Managing Requirement Changes in Health Informatics Projects

Rand Obeidat
Health Information Technology, Department of Information Systems
College of Engineering & Information Technology
University of Maryland-Baltimore County (UMBC)

[Abstract] Several studies have concluded that requirement changes are one of the major problems that have an effect on project environments. Furthermore, requirements could not be stabilized, and change is unavoidable and inevitable. As a result, requirement changes should be managed and controlled in a proper way in order to minimize their consequences. Therefore, this paper investigates the main aspects of Managing Requirement Changes (MRC) in the context of Health Informatics (HI) projects based on a conceptual framework and shows the relationship between these aspects after conducting interviews with four IT health professionals.

[**Keywords**] Requirement change; health informatics; MRC; HI;

Introduction

Changing requirements are inevitable to software. As changes occur during the entire software project lifetime, continuously changing requirements represent a key management challenge for software development efforts (Bohner, 1996; Lam, Loomes, & Shankararaman, 1999; Lauesen, 2002). Changing requirements can affect the cost, schedule, and quality of software projects. As a result, inability to manage the changing requirements may lead to inconsistencies in system requirements and result in failure of the software project (Bhatti et al., 2010; Jones, 1996; Lock & Kotonya, 1999; Oz, 1994; Schwalbe, 2010). Moreover, it is found that lack of user input, incomplete requirements and changes in requirements and scope were the top three causes of IT project failures or cancelation (El Emam & Koru, 2008; Kumar & Kumar, 2011).

In the context of Health Informatics (HI) projects, it is indicated that there is a difficulty of identifying requirements for the various groups involved. Individuals gathering requirements may not include all the necessary people within an organization, or these individuals may not know how to effectively communicate their requirements. As a result, requirements changes are inevitable, and there should be a proper way to manage changing requirement especially in this evolving field (Kaplan & Harris-Salamone, 2009; Prijatelj, 1999).

This paper summarizes causes and factors, main classifications of requirement changes, and processes used for controlling and managing requirement changes. After that, it investigates the main aspects of the MRC phenomenon in the context of HI projects, their relationships, and what IT health professionals think about them.

Background and Related Work

Requirements change is defined as the addition of new requirements or the removal or modification of existing requirements(Kavitha & Sheshasaayee, 2012; Lam & Shankararaman, 1999). Software requirements continue to evolve during software development lifetime and subsequent life of a system. Changing requirements are recognized as a major cause of project failure and can affect the cost, schedule and quality of software (Bhatti et al., 2010; Jones, 1996; Lock & Kotonya, 1999; Oz, 1994; Schwalbe, 2010). The Standish CHAOS Report, which surveyed 9,236 IT projects, found that the top three causes of project failure were lack of user input, incomplete requirements, and changing requirements (Kumar & Kumar, 2011).

The change in requirements can be requested by the customer or end user, the project team, the testing team, and the organizational standard group. For instance, there can be many reasons for the change in initial requirements; customers can request changes in requirement if some changes occur in the business process of the client's organization, if analysts couldn't initially understand the exact requirements of customer, or if initially provided requirements remained incomplete. A project team can, also, request changes in requirements if some requirements are technically impossible to implement or if some requirements fall outside the scope boundaries of the software project (Bhatti et al., 2010; Ting, 2011)

Studies stated that there is an important need to manage requirements change (MRC) for at least three reasons: (1) Much software is delivered incrementally, so changed requirements are established based on feedback from the previous increment and incorporated into the next incremental delivery. (2) Typically, changing requirements are the main drivers for software maintenance and re-engineering activities. (3) Many organizations have legacy systems that are critical to sustaining commercial operation; such systems must evolve in order for organizations to survive and remain competitive because replacing or re-engineering such systems is not always feasible (Gilb & Finzi, 1988; Lam et al., 1999; Lientz & Swanson, 1980). As a result, managing requirement changes affects and is affected by three important dimensions as shown below in Figure 1.

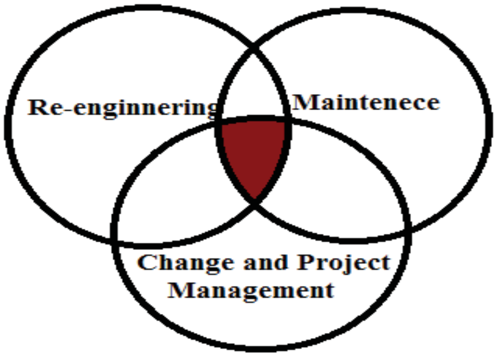

Figure 1. The dimensions of managing requirement changes

Several studies concluded that in order to manage requirements change, it is important from a managerial viewpoint, to understand the cause and effect relationships**.** For example, stakeholders conflict, project overrun, and wrong implementation are caused by requirement change (Lam & Shankararaman, 1998). Also , understanding of the causes and consequences of change can support requirements management and make progress towards the goal of change anticipation (McGee & Greer, 2012).

