UNIVERSITY OF GOTHENBURG

Quality Evaluation in transformation of event logs into visual representations

Saeedeh Jadid Tavaf

Master Thesis in Software Engineering and Management

Report No. 2010:066
ISSN: 1651-4769

University of Gothenburg
Department of Applied Information Technology
Gothenburg, Sweden, May 2009

ABSTRACT

Software products play an important role in everyday life of people from daily operations to more critical one. Nowadays, we are witnessing the increasing demands for software quality as a powerful mean to determine the success or failure of a software product. Accordingly, in safety-critical applications the role of the software quality is more highlighted and important. Therefore, during the design and implementation of such software there should be quality control to ensure the certain level of quality for the final product. To that end, applying quality to both the development process and the final product is needed.

The current work put efforts on reviewing related literatures on different concepts like, crisis management, geo-visualization, software product quality and different quality standards. Among different quality standards the appropriate one which is ISO/IEC 9126 is being discussed and applied to a prototype which is developed to ease the processing and monitoring the event logs.

An experiment is designed and ran to evaluate the quality of the developed prototype. In the experiment two different groups of subjects participated and were given same tasks to perform and the result of the report is presented in the current thesis.

Keywords
Crisis Management, Common Operational Picture, Geo-Visualization, ISO/IEC 9126, Software Quality model, Quality Evaluation

Supervisor: Dr. Urban Nulden

Acknowledgement

It would not have been possible to write this master thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here. At this point, I would like to take the opportunity to thank Dr. Urban Nulden for his supervision, guidance, advice and support from the initial to the final level in my thesis.

The good advice and support of Dr. Jonas Landgren has been invaluable on choosing the current topic. I am extremely grateful to him because of all the advice and guidance and all the time and effort he put on helping me.

Above all, I would also like to thank Dr. Miroslaw Staron and Niklas Mellegård for their valuable guidance and support.

Lastly, I offer my best regards and blessings to my parents who supported me in all respect during the completion of my education.

Contents

1. Introduction

Software plays an important role in everyday life of people from daily operations to more critical one. As a result, we are witnessing the high demand for software quality as a powerful mean to determine the success or failure of a software product. (Antonellis, et al., 2009)

Software quality has turn out to be an essential factor in software products. Delivering a software product without an acceptable level of quality is not the goal. Defining the quality of a software product can be challenging itself due to the fact that software quality is an abstract concept that can be understood in different ways based on the context the software is going to be used and different perspectives. However, there have been significant efforts to define and propose quality guidelines and standards. One of the most recent international standards which have been proposed is ISO-9126(ISO/IEC Standard, 1991), Software Product Evaluation Characteristics, which is a high-level model for characterizing software quality.

It is self-evidence that quality is an important factor in all software products and has high impact on the final product. Consequently the importance of the quality in safety-critical software is undeniable. Owing to the fact that crisis management activities play an important role in saving people's lives and there might be high risk on loss of people's life in case of any kind of system failure, the software supporting it can be classified as safety-critical software. As safety-critical software, certain level of quality is expected. For that reason more focus on quality management in development of such system is needed.

In the current work the main focus has been placed on defining and evaluating the quality of safety related software used in crisis management activities by studying similar systems and interviewing domain experts.

Crisis management is more than an incident happening, it is more about a set of activities to manage an incident by different organizations. For that reason, collaboration among different organizations, responders, involved agencies and local and regional authorities is crucial. Therefore, there is a high demand for an interactive tool to support dynamic monitoring of involved agencies as well as to ease the communication and information sharing among them. The role of visual representations in making gained information touchable for involved agencies is vital. For instance, "They need to examine geospatial maps together and collaboratively develop emergency plans and procedures" (Schafer, et al., 2007).

Consequently, having same understanding of a certain situation leads to an efficient response to it. Indeed, in such collaborative activities, which produce huge amount of data it is very important to provide a common understanding between all organizations. Therefore, having Common Operational Picture (COP) is a way which helps different organizations to share needed information and assist them to manage major incidents. "The common operating picture should focus more on what the various actors do in order to provide mechanisms to coordinate the work, not to force one view to be accepted by all actors" (Landgren).

