



HAL
open science

Agile and user centric SOA based service design framework applied in disaster management

Karima Ait Abdelouhab, Djilali Idoughi, Christophe Kolski

► To cite this version:

Karima Ait Abdelouhab, Djilali Idoughi, Christophe Kolski. 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, Institute of Electrical and Electronics Engineers (IEEE), Mar 2014, Alger, Algeria. 10.1109/ICT-DM.2014.6917792 . hal-03383975

HAL Id: hal-03383975

<https://uphf.hal.science/hal-03383975v1>

Submitted on 19 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Agile & user centric SOA based service design framework applied in disaster management

Karima Ait Abdelouhab , Djilali Idoughi
Laboratory of Applied Mathematics- LMA
Univ. Abderrahmane Mira of Bejaia
Bejaia, 06000, Algeria
{aitabdelouhab,djilali.idoughi}@gmail.com

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

Abstract— 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 to the design, implementation and deployment of SOA based interactive services. This approach is applied to a typical disaster management case study to demonstrate its feasibility.

Keywords— *User centered design, agile methods, SOA, Service design, Disaster management.*

I. INTRODUCTION

The disaster management is a special type of human complex organization in which the communication and collaboration between several types of actors become major management issues. Disaster management implies frequent face to face communication and needs a core team focusing on finding solutions and delivering them to whom it may concern. New types of Information and Communication Technologies have to be progressively envisaged, analyzed, designed and evaluated.

Many alternatives are possible for designing new disaster management systems. For this purpose, Agile Methods (AMs) and techniques [1], [2] are necessary, because they allow to open communication between designers and end users, define clear requirements and produce results quickly with reduced costs and risks of failure or delays.

Disaster management is characterized by the multiplicity and diversity of actors involved. So it is important to adopt a User Centered Design (UCD) [3], [4] as an approach for better identifying the different types of users who would play a major role in a major crisis. The UCD approaches also help to understand the users' information needs and to design useful and usable interactive systems based services.

Moreover the crisis management systems [5] are heterogeneous, and then it is important to use an underlying architecture that will support important features, such as heterogeneity and interoperability. For this, the service oriented architecture (SOA) [6], [7] aims to improve interoperability [8] between the various services involved and to facilitate interfacing with specific technical and scientific monitoring systems for any kind of event and give the possibility to reuse services. This architecture is based on Web technologies [9].

However, due to the fact that a majority of software engineering development processes focus on software architecture, a satisfactory integration has not been achieved yet. Therefore, in this research we have focused on integrating these three global approaches, so as to take into account deeply the user needs in an iterative development, 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 services.

In this paper, we propose a novel approach for complex software development that combines agile characteristics and principles, user-centered techniques and SOA paradigm. This approach is then applied to a typical disaster management case study.

The section II gives an overview of the state of the art in agile methods, UCD and SOA as well as global comparative studies between them. In section III, we highlight the main phases of the proposed design and evaluation framework incorporating UCD principles in an agile process while relying on service-oriented architecture. This framework is applied to a typical disaster management. In Section IV, we draw some conclusions and outline our future work.

II. BACKGROUND AND NEW MOTIVATIONS

This section presents some background on SOA based services, user centered design and agile methods, as well as comparative studies between them.

A. Service Oriented Architecture (SOA)

Many definitions have been given to SOA. One of these is given by Erl [10] “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 [11], 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. SOA allows enterprises and their IT

systems to be more agile to the changes in the business and the environment.

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 [12]. SOA describes a set of characteristics: loosely coupled, distributed, invocable, publishable and business oriented [32] that realizes the technical benefits of SOA [13].

Several SOA methodologies have been given in the literature such as: SOAD (Services-Oriented Development of Application) [8], IBM Service Oriented Modeling and Architecture (SOMA) [14], Thomas Erl's methodology [15], Papazoglou and Heuvel (methodology of development the Web services) [16] and the service-based user interface approach [17].

The Majority of SOA methodologies proposes 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. The first two phases are the most important ones because the success of SOA development mainly depends on them.

B. User-Centered Design

The 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 [3]. The international standard ISO/DIS 13407 (Human Centered Design Process for Interactive Systems) [4] defines User Centered Design (UCD) as an approach to software and hardware design that identifies four different basic principles: (1) an appropriate allocation of function between the user and the system, (2) an active involvement of users, (3) iterations of design solutions and (4) multidisciplinary design teams [19].

Many UCD methods have been proposed in the literature with the aims to make end users and their experiences a focal point of design process [20]: such as Goal Directed Interaction Design (GDID) [21], Contextual Design (CD) [22], Scenario-Based Design [23], Participatory design [24], the Human-Centered Systems Development Life Cycle (HCS DLC) model [25], [26], Persona-based approach [27], 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. One important aspect of UCD is the collaboration between users and developers to build software solutions, so that each group brings its own experience [28].