A number of requirements change classifications have been proposed. Much empirical and theoretical work has focused upon software maintenance classification, which includes corrective, adaptive, and perfective changes (Benestad, Anda, & Arisholm, 2009; Swanson, 1976). In addition, a comprehensive classification, combined with a focus upon activity, has been proposed (such as enhancement, performance improvement, and consideration of future maintenance) with what is being changed (such as interface, properties, and business rules) (Chapin, Hale, Khan, Ramil, & Tan, 2001).

By contrast, some studies identified a classification that will support the need of change anticipation by focusing on the cause of the change. They divided empirically gathered requirements changes occurring during software development into five categories depending upon the source of the change: (1) fluctuations in the organization, (2) product market environment, (3) increased understanding of requirements, (4) consequences of system usage, and (5) changes necessary due to customer migratory or adaptation issues (Harker, Eason, & Dobson, 1993). Moreover, McGee, and Greer indicated that software requirements change source taxonomy containing the change domains of Market, Organization, Vision, Requirements Specification, and Solution (McGee & Greer, 2012) as shown below in Table 1.

Table 1. Requirement Source Change Domain

Change domain	Description
Market	Differing needs of many customers, government regulations.
Organization	Changing strategic direction of a single customer, customer organization considerations, political climate
Vision	Change to the problem to be solved, product direction and priorities, stakeholder involvement, process change
Specification	Change to the specification of the requirements of the established problem, resolution of ambiguity, inconsistency, increased understanding.
Solution	Change accommodating new technical requirements, design improvement, solution elegance.

Further studies discussed managing and controlling requirements change; a measurement-action framework is proposed that can assist software managers manage requirements change and evolution. It is organized around four main areas of concern: (1) planning for change, (2) assessing impact of change, (3) determining changeability, and (4) assessing effectiveness at handling change (Lam et al., 1999). Bhatti identified a formal methodology of the change management process can be adopted in any software company to manage their varying requirements. It consists of six phases; initiation, receiving, evaluation, approval or disapproval, implementation, and configuration of change requests (Bhatti et al., 2010).

Moreover, Ting has constructed a control process of requirements change, and has conducted research on the measure of the requirements stability. He concluded that it is impossible to stabilize requirements completely; it is possible to only try to avoid and reduce the impact of requirements change and actively respond to thinking, management, and technology aspects. The thinking aspect refers to solving problems mainly from the subjective understanding; the management aspect refers to specifying the control process of requirements change, enhance the training and exchange, etc. The technology aspect refers to controlling requirements change through the introduction of advanced technology (Ting, 2011). After that , Kavitha and Sheshasaayee introduced guidelines to manage changes effectively: (1) recognize that change is inevitable; (2) baseline the requirements; (3) establish a single channel to control change, such as a change control board (CCB); (4) use a change control system to capture changes; and (5) manage change hierarchically (Kavitha & Sheshasaayee, 2012). A summary of early studies on managing requirement change is presented in Table 2.

Table 2. Early Studies on Managing Requirements Change

Author (s)	Theme	Findings/Conclusions
(Lam & Shankararaman, 1998) (McGee & Greer, 2012)	Cause and effect of software requirements change	Requirement change is a problem that could have many different effects in a project environment, and that understanding the cause and effect relationships is important from a managerial viewpoint. Increased understanding of the causes and consequences of change can support requirements management and also make progress towards the goal of change anticipation
(Benestad et al., 2009) (Swanson, 1976) (Harker et al., 1993) (McGee & Greer, 2012)	Classifications of requirements change	A number of alternative classifications have been proposed, focused upon software development, maintenance, or both, which often have the intention of meeting different objectives. Cause-focused classifications have been concluded as follows: Harker divides empirically gathered requirements changes occurring during software development into five categories depending upon the source of the change. The software requirements change source taxonomy contains the change domains of Market, Organization, Vision, Requirements Specification, and Solution.
(Lam et al., 1999) (Bhatti et al., 2010) (Ting, 2011) (Kavitha & Sheshasaayee, 2012)	Managing and controlling requirements change	They proposed measurement-action framework that can assist software managers manage requirements change and evolution. ,it is organized around four main areas of concern: 1) planning for change, 2) assessing impact of change, 3) determining changeability, and 4) assessing effectiveness at handling change. A formal methodology of the change management process can be adopted in any software company to manage their varying requirements. It consists of six phases; initiation, receiving, evaluation, approval or disapproval, implementation and configuration of change requests. They constructed a control process of requirements change, and have conducted the research on the measure of the requirements stability. As concluded, it is impossible to stabilize requirements completely, only try to avoid and reduce the impact of requirements change, and actively respond from thinking, management and technology aspects. The authors introduced guidelines to manage changes effectively.