The current work makes an effort to develop an interactive web application supporting collaborative crisis management and evaluate its quality by applying ISO/IEC 9126 standard. The main center of attention of the prototype is to transform the logs into visual representations. To be specific, among different types of data which are produced and used during a crisis, the focus of this thesis is placed on transforming textual event logs into visual representation by visualizing them on a map. To this end a prototype, utilizing positioning data is being developed and the main challenge is to ensure the quality of the prototype. It is worth mentioning that the current work only highlights the focus has been placed on defining quality of the prototype rather than the quality of the process.

1.1 Purpose

The goal of this thesis work is to evaluate the quality of the prototype which supports transformation of event logs into visual representation. The developed prototype offers interactive environment to visualize situations on the map as well as providing functionality to monitor event logs by visualizing, exploring, filtering and geo-tagging them. The main aim of such prototype is to address the lack of computerized techniques to support monitoring event logs.

The main contribution of this thesis work is to address the challenges regarding define quality attributes for such prototype and evaluation of those attributes. To this end, different candidate of quality standards and quality models have been studied and compared.

Much of this thesis work looks at:

- Background study and literature review in the domains of crisis management and software quality management
- Defining a quality model for the prototype
- Defining the most important quality attributes
- Designing and running an experiment to evaluate the selected quality attributes

1.2 Report Outline

After the introduction, section two reviews the relevant literature as the background study for the thesis. Section three is dedicated to describe the research methodology applied to this work. In that section the data collection and data analysis will be reviewed. Section four, prototype specification, takes a brief look at the functional and non-functional requirements. Section five elaborates more on the different phases of the experiment from the design to the running and evaluation. It presents the result of the quality evaluation. Finally I conclude the thesis by giving conclusions.

2. Background

This chapter provides the background knowledge and definitions of underlying concepts to give the reader a brief introduction into the topics that are relevant to current work. These concepts include crisis management and COP, event loges, Geo-visualization, ISO/IEC 9216 standard and quality model.

2.1 Crisis Management and COP

A crisis can be defined as a situation, simple or complicated, in which there are problems need quick responses. When it comes to crisis management, it is not just about an incident that has been occurred, it is also about “a variety of persistent activities such as planning, training, assessment, and organizational change” (Schafer, et al., 2007). Mehrotra et al defines crisis management as “activities that encompass the immediate response to a disaster, recovery efforts, mitigation, and preparedness efforts to reduce the impact of possible future crisis” (Mehrotra et al, 2008).

The Emergency Management Agency (EMA) is responsible to improve the society’s ability to manage serious crises. Although, the EMA doesn’t have operative role during a crisis, however, it is involved to the crisis. Firstly, it supports other public actors by giving them some advice and expert support like communication and management policies. Secondly, EMA monitors the emergency situation during a crisis and updates the government with the latest reports regarding the situation. Finally, EMA takes the responsibility to follow up and evaluate the activities from different actors. It may needs a deep study after crisis, but it helps different actors to learn from the past and apply the gained knowledge to the similar crisis in the future.

In the smaller scale there is regional crisis management that plays important role in emergency preparedness and in increasing society’s ability to manage crisis in a larger scale as regional and national level. One responsibility of regional crisis management is to monitor the activities of involved agencies during and after a crisis. Monitoring the activities of agencies needs information about the incident they respond, the location of the incident and the time of the incident in addition to the involved organizations. This information is also provided by report from the responders and most of them are kept as logs in different systems.

Due to the nature of crisis management, there is huge amount of data coming from heterogeneous sources such as news, retrospective data, other involved actors and so forth (Tomaszewski, et al., 2007) and most of the data are of critical importance. Therefore, keeping track of what is happening and where it is happening is a big challenge. Consequently, it is very difficult for the responders to have a complete overview of a situation. In other words, “a situational overview is very incomplete and mainly in the head of the professionals involved” (Kyng, et al., 2006), means there is no COP among involved individuals. This COP, in fact, is distributed across responders and just exists in their heads.