C. Agile development process

The Agile manifesto [1] 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 [2]. These methodologies focus on people, communication and the ability to adapt to change rather than the process, tools and predictive planning [29].

The main agile methods are: **XP** (Extreme Programming, www.extremeprogramming.org), **Scrum** (www.controlechaos.com), **DSDM** (Dynamic Software Development Method (www.dsdm.org), **AM** (Agile Modeling, www.agilemodeling.com), **ASD** (Adaptive Software Development, www.adaptived.com), **Crystal** series (www.crystalmethodologies.org), **FDD** (Feature Driven Development (www.featuredrivendevelopment.com)).

Hereafter, we outline two comparative studies involving the three approaches defined previously.

D. Comparative studies

- *Comparative study between Agile Methods and SOA*

One major importance of the service oriented paradigm as well as agile approaches is the embracing 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 [25].

SOA as an architectural style for developing and integrating enterprise applications, stresses that business must be able to respond to the market by building appropriate business services [30]. SOA is a development approach decomposing all in services (software components). It addresses a specific need within reusability, simplicity and interoperability aims, and can hide the heterogeneity of the underlying information system.

Both SOA and agile methods recognize that change is inevitable and that organizations need to effectively cope with those changes. But, they are fundamentally different, and cover different areas. SOA aims to make the whole agile business using services as building blocks for applications [31], unlike the agile software development aims to make agile organizations by introducing practices that increase communication and feedback.

Furthermore, SOA is a top-down approach, while the agile approach is a bottom-up system development methodology [31]. Moreover, SOA doesn't stress in user feedback and change of the services once they are built, whereas the agile approach focuses on frequent feedback at both a technical and personal level. Also, SOA encourages that architecture be upfront BDUF (Big Design Up Front) [31], while agile methods suggest LDUF (Little Design Up Front) [18].

Several approaches in the literature have been studied in order to support a dynamic and turbulent business environment. For instance Ivanyukovich [33] proposes a structured approach to analyzing software development methodologies in light of the specific features of service-oriented applications; Boesteanu [34] describes a more effective approach to adopting and implementing SOA within medium and large organizations by combining a lean approach to SOA strategy with an Agile approach to SOA projects, etc.

- *Comparative study between Agile Methods and UCD*

The collaboration between users and developers is an important aspect of UCD to build interactive software solutions, each one bringing their experience to bear [35]. UCD is a philosophy that tries to understand the users and their tasks [24].

Both User Centered Design (UCD) and agile software development are iterative approaches to software development, and they can increase the chances of delivering a successful project [36].

The difference between the two approaches is that UCD is a design process focusing on user research, user interface design and usability evaluation [37]. However, an agile process focuses on how to organize the required tasks to reach the overall goal of delivering working software. In addition, AMs focus on code development, while the UCD methods focus on the design of the interaction that users will engage in [35]. AMs place less emphasis on the process and its deliverables, and center instead on the people involved and their cooperation in order to produce results quickly with reduced risk of failure or delays [37]. Whereas, UCD is an approach that places the end user of an application in the center of each design phase in order to ensure that the end product will answer to the users' needs, and want. Furthermore, an agile development focuses on making coding more efficient, while UCD aims to produce systems or software that are highly usable and this involves using methods and techniques that are oriented towards usability [37].

Finally, both approaches seek to satisfy the users' needs. However, in AMs users are involved in checking that the functionality has been correctly implemented, while in UCD users give input regarding other aspects such as user satisfaction or efficiency of use for the whole application [35].

Due to a number of similarities between user-centered design (UCD) and agile development, several approaches have been proposed in the literature in order to integrate UCD and agile methods. For example, Beyer [38] shows how integrating the approach fills the gaps in agile methods for both fast-turnaround iterative projects as well as the large-scale, high-impact, enterprise projects, Sy [39] describes the process of integrating UCD with agile methods currently being successfully adopted by Autodesk, etc.

This comparative study shows that it is necessary to integrate these three different approaches to benefit from their specificities and advantages. Fig.1 illustrates the intersection between the three approaches.

As illustrated by the Fig.1, there is a need and much remains to be done 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 services.

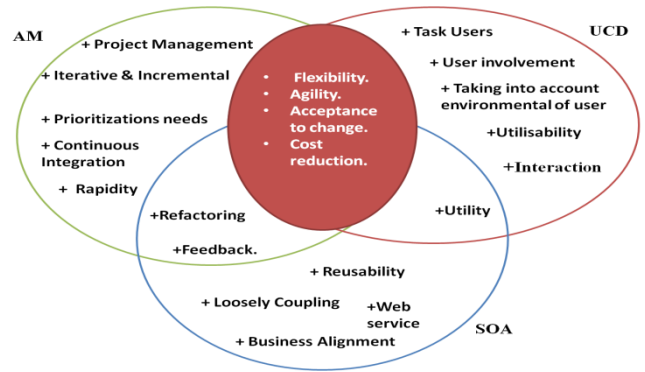


Fig.1. Combining UCD, Agile Methods and SOA concepts.