In the context of Health Informatics (HI) projects, Haux identified three core driving forces influencing HI research as follows: (1) progress in information processing methodology, information, and communication technology, (2) progress in medicine and health care, (3) changes in needs, requirements, and expectations of societies (Haux, 2010).

One of the critical success factors is to fulfill needs (whether stated or not) rather than only the requirements of the users (Brender, Ammenwerth, Nykänen, & Talmon, 2006). However, HI projects tend to have a complexity and difficulty in communication among specialties, stakeholders, clinicians, and implementers. So, it is difficult to identify requirements for the various groups involved because individuals gathering requirements may not include all the necessary people within an organization, or these individuals may not know how to effectively communicate their requirements; sometimes, senior management or IT personnel don't understand the clinical environment or workflow or don't agree on what needs to be done; sometimes, they don't provide sufficient or meaningful incentives to change. As a result, the difficulty in fully understanding workflow besides regulatory and policy requirements, as evidenced by workflow changes, will result in endless workarounds and iterations (Heeks, 2006；Kaplan & Harris-Salamone, 2009; Teixeira, Ferreira, & Santos, 2007). So, requirement changes are inevitable, and there should be a proper way to manage requirements changes, especially in this evolving context of HI projects.

Due to the limited number of studies in the literature dedicated to the MRC in the context of HI projects, we decided to conduct this pilot study to investigate the main aspects of MRC in this context and what IT health professionals think about them.

Research Methodology

This section covers the research questions, the research methodology, and the data collection methods used in the study.

Table 3. Research Questions

Research question	Aim	Example Answer
RQ1: Does the methodology approach for software development affect the likelihood of a software requirements change?	Are there certain types of SW development approaches become susceptible for requirement changes more often than others?	" Yes, it does because a good example of that right now is the waterfall approach; it does not really fit working of what we need"
RQ2: What factors or causes that influence MRC?	As a step in understanding MRC, it is important to identify factors or causes that have an influence on MRC phenomenon.	"Just because someone says that they want a total track volume does not mean that they know all things they wanna track in addition to the volume or what they wanna do with the data."
RQ3: what have been the overall impacts of Managing Requirements Changes (MRC)?	To investigate to what degree is MRC a serious concern.	"Well, right now we are about a year and half late on updating our product……….and that is directly related to resource limitations."
RQ4: When requirement change requests are identified, how are they typically handled?	In order to identify possible alternatives for MRC handling, we first need to understand how they are handled today.	"We have a committee that meets on a weekly basis depending on the project, so if we have any updates, then it will be defined if they are needed."

Research Questions

Due to the limited number of studies in the literature dedicated to managing requirements changes (MRC) in the context of Health Informatics (HI) projects, we decided to focus on understanding the MRC phenomenon and its place in the HI landscape. Thus, most of the research questions outlined in Table 1 are about cause and effect, classification, managing, and controlling requirement change.

Research Design

Interviewing as a data collection method is used to find out what people think and focuses on their opinions and procedures or approaches they actually use (Merriam, 2009). Moreover, a semi structured interview provides good balance between richness and replicability. Thus, it was chosen as the main tool to collect data to determine what IT health professionals think about the MRC phenomenon and what they actually do to handle it. In addition, we reviewed the literature and used document analysis to have an access to how MRC is formally processed, since it is stable and has no observer or interviewer bias in order, to triangulate the collected data (Merriam, 2009). According to the reviewed literature in this paper, we model a conceptual framework that includes the whole aspect of managing the requirement change phenomenon as shown in Figure 2.