Having a COP among involved organizations and individual is kind of vision which seems out-of-the-way. There are so many obstacles to achieve the COP; however, there exist technologies which help to overcome these obstacles.

Considering amount of data and different types and resources, in order to manage an emergency, the importance of having common insight should be highlighted. In fact, COP or situational awareness helps to have more efficient response from all agencies involved. In most crises, paper maps are key component to represent this picture. Although, paper maps are used to give an overview of the whole situation area, however, they are just shared by co-located actors. In contrast, digital maps and visualization tools support sharing information among more actors. Using digital maps and visual geographical representations facilitates communication and coordination among distributed agencies. Yet, type of the information should be shared is challenging itself.

2.2 Geo-Visualization and Geo-visual analytics

Geo-visualization, in fact, eases the integration of different approaches like visualization in scientific computing (SiSC), information visualization and geographical information systems (GISysyems) to provide all needed theories, methods and tools to support presentation of geospatial data (MacEachren & Kraak, 2001). Sheppard et al enumerate some benefits of visualization for crisis management as the following:

- To accelerate the access to the information by users,
- Comprehensive visualization through images, for instance maps, to help people who are unfamiliar with the location to obtain a better understanding.

We have been witnessing the increasing emphasis on Geo Visualization tools in supporting crisis management activities in recent crises. Besides, geospatial data is an important component of most crisis management activities (MacEachren & Cai, 2006). Nowadays, we face vast amount of digital data and resources which contain geographical references (MacEachren, et al., 2004). This referencing includes geographic coordinates, addresses, postal code and etc. Besides, this referencing “enables integration of vast stores of diverse information” (MacEachren & Kraak, 2001) based on the geographical location.

Crisis management activities demand analysis of huge amount of data including spatial data sources to assess the situation and make decisions. Such tasks are in need of collaboration among professionals and scientists to take advantage of collection of skills and intelligence (Cai, 2005). Thomas and Cook (2005) in their book “Illuminating the path” give definition of visual analytics as interactive visual interfaces which support analytical reasoning. Moreover, Geovisual analytics supports the collaboration between different actors to take on tasks when the information in incomplete (MacEachren, et al., 2005) and is from heterogeneous sources.

Geo-visual analytics speeds up routine activities and help decision makers to gain insight of huge amount of data in less time by visually presenting the data. In fact, “the goal of visual analytics is to make our way of processing data and information transparent for an analytic discourse” (Keim, et al., 2008).

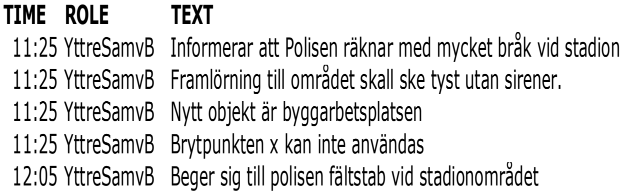
2.3 Logs

Logs are produced by different applications used in different organizations; logs are also produced in the format of reports from responders. It is worth mentioning that the logs are discussed in this work are not those system logs which are kept as archives, but mainly the logs from responders containing information about ongoing or past incidents that can be used for monitorizaion and for improvement of responses to crises.

Traditional approach in managing crises was mainly based on the knowledge of domain experts to keep track of situations and involved actors. But, with increasing in the size of crises and amount of data need for automatic and efficient approaches to monitor and manage crises is increasing. On the other hand, "with advancement in science and Technology, computing systems are becoming increasingly more complex" (Peng et al, 2005) and consequently logs contain more information which is needed to be mined. In fact, they are getting more difficult to monitor, keep track and manage.

One of the important inputs to crisis management is information about previous crises (Tomaszewski, et al., 2007). These data, mostly, are kept as logs in different systems and event log is one example of such information coming from real responses to a crisis. To be specific, event log is a feedback or report from the incident-side commanders and gives detail information about an incident which is helpful for different actors in different ways. For instance, it can be used to assist commanders and decision makers at EOC (Emergency Operation Center) and enables them to get a good insight into a situation and performed actions by providing information about date/time, involved organizations, taken actions and a text containing embedded geographical information in format of the addresses.