III. A NOVEL DESIGN FRAMEWORK COMBINING AGILE, UCD AND SOA APPROACHES

The objective of the design phase of the proposed framework is to transform the conceptual model into visual and design structures. The design propositions are iteratively refined towards mock-ups, animated page-screen-based and deployable interactive prototypes. Moreover, it starts with identifying user characteristics and needs. These are mapped onto system functionalities. Afterwards, the interaction dynamics are designed and usability issues are addressed. Last, the concrete look and feel of the system is designed, and a concrete prototype can be presented in the different categories of users.

Hereafter, we briefly describe the main design activities of the proposed Agile-UCD-SOA framework applied to an *earthquake management* case study.

The applied Agile-UCD-SOA framework has four main phases (Fig. 3): (1) Study of the organization and agile business analysis, (2) Just In Time (JIT) Requirements Analysis and Elicitation, (3) Iterations Prioritization and Planning, (4) Release to Iteration. It follows an agile life cycle while respecting contents of a SOA life cycle.

1. Study of the organization of disaster management and agile business analysis

This phase considers the preliminary study of the complex organization in order to identify: (1) all the stakeholder's 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 (Fig. 2).

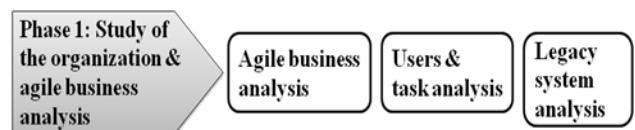


Fig.2. Phase 1 of the proposed approach.

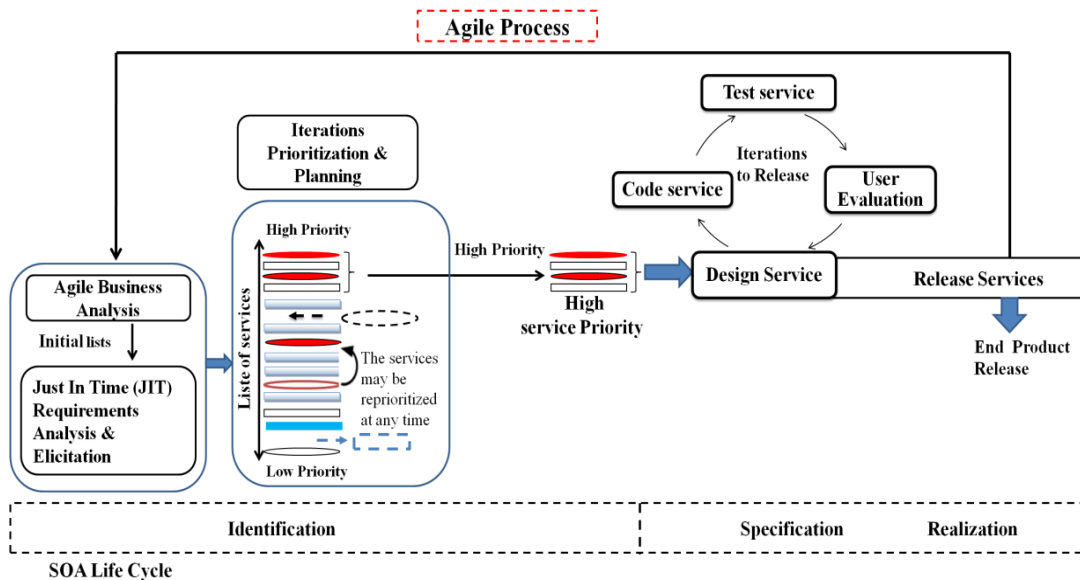


Fig.3. Global view of the Agile-UCD-SOA framework.

In this article, we focus our case study on the coordination and communication issues occurring within **The Direction of Civil Protection of Bejaia (Algeria)** as well as the decision-making of the superior authority. In fact, the disaster management service at the level of the disaster and emergency services depends on several internal services, besides; it is in connection with other external services (Fig.4).

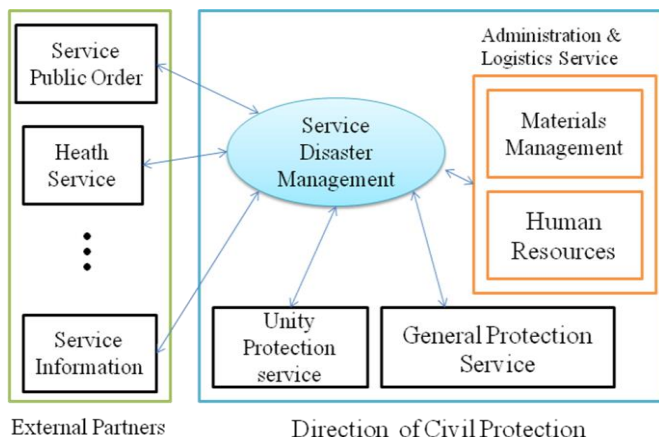


Fig.4. Disaster Management services and External Partners

This phase aims to elaborate an initial *service model* [40] which includes a first set of candidate services that can support disaster management services, processes and goals of the organization.

