOA manifesto,' SOA Manifesto, October, (Accessible at: http://www.soa-manifesto.org).

Arsanjani, A. (2005) 'SOMA : Service-oriented modeling and architecture. IBM Corporation.

Baccino, T., Bellino, C., and Colombi, T. (2005) 'Mesure de l'Utilisabilité des Interfaces,' Paris : Hermes.

Beck, K. (2000) Extreme Programming Explained, Addison-Wesley, Pearson Education.

Benssam, A., Nouali-Taboudjemat, N., and Nouali, O. (2013) ' Towards in IT-based platform for disaster risks management in Algeria', In Proceedings of the 10th international ISCRAM conference: Baden-Baden, Germany , pp. 72-77.

Beyer, H. and Holtzblatt, K., (1998) 'Contextual Design: Defining Customer-Centered Systems', Morgan Kaufmann Publishers, Inc, San Francisco, CA, USA.

Blomkvist, S., (2005) 'Towards a Model for Bridging Agile: Development and User-Centered Design', In: Human-Centered Software Engineering—Integrating Usability in the Software Development Lifecycle, Springer Netherlands, 2005, pp. 219-244.

Borges M., (2014) 'ICTs for Collaboration in Emergency Management Systems', PowerPoint Prresented at the ICT-DM'2014, 1st IEEE International Conference on Information and Communication Technologies for Disaster Management (March 24-25, 2014), Algiers, Algeria, unpublished.

Braune, S., Brucker, A. D., Kleser, G., Li, K., Meijler, T. D., Paulheim, H. and Probst, F. (2011) 'A Service-oriented Architecture for Emergency Management Systems," In Software Engineering (Workshops), pp. 225-232.

Carroll, J.M. (2000) Making Use: Scenario Based Design of Human–Computer Interactions. MIT Press.

Cockburn, A., (1999) 'A Methodology Per Project. [online] http://alistair.cockburn.us/crystal/articles/mpp/methodologyperproject.html.

Cohn M. (2004) 'User stories applied: For agile software development', Addison-Wesley Professional.

Cooper, A., Reimann, R., and Cronin, D., (2007) 'About Face 3.0: The Essentials Of Interaction Design,' Wiley publishing, Indianapolis.

Coppola, D. P. (2006) 'Introduction to international disaster management', Butterworth-Heinemann.

Erl T. (2009) ' SOA: Principles of service Design', Vol. 1, Prentice Hall.

Erl, T. (2005) 'Service-Oriented Architecture: Concepts, Technology, and Design', Prentice Hall.Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J., and Cajander, Å., (2003) 'Key principles for user-centred systems design', Behaviour and Information Technology, Vol. 22 No 6, pp. 397-409.

Hanachi, C., Charoy, F., and Stinckwich, S. (2012) 'Introduction to the Collaborative Technology for Coordinating Crisis Management', IEEE 21st International Workshop On Enabling Technologies: Infrastructure For Collaborative Enterprises (WETICE), pp. 349-351,

Haubrock, S., Theisselmann, F. and Dransch, D. (2007) 'Service-Oriented Architectures for Natural Disaster Management', In Proceedings of the Joint CIG/ISPRS Conference on Geomatics Solutions for Disaster Management, CIG/ISPRS Joint Conference, Toronto, Conference CD.

Humayoun, S. R., Catarci, T., de Leoni, M., Marrella, A., Mecella, M., Bortenschlager, M. and Steinmann, R. (2009) 'Designing mobile systems in highly dynamic scenarios: The workpad methodology', Knowledge, Technology & Policy, Vol. 22, No. 1, pp. 25-43.

Hurtado N., Ruiz M., Orta E. and Torres J. (2015) 'Using simulation to aid decision making in managing the usability evaluation process', Information and Software Technology, Vol (57),pp. 509–526.

Idoughi, D. (2008) Contribution à un cadre de spécification et conception d'IHM de supervision à base de services web dans les systèmes industriels complexes: Application à une raffinerie de sucre (in french). Unpublished PhD thesis, University A. Mira of Bejaia, Algeria and University of Valenciennes and Hainaut-cambrésis, France.

Idoughi, D., Kerkar, M., and Kolski, C. (2010) 'Towards new web services based supervisory systems in complex industrial organizations: basic principles and case study. Computers in Industry, Vol.61, No.3, pp. 235-249.

Idoughi, D., Seffah, A., and Kolski, C. (2012) 'Adding user experience into the interactive service design loop: a persona-based approach', Behaviour & Information Technology, Vol. 31, No.3, pp. 287-303.

International Organization for Standardization (2010) DIS, ISO 9241-210: Ergonomics of human system interaction- Part 210: Human-centred design for interactive systems, Switzerland, ISO.