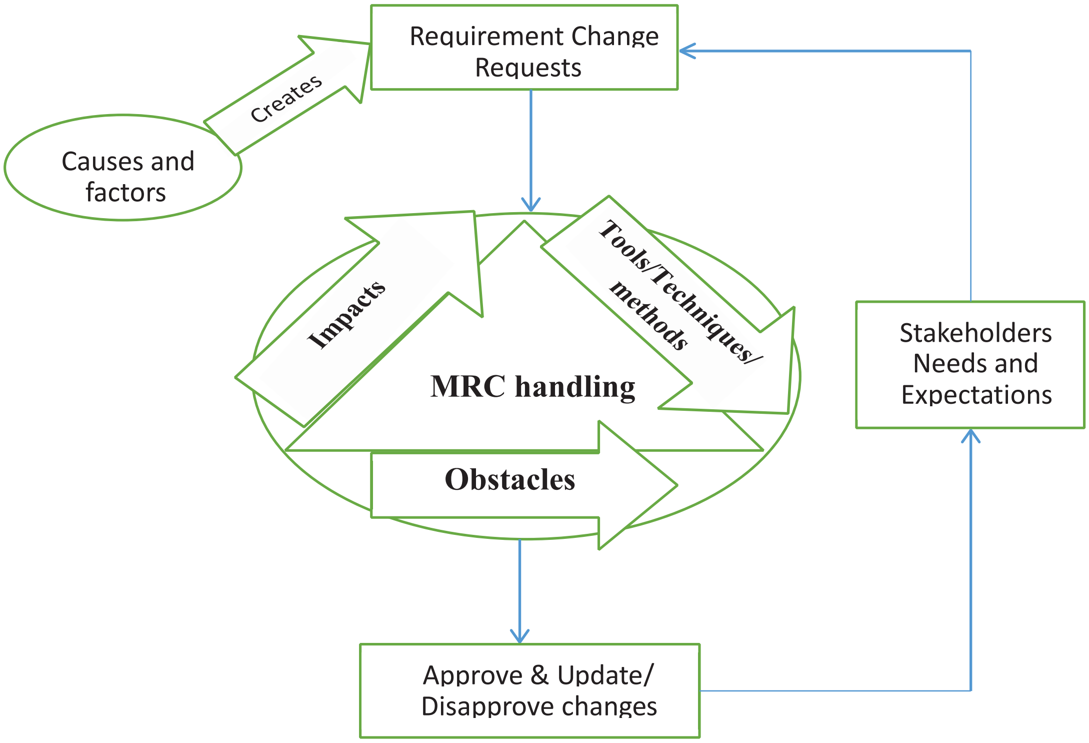
Requirement Change Requests
Creates
Causes and factors
Impacts
Tools/Techniques/ methods
MRC handling
Obstacles
Stakeholders Needs and Expectations
Approve & Update/ Disapprove changes

Figure 2. Conceptual framework of the whole aspects of managing requirement changes phenomenon

This conceptual framework resulting from the literature review formed the starting point; a deductive approach was chosen first. Deductive reasoning, in contrast to inductive reasoning, starts with a theory about the topic of interest (Merriam, 2009). Moreover, a deductive approach is useful if the general aim was to test a previous theory in a different situation or to compare categories at different time periods (Elo & Kyngäs, 2008). Then, the data can be structured by the use of existing theoretical notions and frameworks (deductively) and combined with the flexibility of exploring the data with an open mind (inductively), which is called analytic induction (Merriam, 2009).

The process is illustrated as follows:

Theory → Hypothesis → Interviews → Confirmation (Disconfirmation) → Further Improvements

--Theory: Managing software requirement changes

--Hypothesis: The conceptual framework is applicable in the situation of Health Informatics (HI) projects.

--Interviews: 4 qualitative semi-structured interviews.

--Further Improvements: what could be added or removed, any interesting insights.

The Procedures

Our findings are based on reviewing the literature, documents reviews, and interviews with 4 practitioners with varying job titles who work in different companies and different HI projects, as shown below in Table 4.

Table 4. Organization of Interviews

Number of Interviews	Interviewee role	Description
1	P1, Informatics and Reporting Senior Manager.	The interviewee was responsible for automating the customer reports recording at hospital.
1	P2, Systems Analyst	The interviewee was responsible for the implementation of nurses productivity system.
1	P3, Senior Clinical Scientist	The interviewee was responsible for designing clinical decision support tools.
1	P4,Business Analyst	The interviewee was responsibility for writing requirements and integrating work between subject matter expert, development and testing teams through the implementation of health monitoring system that help doctors and nurses remote monitor of patients.