At the beginning, we deal with the hierarchical decomposition of the business domain of disaster management into functional areas give rise to business use cases.

Fig.6 shows a simplified view of the decomposition domain model (or functional mapping system). In addition, we have highlighted, in this model, business processes related to the resolution of disaster and areas that are directly related to these processes.

Fig. 7 shows a set of business use cases that are considered as a good candidate for the high level of disaster management business services.

- Users and task Analysis

In this step, we define and identify user profiles and their tasks. For this, it is important to collect different information about users. For that purpose, we use some adequate UCD techniques such as: field observations, interviews, etc.

Thereafter, we use a hierarchical model defining the types of users that participate in disaster management. In addition, a task analysis of these users 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 (Fig.5).

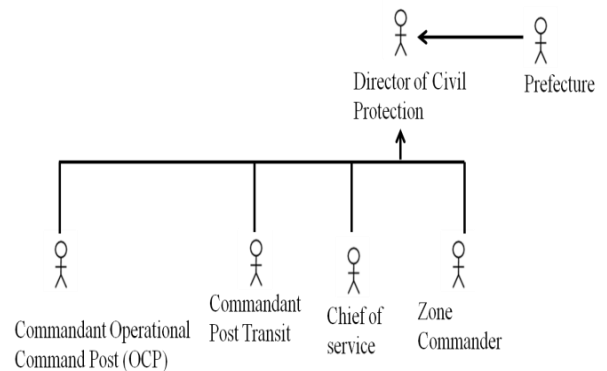


Fig.5. Typology of actors involved in the earthquake management.

- 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.

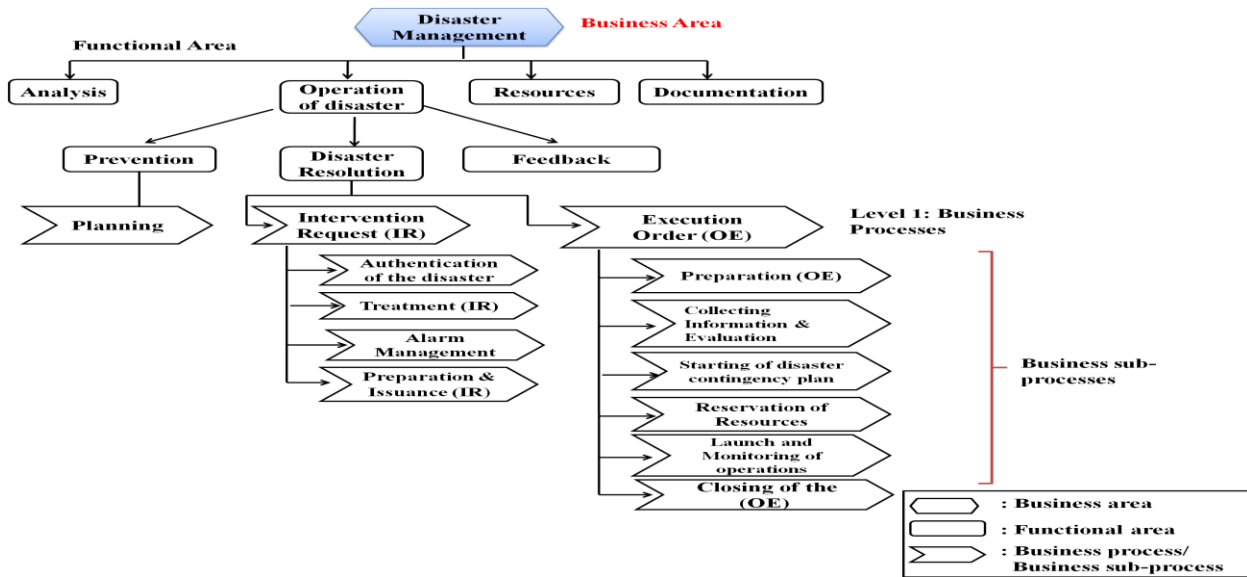


Fig.6. Decomposition of the business domain “Disaster Management”.