Event loges are usually in textual format and exported from different applications in the format of Excel files, MS Word, XML or RSS feeds. Figure 2-3 gives an example of real event log entries (in Swedish) with three columns stating the time, role and a text.



TIME	ROLE	TEXT
11:25	YttreSamvB	Informerar att Polisen räknar med mycket bråk vid stadion
11:25	YttreSamvB	Framlörning till området skall ske tyst utan sirener.
11:25	YttreSamvB	Nytt objekt är byggarbetsplatsen
11:25	YttreSamvB	Brytpunkten x kan inte användas
12:05	YttreSamvB	Beger sig till polisen fältstab vid stadionområdet

Figure 2-3 Example of Event log exported as an Excel file (in Swedish)

The importance of the event logs is that the information can be extracted from and can be used to monitor an ongoing situation or evaluate a finished one. Currently in Sweden most event logs are accessible in textual format and there is no computerized aid to visually present them. Due to the time critical aspect of crisis management, visualizing event logs smoothes the process of monitorization and management of event loges and ease the performing of different tasks like storing, filtering, Geo-tagging and visualizing them on the screen. To be specific, it assists analytical reasoning using spatial analysis by using time and location of the incident and combines them with the involved organization and taken actions. Indeed, visually presenting event logs provides more precise insight besides save more time for monitoring and processing.

Fundamentally, event logs even in textual format have two important dimensions, time and location. Time is generated by system; however, the location is embedded in the text and needs to be processed. Considering everyday improvement in technologies it is not far off to automatically add geographical location in the event logs. From an interview with a domain expert in the Police Activities, maybe years ago it was far from someone's mind to think about automatic time stamp which is currently generated by systems. And nowadays no one worries about it since it is generated by all systems. The same goes for geographical coordinates. In few years, even month, we will have the geographical positions automatically generated and added to the event logs. In that case it will be much easier to visualize event logs automatically and ease the process of monitorization and analytical reasoning.

In view of the fact that currently location is embedded in the text and needs to be processed, the proposed prototype is designed to provide the functionality to manually assign event logs to the specific geographical position on the map.

2.4 Software Product Quality

Software product, according to IEEE definition (IEEE, 1994), is defined as a set of computer programs, documentations and data which are delivered to the user/customer.

When it comes to software development and particularly to quality management the first question to be answered is: “what is quality?” answering this simple yet important question leads to definition of quality and its attributes and on how these attributes affect the quality of product. Meanwhile, there are two general approaches that can be used to ensure software quality, by assuring the quality of the development process or evaluating the quality of the final product. In the present work, although quality of both process and prototype has been considered, however, the main focus is placed on highlighting the quality of prototype rather than the process.

Quality of a software product can be seen from different point of views (Garvin, 1984):

- Transcendental view
- User view
- Manufacturing view
- Product view
- Value-based

Taking into account each of these views leads to the definition of quality, however it leads to different definitions and still is not enough. Indeed, as kitchenham pointed out, the software quality is also defined by the context in which it is used (Kitchenham & Pfleeger, 1996). Therefore, answering the question “What is Software Quality?” is impossible unless we first know who the main stakeholder or audience is and in which context it is going to be used.

Many attempts have been already put on defining software quality, and consequently there are different definitions of it; however, there are two major areas which software quality definitions may contribute to (Hoyer, 2001):

- Conformance to specification
- Meeting customer needs

Conformance to specification is all about addressing the defined specification for particular software, while meeting customer needs emphasizes on addressing customer’s specifications and expectations. Some high-profile quality leaders like Crosby, Deming and others have proposed their definition of quality in respect to the two above areas. For instance, Corsby’s definition of quality addresses the conformance to specification rather than meeting customer needs. Corsby in his book, “Quality is free: the art of making quality certain”, clearly mentioned that “quality must be defined as conformance to specification” (Corsby, 1979).