Janssen M. , Lee J., Bharosa N., and Cresswell A. (2010) 'Advances in multi-agency disaster management: Key elements in disaster research. Information Systems Frontiers', Vol. 12, No.1, pp.1-7.

Jiugang, L., Xinming ,T., Zhengjun, L. and Minyan, D. (2009) 'Design and implementation of WebGIS for government emergency management based on SOA', International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences, Vol. 28, No.7.

Lewis, C. H. (1982) 'Using the Thinking Aloud Method In Cognitive Interface Design', Technical Report IBM RC-9265.

Liu M., Wang H J. and Leon Zhao J. (2012) 'Technology flexibility as enabler of robust application development in community source: The case of Kuali and Sakai', Journal of Systems and Software, Vol (85), No.12, pp. 2921-2928.

Loup-Escande E., Burkhardt J M., Christmann O. and Richir S (2014) 'Needs' elaboration between users, designers and project leaders: Analysis of a design process of a virtual reality-based software', Information and Software Technology, Vol (56), No.8, pp.1049-1061.

Mahmood Z. (2007) 'The promise and limitations of service oriented architecture', International Journal of Computers, vol. 1, No 3, pp. 74-78.

Maiers C., Reynolds M., and Haselkorn M. (2005) 'Challenges to effective information and communication systems in humanitarian relief organizations', In Professional Communication Conference, 2005. IPCC 2005. Proceedings. International, IEEE, pp. 82-91.

Markus, M. L., Fedorowicz, J., Bodeau, D., and Brooks, J. (2009) 'Characterizing and Improving Collaboration and Information-Sharing Across Emergency Preparedness and Response Communities', In Proceedings of ICEG 2009: The 5th International Conference on e-Government, Suffolk University, Boston, USA.

Millard, D. E., Howard, Y., Abbas, N., Davis, H. C., Gilbert, L., Wills, G. B., and Walters, R. J. (2009) 'Pragmatic web service design: An agile approach with the service responsibility and interaction design method', Computer Science-Research and Development, Vol.24 No 4, pp.173-184.

Mushtaq, Z., and Qureshi, M. R. J. (2012) 'Novel Hybrid Model: Integrating Scrum and XP," International Journal of Information Technology and Computer Science (IJITCS), Vol. 6, pp. 39-44.

Nawaz, A. I. and Zualkernan, I. A., (2009) 'The role of agile practices in disaster management and recovery: a case study', In Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research, pp. 164-173, IBM Corp.

Papazoglou, M. P. and van den Heuvel, W. J. (2006) 'Service-oriented design and development methodology', International Journal of Web Engineering and Technology (IJWET), Vol. 2 No 4, pp. 412-442.

Rivero J M., Grigera J., Rossi G., Robles Luna E., Montero F. and Gaedke M. (2014) 'Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering', Information and Software Technology, Vol (56), No.6. pp. 670-687.

Robey, D., Welke, R., and Turk, D. (2001) 'Traditional, iterative, and component-based development: a social analysis of software development paradigms', Information Technology and Management, Vol. 2, No. 1, pp. 53–70.

Rumbaugh, J., Jacobson, I., and Booch, G. (1999) The unified modelling language reference manual. Addison-Wesley. Boston, MA, USA.

Shahrbanoo, M., Ali, M., and Mehran, M. (2012) 'An Approach for Agile SOA Development Using Agile Principals," arXiv preprint arXiv:1204.0368.

Te'eni, D., Carey, J., and Zhang, P. (2007) ' Human–Computer Interaction: Developing Effective Organizational Information Systems John Wiley & Sons, Inc, Hoboken, NJ, USA.

Wang, M., Wang, H., Xu, D., Wan, K. K., and Vogel, D. (2004) 'A Web-service agent-based decision support system for securities exception management', Expert Systems with Applications, Vol. 27, No.3, pp.439–450.

Waugh, W. L., and Streib, G. (2006) 'Collaboration and leadership for effective emergency management', Public Administration Review 66.s1, pp. 131-140.

Wood, L., Büscher, M., Van Veelen, B., and Van Splunter, S. (2013) 'Agile response and collaborative agile workflows', International Journal of Information Systems for Crisis Response and Management (IJISCRAM), Vol.5, No.3, pp.1-19.

Yu X and Petter S (2014) Understanding agile software development practices using shared mental models theory. Information and Software Technology, Vol (56), No.8, pp. 911-921.

Zhang, P., Carey, J., Te'eni, D and Tremaine, M. (2005) 'Incorporating Human-Computer Interaction Development into SDLC: A methodology', Communications of the AIS, Vol. 15, pp. 512-543.

Zimmerman O., Korgdahl P., and Gee C. (2004) 'Elements of Service-Oriented Analysis and Design', IBM Corporation (Accessible at: www-106.ibm.com).