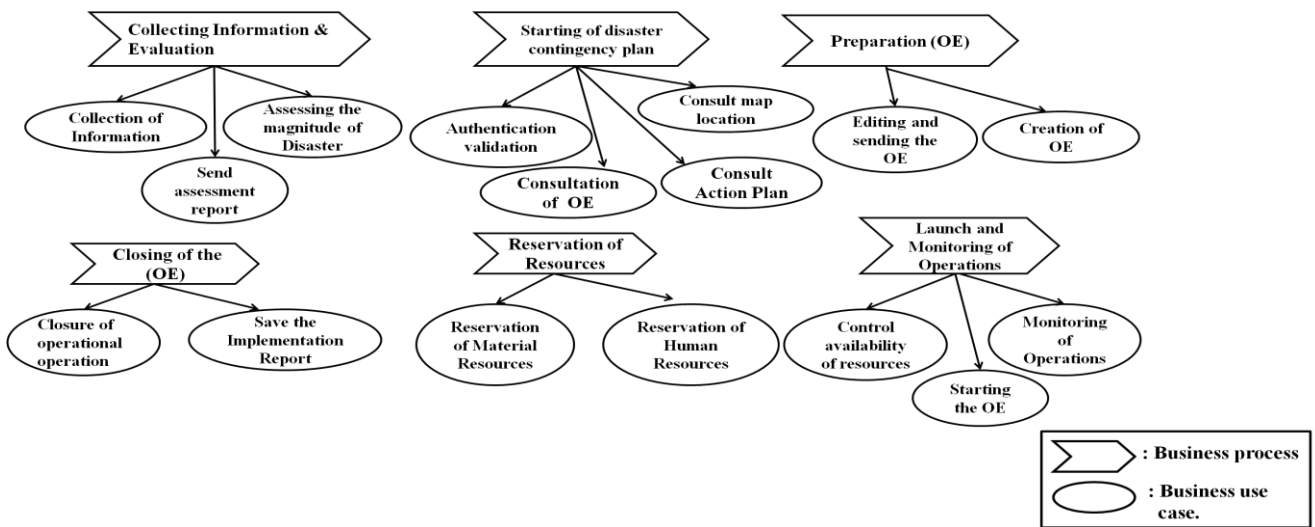


Fig.7. Example of candidate business services of Disaster management.

2. Just In Time Requirements Analysis and Elicitation

The aim of this phase is to capture analysis and define requirements Just-in-Time when they are needed. These requirements are identified and expressed in terms of user stories. Therefore, two steps are defined as follows (Fig.8):



Fig.8. Phase 2 of the proposed approach.

- Identifying and creating user stories

The *User Stories* [41] are a very effective way to understand the user’s needs and help define them because they

focus on the **Goal** of the user, and the value the user expects from the use. Moreover, they include the **role** of the user and the activity they wish to perform: the achievement of some **business goals**, in the context of some constraint. Cohn [41] proposes the following template for requirement modeling (Fig.9).

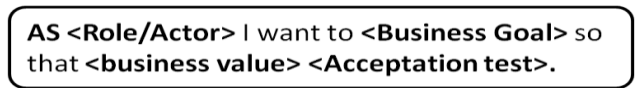


Fig .9.Template of User story.

It serves as a communication tool between the project team members. However, 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.

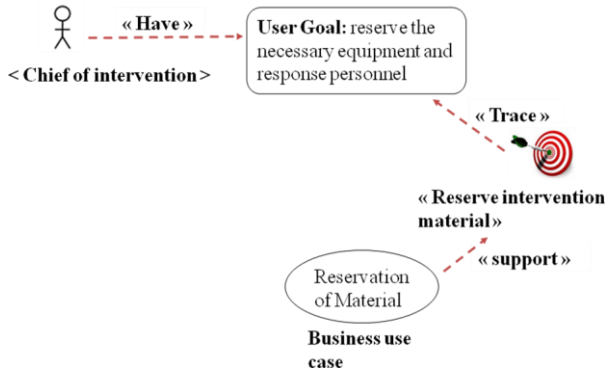


Fig.10. Example of User story representation.

- Description of business scenarios

In this step, we use Business Goal (supported by business use case) defined in the previous step in order to identify and describe business scenarios relative to the actors and their stories (Fig.11). Furthermore, we describe the different interactions from these scenarios with a UML sequence diagram. We obtain realized activities which can define new

web services, while the exchange of messages can then match the operations of these services.

3. Iteration Prioritization and Planning

The aim of this phase is 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. The end users and development team cooperate to prioritize and estimate the business service. In this phase, we borrow the principle given by [30] in which business service prioritization must be done in the manner in which all kinds of stakeholders give different viewpoints and give their importance (Fig.12).

We start to prioritize the business services according to 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.13).



Fig.12. Phase 3 of the proposed approach.