A purposeful, snowball sampling method was used (Merriam, 2009) to gain in-depth information from key informants about the MRC in HI projects. Participants were identified from the researcher’s initial contact with individuals known to our mentor. After identifying two participants, a network was built by asking these first participants to suggest additional participants for interview. The sample wasn’t random or large, but it contains various companies, HI projects, and job titles.

The interviews were semi structured, including both specific and open-ended questions; the questions began with background information (such as experience in the domain of HI projects, current HI project, and participants' roles and responsibilities in the project). Then, we asked the participants to describe a particular example of requirement change request, including its causes or factors, obstacles, and impacts. We conducted interviews over the phone, then audio-recorded and transcribed them all for analysis. In addition, this study procedure was approved by the Institutional Review Board (IRB), participants were assured of confidentiality, data were processed anonymously, and informed phone interview consent was obtained.

We coded the interview data using a combination of pen, paper, and Microsoft Excel; the interview transcripts were mainly analyzed using content analysis. Content analysis is defined as a research method for the subjective interpretation of the content of text data through the systematic classification process of coding and identifying themes or patterns (Hsieh & Shannon, 2005). This approach was chosen because the main strength of the content analysis is that existing theory can be supported and extended with the main intention of classifying the many words of the interview text into much smaller content categories (Zhang & Wildemuth, 2009). In the analytic process, three phases could be classified: the preparation phase; the organizing phase; and the reporting phase (Zhang & Wildemuth, 2009).

Organizing phase: In the organizing phase, a categorization matrix was developed as shown in Table 5 below. The categories (main codes) were divided into more detailed codes; the data has been coded in correspondence with the identified categories (according to the conceptual model). While the detailed codes were derived inductively, the analysis was mainly deductive because the pre-defined "code families" formed the starting point of the analysis, and the newly identified codes were clustered within these codes.

Table 5. Categorization Matrix

	Understanding causes/factors	Managing & understanding impacts	Knowing obstacles	Managing and handling requests	Using Tools/Methods /Techniques
What are the aspects IT health professionals have to deal with for managing requirement changes?					

Reporting phase: The outcomes of the categorization and coding process are further discussed in the following section.

Results and Discussion

In this section, the results of the analysis and discussion are provided; also, the way that some of IT health professionals deal with managing requirement changes was experienced by the participants in this project is described. An overview of the actual implementation of coding the data, as explained in section 3.3, can be found in Appendix I.

Understanding Causes/ Factors

All participants interviewed could recognize requirement change causes/factors; most of them focused on the relationship between the software (SW) development lifecycle used and the likelihood of requirement changes. Some of them focused on the need for transition from the waterfall to the agile approach as one of the factors that play a major role in managing and controlling requirement changes. *"Yes, it does because a good example of that right now is the waterfall approach, it does not really fit working of what we need (P1)"; " The benefit of the agile is that we can make timely changes and take the requirements in smaller spoon rather than everything in the end in waterfall approach (P4)."* However, other participants believe that the software development approach affects the likelihood of requirement changes but it is situational. *"Yes, I think so. It is hard to say just say is that any specific general school or approach that we have needed because it is really situational to the actual situation (P2)"; "I think it changes the likelihood of requirement changes in certain areas or for certain reasons (P3)."*

Moreover, some of the participants indicated that the causes of requirement changes may depend on SW development lifecycle's stage. *"It depends on where we are lying in the lifecycle of the product, in the very beginning when we release a product; changes in the requirement have to do with defects or errors. As we move forward in the lifecycle of the product, it will become more about enhancements (P1)"; "From my perspective I would almost call them as transcriptional errors, cleanup in the implementation (P3)."*

Some participants stated that as software technologies and equipment become unrelated to each other, requirement changes increase. *"I think the reason when we have requirement changes all the time because of dealing with multiple different software technology and equipment and they are not related (P2)."*

Another causes identified by the participants are as follows; people don't know what they need or they may come up with new ideas for different purposes, such as how the product can be changed and can bring better health care delivery. Besides, another major cause is the compressed schedule in project development, which impedes the project team from implementing all requirements or the change itself. "*I say that a lot of examples would be people don't know exactly what they need, you know not having all the details or the specifics. Just because someone says that they want a total track volume, does not mean that they know all things they wanna track in addition to the volume or what they wanna do with the data (P2)"; "We always get customers have come with new ideas how a product can change, or better their health care delivery or analytics behind what they see (P1)"; "Most of the requirement changes are that are needed due to lack of time to manage; you know, we have a compressed schedule, and, sometimes, we don't have enough time to dive into different scenarios (P4)".*

According to the document reviews, we found that request forms indicate the following causes to be selected by the requestors: (1) new software installation/implementation; (2) replacement of existing system(s); (3) software upgrades to existing product(s); (4) add-on functionality/modules/custom; (5) creation of interfaces; and (6) new business programs requiring infrastructure. However, they consider each of these reasons as a need for proposing a separate project and could not find any specific forms to request requirement changes in a particular single project.