Deming’s definition of quality falls in the second category which is meeting customer needs. Deming in the book “Out of the crisis: quality, productivity and competitive position” says: “the difficulty in defining quality is to translate future needs of user in to measurable characteristics, so that a product can be designed and turned out to give satisfaction at a price that the user will pay” (Deming, 1988).

According to IEEE the definition of software quality is the degree to which a system, component or process meets specified requirements or customer needs and expectations (IEEE, 1990). Following each of the given definition leads us to cover one of the two mentioned areas.

The International Standard Organization (ISO) describes quality as: “the totality of features and characteristics of a product or service that bear on its ability to satisfy specified or implied needs” (ISO, 1986).

Overall, software quality is not a single property of software, however is contains set of properties which may even conflict with each other. To reach an acceptable quality in software product we may follow a quality standard, and then need to establish quality goal/s and at the end compare the result with our requirements (Schoitsch, 1988).

2.5 The ISO/IEC 9126 Quality Model

ISO and the International Electrical technical Commission (ICE) have proposed the ISO/IEC 9126 Standards for Software Engineering – Product quality (ISO/IEC, 1991, 2001 and 2002) which defines quality model, quality characteristics and related metrics. Currently the ISO/IEC 9126 is one of the most widespread quality standards and in fact its flexibility lets it to be used in many systems including safety-critical systems.

The ISO/IEC 9126 standard has four parts. The first part is quality model that has been defined as "a framework which explains the relationship between different approaches to quality" (ISO/IEC 9126-1, 2001). Additionally, it identifies the characteristics and sub-characteristics of software product. The second part describes the internal quality as "totality of the characteristics of the software product from an internal viewpoint" (ISO/IEC, 9126-2). External quality is the third part of the ISO/IEC 9126 which presents the "totality of characteristics of the software product from an external viewpoint" (ISO/IEC 9126-3, 2002). Finally, the fourth part introduces the quality in use which is representative of the "user's view of the quality of the software product when it is used in a specific environment and a specific context of use".

The first three parts, which described above, deal with describing and measuring the quality of the software product, while the forth part evaluate the product from the user's perspective. Fundamentally, in ISO/IEC 9126 the internal quality has an impact on the external quality and consequently, the external quality has an impact on quality in use. Therefore, achieving the quality in use highly depends on achieving the external quality, which in turn depends on achieving the internal quality of software product (Cheikhi, et al., 2006).

2.5.1 Quality Model

"A quality model is a schema to better explain of our view of quality" (Khosravi & Gueheneuc, 2004). There have been significant attempt to propose quality models and there are different quality models which can be used to define and evaluate software product quality. Among the available quality models, the ISO/IEC 9126 has been chosen and applied in this thesis.

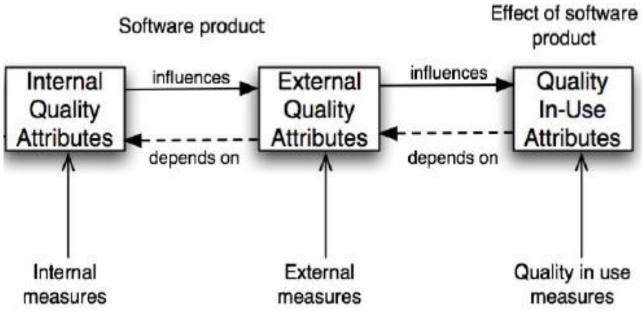


Figure 2-5-1-1 The ISO/IEC 9126 quality model (adapted from (Al-Kilidar, et al., 2005))

The ISO/IEC 9126-1 addresses the definition of software quality model (see figure 2-5-1-1). As it is shown in the figure 2-5-1-1 it consists of two parts: 1) the internal and external quality attributes and 2) the quality in use attributes. The first part which is the main focus of the current work has six quality characteristics (see figure 2-5-1-2) in which each of them has its sub-characteristics. Just to mention, the ISO/IEC 9126 quality model for the quality in use has four characteristics: effectiveness, safety, productivity and satisfaction. In fact, the quality in use as it was previously mentioned relies on the user's view of the product quality when it is running in its operational environment.