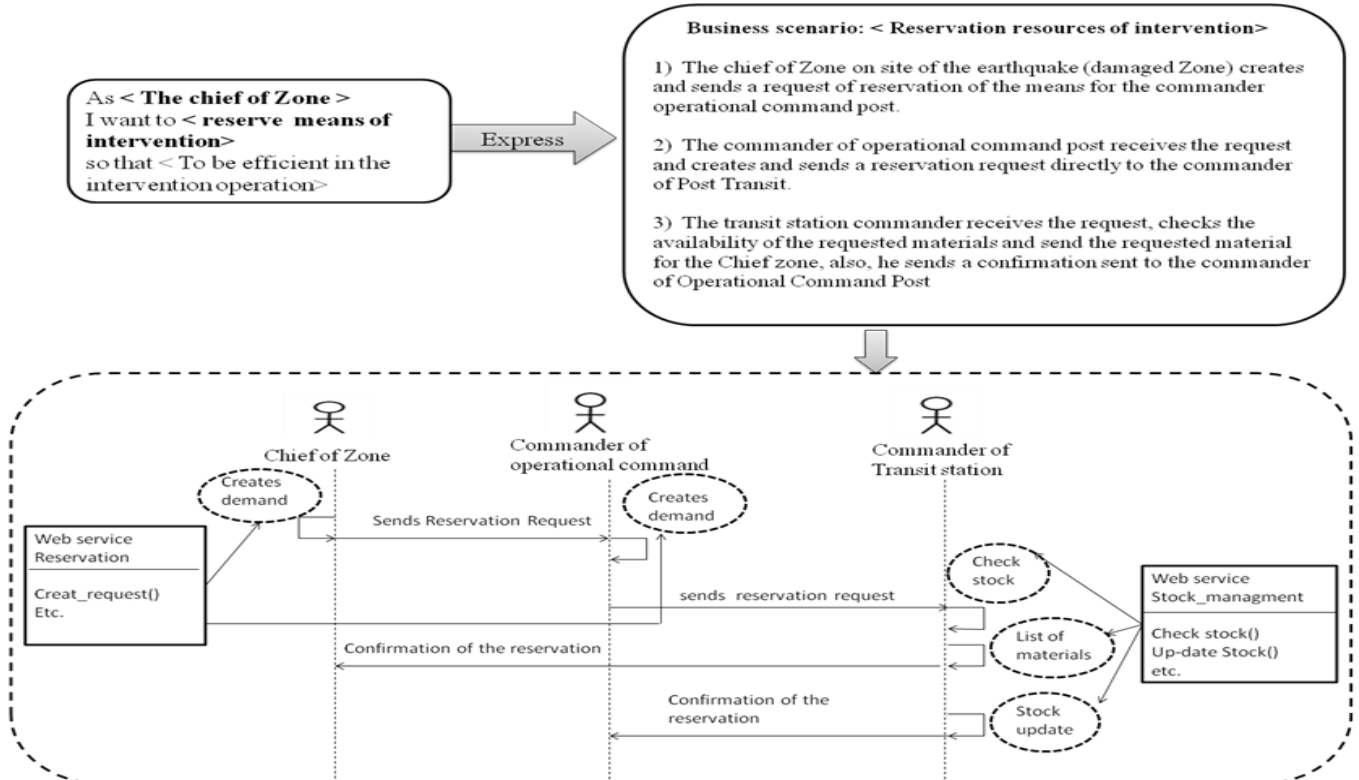


Fig.11. Example of transformation of User story into web service.

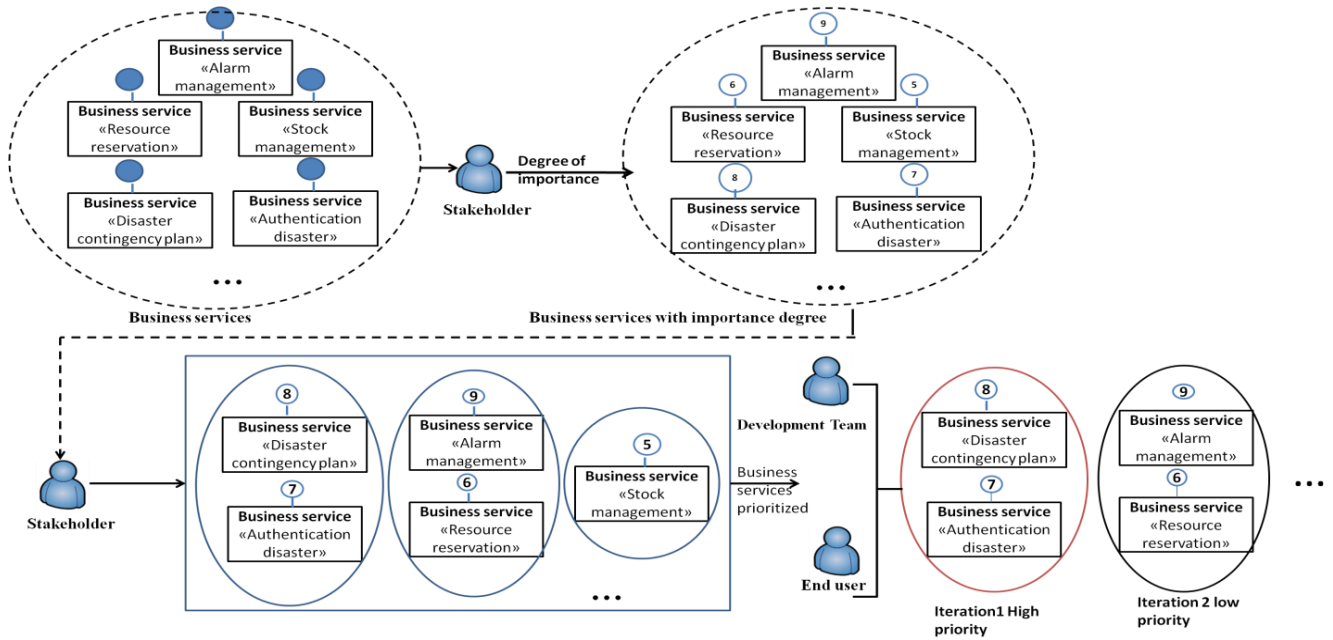


Fig.13. Example of prioritization of business service.