Managing & Understanding Impacts

With respect to the impacts, two aspects were found. First, the influence of the requirement change itself on other projects triples constraints like time, cost, and other requirements or project scope. "*Well, you know it is a lot of regulatory and tractability things that we need to take in mind when we manage the requirement (P4)"; "Well, right now we are about year and half late on updating our product, and we should be updating our product every six months (P1)"; "Projects are done faster and run on a timely basis; they completed in the scope that usually requested in on the negative side (P2)"; "I think there is some effort for maintaining the documentation and probably it would take a lot of time (P3)."* Referring to document reviews, we noticed that request forms contain sections for the impact of the requested change on budget and expenses; also, we noticed a section for justifying the impact of the change on the overall scope of the project.

Second, if the requirement changes are not managed properly, other factors, such as the final outcomes, leadership and management and bad complications and implications on patient's health care after using the product will be influenced. *"Better, more accurate outcomes on the negative side if they are not managed properly, frustrations from leadership and requesters, more push back that we get (P2"; " The negative side if they not managed properly, especially in the health care industry, you could have serious complications or implication on patient's health care. When things need to be done, and they are not documented and not traced, it is possible that we could be in bad situation with customer's health care (P4)."*

Knowing Obstacles

We believe that it is important to be aware of the obstacles to managing requirement changes, since this is one of the major steps to know the solution for managing and controlling changes properly. Participants identified some major obstacles, such as the time needed to control and manage changes: "*We do have a specific process we need to follow, and that takes time. Sometimes, the checks for the changes need to be done immediately, but it does take a week or may be more to get the change on (P4)";* resistance to fill all information needed in change request forms: *"Some of obstacles have been, I think, in regard to resistance from the requesters; they don't want to fill out all the information in the forms. They feel that it is an additional barrier, frustrating, and takes too long. They don't wanna follow the proper steps (P2)"*; and unlimited resources to handle and make a change and sufficient background in the software: "*From my perspective, we just don't have the unlimited resources for fixing the request; we don't have enough QA resources to handle the number of enhancement requests that are going on, and that also has to do* *with the fact that our software is intensive and analytic. So, if we make any enhancement with analytic product, we have to have a QA person who knows that (P1)"; "I think mainly resources and prioritizations (P3)."* Moreover, some of the participants believe that whether the software development approach is in-house or outsourced affects managing requirement changes easily: *"Yes, I think so because we don't have an access to some component, and we don't have the same people."*

Managing and Handling Requests

All the participants focused on the importance of having documented process and formal steps, a committee or board, prioritization and study of the impact of a change, and reporting to stakeholders are all profound sub-aspects to managing and handling change requests in a proper way through various communication means, such as phone and customer IT help desks: *"Yes, we have a committee that meet on a weekly basis depending on the project, so if we have any updates, then it will be defined if they are needed (P2)"; "What happens is the majority of requests come from the development team and testing team and what they do is contact me; if we find a change after reviewing them would contact the manager to make sure that what actually defined is still in the scope of the system (P4)."*

Using Tools/Methods/Techniques

All the participants stated that they use specific tools and techniques to help them document and track changes and identify and mark different status changes and reasons for requirement changes: *"Yes, we have specific forms and documents, regulations setup to keep on have additional steps for additional requirements (P2)"; "Yes we use a requirement management tool that has a way to let us track and document the changes (P4)"; "Yes, we do. I don't remember the name of the product. I am pretty sure it is a Microsoft product (P1)."* In addition, make changes available and readable for other team members so they can make decisions: *"I don't personally use it; it seems comprehensive in version control and tracking, different status changes and reasons for requirement changes, and they can export them in a revised version, which typically I can read. "*.

Relationships between Aspects

The different aspects of managing requirement change (MRC) requests cannot be viewed in isolation. In some ways, they are all related to each other. Two aspects that most influence the other aspects are understanding causes or factors and using tools/methods or techniques (see Table 5). This is in accordance with the aspects found most important in the literature review.