The ISO/IEC 9126 standard states that the quality of a software product "can be evaluated by measuring internal attributes (typically static measures of intermediate products), or by measuring external attributes (typically by measuring the behavior of the code when executed), or by measuring quality in use attributes ...[where] appropriate internal attributes of the software are a pre requisite for achieving the required external behavior and the appropriate external behavior is a pre-requisite for achieving quality in use" (ISO/IEC, 2001).

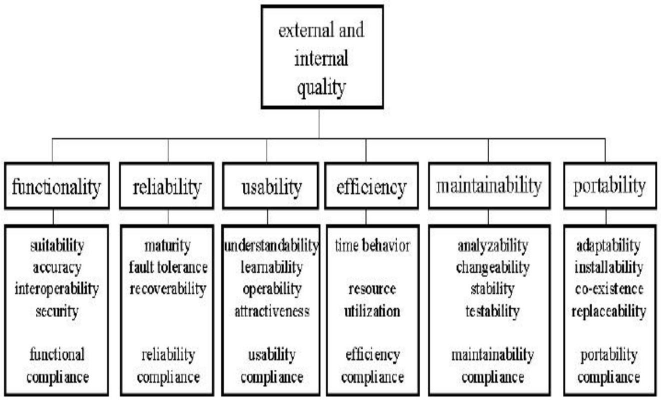


Figure2-5-1-2 ISO/IEC 9126-1 External and Internal Quality Attributes (ISO/IEC, 9126-1)

In the current work the quality of the prototype has been evaluated according to the ISO/IEC 9126 quality model and by measuring the external attributes of the prototype. Therefore, based on the interview with domain experts and literature review on the quality requirements of the same application, number of quality attributes has been selected to measure the quality of the prototype.

2.6 Current Work

Many efforts have been put on literature review containing scientific papers and different books to get insight into the crisis management and current technologies are used to propose means for monitoring event logs. Given that the main purpose of current work is to provide computerized aid for dealing with event logs by the mean of software engineering, the both domains are essential to focus. I realized that there are vast amount of articles and book that are related to my topic in general but not too much on processing event logs. Therefore, I gathered all needed information about logs and their usage by conducting interviews which will be explained more in the research methodology section.

The other area of focus in this work is software engineering and particularly software quality management. This thesis started by requirements specification through interviewing the domain experts and followed by discussing and finalizing the requirements. The current work, like many other similar projects, involved design and implementation of a prototype. Although, the current thesis went through all phases of software development, however, only the quality management part will be highlighted and focused in this report.

3. Research Methodology

This thesis is based on previous empirical studies conducted on the impact of geo-visual analytics on crisis management activities. The research was started by literature review to gain a precise understanding of existing problems in the domain. There are many areas in the need of improvement; however, the scope of this thesis just covers a small part that is processing textual event logs by use of geo-visualization. To that end, the existing ways of processing event logs using current technologies were studied. Furthermore, the possible solutions analyzed and a working prototype was developed to demonstrate the proposed solution.

According to the interviews with domain experts and end-users, currently in Sweden there is no computerized tool for managing and monitoring the textual event logs. In the current systems, the event logs are exported from different application used in different organizations in the format of Excel files as well as Doc and RSS. To process these event logs, users have to manually process them and extract the needed information themselves and again enter data to their systems whenever it is need. One of the users states that "when I want to monitor or edit a situation in our system, if I need some information from an event log I need to print that event log and then try to use the parts is related to my situation and enter the data into our system."

Keeping the current situation in mind and after gaining insight into the domain and existing problem, the next step was to employ software engineering to address the problem. Software engineering is a broad domain. Although the whole phases of development process was followed for design and implementation of the prototype, but I need to narrow it down and scale it to the scope of this thesis, therefore among different areas that could be investigated like architecture design, project management activities and etc the software quality is chosen and discussed in this work. Consequently, studying quality metrics and measurement techniques was another main part of the literature review and also the interviews.