4. Iterations to Release

In this phase, we include several iterations of the management disaster system before the first release (Fig.14).

In the first time, we proceed to specify and design all the service-oriented architecture components of the management disaster system.

In the second time, we use 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 the third time, we test all services that are coded in the previous step. For this purpose, we use unit tests. This process aids the programmers to understand all the coding problems.

Finally, we proceed to evaluate the disaster management system by the actors of direction of civil protection of Béjaia. For this, we use some adequate evaluation techniques such as thinking aloud for instance, in order to study how the users use this system and their reactions.

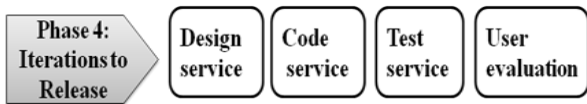


Fig.14. Phase 4 of proposed approach.

IV. CONCLUSION

In this paper, we have proposed an agile process combining a user centered approach and the service oriented paradigm for the development of interactive services applied to the disaster management domain. 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 systems 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.

One importance of Agile Methods is that the work is organized in a series of iterations in which the User goals to be dealt with are developed and the user-centered approach seems to be more appropriate for the early phases of a development process where an involvement of end users is of great importance.

In this article, we focus primarily on the first two analysis and iterations prioritization and planning phases due to space constraints. We applied this approach to an Earthquake management case study relative to the direction of the Civil Protection of Béjaia (Algeria). 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.

Finally, as a research perspective, we tend to go further towards implementation and deployment of the designed services in collaboration with the general protection service, the unity protection service and the administration and logistics service.