Conclusions and Future Work

This paper focuses on the importance of managing requirement changes (MRC) in the context of Health Informatics (HI) projects. It illustrates the main aspects that should be considered in MRC as understanding causes and factors, managing and understanding impacts of changes, knowing obstacles, managing and handling requests, and using tools and techniques. In addition to the relationships between those aspects, this study deduced that understanding causes or factors and using tools or techniques are the two factors that have the most influence on the other aspects. Furthermore, different IT health professionals look at the aspects from different angles, as illustrated in the results and discussion.

In this study, we conclude that managing requirement changes is influenced by the software development approach (methodology, lifecycle, and whether it is in house or outsourced) and the extent that software technologies and equipment become unrelated. Also, it influences project constraints such as time, cost and scope, and, also, on other on factors, such as the final outcomes, leadership and management, complications, and implications on patients' health care after using the product. Some obstacles, such as resistance to filling all needed information in request forms and a lack of time and resources with appropriate backgrounds in the field to manage them, are present. Moreover, it is necessary to have formal and documented processes and using tools, methods, and techniques that help document, track and report changes properly as they occur.

Future research is needed to investigate the perspective of front-end stakeholders, such as physicians and nurses who use specific IT health care requirement changes, causes and factors, how they report them, and efficiency of changes. It is worth increasing the sample size of the IT health professionals and to study the correlation and significance of causes and factors that influence requirement changes and their management.

Table 6. Relationships between MRC aspects

	Understanding causes/factors	**Managing & understanding impacts**	**Knowing obstacles**	**Managing and handling requests**	**Using Tools/Methods /Techniques**
Understanding causes/factors	X				
Managing & understanding impacts	For each cause of change, there would be an influence on project constraints itself, such as time, scope, and cost. Sometimes, there is a severe impact on management and leadership, also patient's health care	X	For instance: SW approach (in-house or outsourced) may have an impact on managing a requirement change easily and properly; there should be an efficient way of communicating between all parties even though they are working virtually, remotely, or from various companies	Documented processes and formal precise steps have an impact on getting better outcomes and creating a stronger team environment .	
Knowing obstacles	Compressed schedules through software or product development cause an obstacle to adjust all required needs. As a result,		X	It is necessary to have sufficient resources who follow formal steps to manage and handle requirement changes.	
Managing and handling requests	Justification for request changes cause (s) is(are) needed in the first place to manage requirements in proper way and look for possible solutions.		There should be continuous improvements in requests forms used by stakeholders and the way they report to the committee.	X	
Using Tools/Methods /Techniques	Causes of change can be documented using specific tools, methods, or techniques; it is important to do that, since project stakeholders may ask about the reason or help them make a decision.	The impact of requirement change, progress, and status of change can be tracked using specific tools and techniques.	Resources should have sufficient background and knowledge in tools and techniques used.	To manage and handle requirements, change requests in sufficient time, should be automated business processes to support timely changes.	X

References

Benestad, Hans Christian, Anda, Bente, & Arisholm, Erik. (2009). Understanding software maintenance and evolution by analyzing individual changes: A literature review. *Journal of Software Maintenance and Evolution: Research and Practice, 21*(6), 349-378.

Bhatti, Muhammad Wasim, Hayat, Farah, Ehsan, N, Ahmed, Sohail, Ishaque, A, & Mirza, E. (2010). *A methodology to manage the changing requirements of a software project.* Paper presented at the Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on.

Bohner, Shawn A. (1996). *Impact analysis in the software change process: A year 2000 perspective.* Paper presented at the Software Maintenance 1996, Proceedings., International Conference on.

Brender, J., Ammenwerth, E., Nykänen, P., & Talmon, J. (2006). A Pilot Delphi Study. *Methods Inf Med, 45*, 125-136.

Chapin, Ned, Hale, Joanne E, Khan, Khaled Md, Ramil, Juan F., & Tan, Wui-Gee. (2001). Types of software evolution and software maintenance. *Journal of software maintenance and evolution: Research and Practice, 13*(1), 3-30.

El Emam, Khaled, & Koru, Akif Günes. (2008). A replicated survey of IT software project failures. *Software, IEEE, 25*(5), 84-90.

Elo, Satu, & Kyngäs, Helvi. (2008). The qualitative content analysis process. *Journal of advanced nursing, 62*(1), 107-115.