ACKNOWLEDGMENT

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

REFERENCES

- [1] Agile Alliance, "Manifesto for Agile Software Development," Technical report, 2001, Accessible at: <http://www.agilealliance.org>.
- [2] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "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>, 2002.
- [3] E. Not, C. Leonardi, C. Mennecozzi, F. Pianesi, and M. Zancanaro, "Beyond Usability: A New Frontier for User-Centered Design of "Future Internet" Services," LNCS 5468, pp. 107–116, Springer-Verlag, 2009.
- [4] ISO 13407, "Human-centered design processes for interactive systems," International Standard, 1999.
- [5] B. Van de Walle, M. Turoff, and S. R. Hiltz, eds., Information systems for emergency management. ME Sharpe, 2009.
- [6] A. Arsanjani, G. Bouch, T. Boubez, P.C. Brown, D. Chappell, J. Vadoss, T. Erl, N. Josuttis, D. Krafzig, M. Little, B. Loesgen, A.T. Manes, J. McKendrick, S. R. Talbot, S. Tilkov, C. Utschig-Utschig, and H. Wilhelmsen, "SOA manifesto," 2009, Accessible at: <http://www.soa-manifesto.org>.
- [7] D. Groves, "Successfully planning for SOA," *BEA Systems Worldwide*, 11 Sept 2005.
- [8] O. Zimmerman, P. Korgdahl, and C. Gee, "Elements of Service-Oriented Analysis and Design", IBM Corporation, June 2004, Accessible at: www-106.ibm.com.
- [9] Web Services Activity. W3C. Accessible at: <http://www.w3.org/2002/ws/>
- [10] T. Erl, "SOA Principles of service Design," Chap. 3, pp. 25-66, Prentice Hall, 2009.
- [11] Z. Mahmood, "The promise and limitations of service oriented architecture," *International journal of Computers*, vol. 1, no 3, 2007, pp. 74-78.
- [12] A. Arsanjani, and K. Holley, "The Service Integration Maturity Model: Achieving Flexibility in the Transformation to SOA," *IEEE International Conference on Services Computing*, Chicago, 2006, pp. 515.
- [13] P. F. Brown, R. Metz and B.A. Hamilton, "Reference model for service oriented architecture 1.0," Tech. rep, 2008, Accessible at: <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>.
- [14] A. Arsanjani, *SOMA : Service-oriented modeling and architecture*. IBM Corporation, 2004.
- [15] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall, 2005.
- [16] M. P. Papazoglou, and W. J. van den Heuvel, "Service-oriented design and development methodology," *International Journal of Web Engineering and Technology (IJWET)*, vol. 2, no 4, 2006, pp. 412-442.
- [17] D. Idoughi, "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)", PhD thesis, University A. Mira of Bejaia, Algeria and University of Valenciennes and Hainaut-cambresis, France, 2008, unpublished.
- [18] T. Silva da Silva, A. Martin, F. Maurer, and M. Silveira, "User-Centered Design and Agile Methods: A Systematic Review," *Agile Conference (AGILE)*, 2011, IEEE, 2011, pp. 77-86.
- [19] H. Javahery, H. Deichman, A. Seffah, and M. Taleb, "A User-Centered Framework For Deriving A Conceptual Design From User Experiences: Leveraging Personas And Patterns To Create Usable Designs", In *Patterns and Architectures for HCI*, Springer Verlag, pp. 53-81, 2009.
- [20] J. Gulliksen, B. Göransson, I. Boivie, S. Blomkvist, J. Persson, and Å. Cajander, "Key principles for user-centred systems design," *Behaviour and Information Technology*, vol. 22, no 6, 2003, pp. 397-409.
- [21] A. Cooper, R. Reimann, and D. Cronin, *About Face 3.0: The Essentials Of Interaction Design*. Wiley publishing, Indianapolis, 2007.
- [22] H. Beyer, and K. Holtzblatt, *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann Publishers, Inc, San Francisco, CA, USA, 1998.
- [23] J.M. Carroll, *Making Use: Scenario Based Design of Human-Computer Interactions*. MIT Press, 2000.
- [24] X. Ferre, "Integration of Usability Techniques into the Software Development Process," In *Proceedings of the International Conference on Software Engineering, ICSE'2003*, May 3-10, Portland, Oregon, USA, pp. 28-35, 2003.
- [25] P. Zhang, J. Carey, D. Te'eni, and M. Tremaine, "Incorporating Human-Computer Interaction Development into SDLC: A methodology," *Communications of the AIS*, Vol. 15, 2005, pp. 512-543.
- [26] D. Te'eni, J. Carey, and P. Zhang, *Human-Computer Interaction: Developing Effective Organizational Information Systems*. John Wiley & Sons, Inc, Hoboken, NJ, USA, 2007.
- [27] D. Idoughi, A. Seffah, and C. Kolski, "Adding user experience into the interactive service design loop: a persona-based approach," *Behaviour & Information Technology*, vol. 31, n.3, 2012, pp. 287-303.
- [28] D. Robey, R. Welke, and D. Turk, "Traditional, iterative, and component-based development: a social analysis of software development paradigms," *Information Technology and Management*, vol. 2, n. 1, pp. 53–70, 2001.
- [29] Z. Mushtaq, and M. R. J. Qureshi, "Novel Hybrid Model: Integrating Scrum and XP," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 6, 2012, pp. 39-44.
- [30] M. Shahrbanoo, M. Ali, and M. Mehran, "An Approach for Agile SOA Development Using Agile Principals," *arXiv preprint arXiv:1204.0368*, 2012.
- [31] A. Elsamadisy, "SOA and Agile: Friends or Foes?," 2007. Available: <http://www.infoq.com/articles/SOA-Agile-Friends-Or-Foes>.
- [32] D. Idoughi, M. Kerkar and C. Kolski, "Towards new web services based supervisory systems in complex industrial organizations: Basic principles and case study". *Computers in Industry*, vol. 61, n. 3, 2010, pp. 235–249.
- [33] A. Ivanyukovich, G. R. Gangadharan, V. D'Andrea, and M. Marchese, "Towards a service-oriented development methodology," *Journal of Integrated Design and Process Science*, vol. 9, no 3, 2005, pp. 53-62.
- [34] A. Boesteanu, "Agile SOA: a more effective approach to implementing SOA within your organization," 2009, Accessible at: <http://www.liquidhub.com/docs/agilesoa.pdf>.
- [35] B. Losada, M. Urretavizcaya, and I. Fernández-Castro, "A guide to agile development of interactive software with a "User Objectives"-driven methodology". *Science of Computer Programming*, vol. 78, n. 11, 2012, pp. 2268–2281.
- [36] P. Rannikko, "User-Centered Design in Agile Software Development," M.Sc. Thesis, University of Tampere, School of Information Sciences, *Interactive Technology*, April 2011, 72 pages.
- [37] S. Blomkvist, "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.
- [38] H. Beyer, K. Holtzblatt, and L. Baker, "An Agile Customer-Centered Method Rapid Contextual Design," In: *Extreme Programming and Agile Methods - XP/Agile Universe 2004*, Springer, 2004, pp. 50-59.
- [39] D. Sy, "Adapting Usability Investigations for Agile User-centered Design". *Journal of Usability Studies*, vol. 2, no.3, 2007, pp. 112-130.
- [40] C. Mörck, "A Service Layer Taxonomy and Structure for Service-Oriented", Thesis, Stockholm, Sweden, 2009.
- [41] C. Mike, "User stories applied: For agile software development," Addison-Wesley Professional, 2004.