Gilb, Tom, & Finzi, Susannah. (1988). *Principles of software engineering management* (Vol. 4): Addison-Wesley Reading, MA.

Harker, Susan D. P., Eason, Ken, D, & Dobson, John E. (1993). *The change and evolution of requirements as a challenge to the practice of software engineering.* Paper presented at the Requirements Engineering, 1993., Proceedings of IEEE International Symposium on.

Haux, Reinhold. (2010). Medical informatics: past, present, future. *International Journal of Medical Informatics, 79*(9), 599-610.

Heeks, R. (2006). Health information systems: Failure, success and improvisation. *International Journal of Medical Informatics, 75*(2), 125-137.

Hsieh, Hsiu-Fang, & Shannon, Sarah. E. (2005). Three approaches to qualitative content analysis. *Qualitative health research, 15*(9), 1277-1288.

Jones, Capers. (1996). *Patterns of software system failure and success*: Itp New Media.

Kaplan, Bonnie, & Harris-Salamone, Kimberly D. (2009). Health IT success and failure: recommendations from literature and an AMIA workshop. *Journal of the American Medical Informatics Association, 16*(3), 291-299.

Kavitha, D., & Sheshasaayee, Ananthi. (2012). Requirements Volatility in Software Maintenance *Advances in Computer Science and Information Technology. Computer Science and Information Technology* (pp. 142-150), Springer.

Kumar, S. Arun, & Kumar, T. Arun. (2011). Study the impact of requirements management characteristics in global software development projects: an ontology based approach. *International Journal of Software Engineering and Application, 2*(4).

Lam, W., Loomes, Martin, & Shankararaman, V. (1999). *Managing requirements change using metrics and action planning.* Paper presented at the Software Maintenance and Reengineering, 1999. Proceedings of the Third European Conference on.

Lam, W., & Shankararaman, V. (1998). *Managing change during software development: an incremental, knowledge-based approach.* Paper presented at the Proc. of the 10th Int. Conf. on Software Engineering and Knowledge Engineering, San Francisco, California.

Lam, W., & Shankararaman, V. (1999). *Requirements change: a dissection of management issues.* Paper presented at the EUROMICRO Conference, 1999. Proceedings. 25th.

Lauesen, Søren. (2002). *Software requirements: Styles and techniques*: Pearson Education.

Lientz, Bennet P, & Swanson, E Burton. (1980). Software maintenance management: A study of the maintenance of computer application software in 487 data processing organizations.

Lock, S., & Kotonya, G. (1999). An integrated framework for requirement change impact analysis.

McGee, Sharon, & Greer, Des. (2012). Towards an understanding of the causes and effects of software requirements change: two case studies. *Requirements Engineering, 17*(2), 133-155.

Merriam, Sharan B. (2009). *Qualitative research: A guide to design and implementation*: John Wiley & Sons.

Oz, Effy. (1994). When professional standards are lax: The CONFIRM failure and its lessons. *Communications of the ACM, 37*(10), 29-43.

Prijatelj, Vesna. (1999). Success factors of hospital information system implementation: what must go right? *Studies in Health Technology and Informatics*, 197-202.

Schwalbe, K. (2010). *Information technology project management*: Course Technology Ptr.

Swanson, E. Burton. (1976). *The dimensions of maintenance.* Paper presented at the Proceedings of the 2nd international conference on Software engineering.

Teixeira, Leonor, Ferreira, Carlos, & Santos, Beatriz Sousa. (2007). *Using task analysis to improve the requirements elicitation in health information system.* Paper presented at the Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE.

Ting, Chen. (2011). The Control and Measure of Requirements Stability in Software Project *Innovative Computing and Information* (pp. 387-394): Springer.

Zhang, Yan, & Wildemuth, Barbara M. (2009). Qualitative analysis of content. *Applications of social research methods to questions in information and library science*, 308-319.

Appendix I: Coding

	Understanding causes/factors	**Managing & understanding impacts**	**Knowing obstacles**	**Managing and handling requests**	**Using Tools/Methods**
What are the aspects IT health professionals have to deal with for managing requirement changes?	SW development lifecycle New ideas SW technologies & equipment People needs Time to manage Defects Enhancement	Updating product Team environemnt Projects time Outcomes Scope Regulatory Complications and implications	Resources Background Resistance Time to follow Specific process SW approach	Process Reporting Committee Communication means	Usage Adequate Forms Documents Tracability