Moreover, retrospective theories definitely helped by giving a comprehensive vision and a set of quality requirements of visual analytics tools. However, in addition to litterateur review and theoretical aspect of geo-visual analytics, this thesis is considerably involved in technical concepts such as implementation of a prototype and evaluation of it. As such, some empirical research methods like observation and interviews with domain experts in both domains crisis management and software engineering employed.

3.1 Data Collection

The main data collection techniques used in the current work was observation and semi-structured interviews as well as document analysis. Among different qualitative research methods although once an observation has been done in the field, however, the personal interview and document analysis were chosen and applied as the main method in.

The observation was done once at the SOS center in Gothenburg. The SOS center is the one which gets emergency calls from the citizens when an incident happens. In addition to the information extracted from incoming calls, they also receive huge amount of data from different organizations. In this case, keeping track of all the incoming data and manage them is a big challenge.

The observation was done in order to see how people act and how they use different application and technology to manage a situation. Each person has four screens to monitor the situation from different perspective, for instance, a map which shows the location and the area of the situation or a screen which shows the latest taken actions by different units and coming data from other organizations and etc. In fact, each screen represents some piece of information in different format and level of detail. Taking pictures of their real systems and data during their job was forbidden. The figure 3-1 shows different screens and running application in which was running just to give us an overview the work space and the technology they already use in managing crises. Actually, it gave a very good perception about various kinds of data and information that are produced and are used during an incident.

The observation gave the vision of how important is to manage and process the data in order to get much out of it and save more time. The main point in this observation was, to understand the importance of an interactive information system to handle that huge amount of data and provide needed information for responders based on the most accurate real time data.

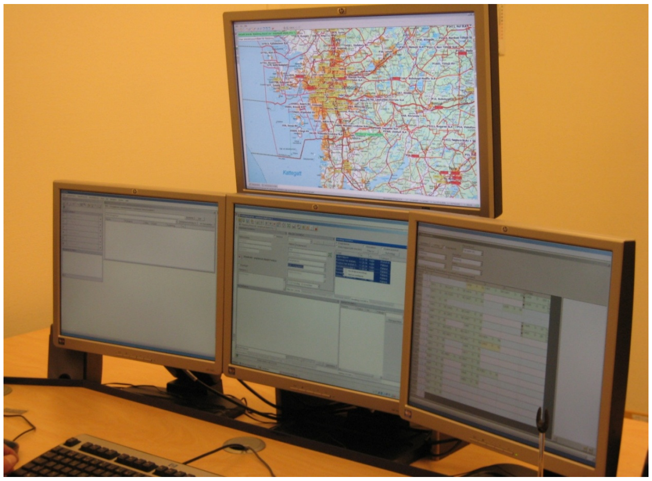


Figure 3-1 shows an example of work space at SOS in Gothenburg

The issue stopped doing observation was lack of Swedish language knowledge. Naturally, people who work together communicates in Swedish, also the incoming calls are in Swedish as well. Therefore, even knowing what is going on and what the situation is could be confusing for me who doesn't know Swedish. On the other hand, the crisis response work is time critical and asking people to explain everything in English during their job is time consuming. In such case, the observation couldn't be a real one or a participant observation. As it was mentioned earlier, even taking picture or video of their real job and applications was forbidden therefore, it wasn't possible to have videos or photos for further analysis. As result, the semi-structured interview with domain experts was chosen as the main method to collect data.

Informal interviews were done with domain experts from different groups and organizations in order to gain insight into the defined problem and also in order to get to know their needs and requirements. In fact, the type of the data to be collected directly depends on the defined research questions and also the unit being interviewed. The different groups being interviewed were experts in diverse organizations like police and fire and rescue.

As mentioned above, interviews were done informally during informal meeting with supervisors and domain experts. In one case the interview was done using Skype with a domain expert from police organization. The outcome of the interviews was field notes which were used for extracting the requirements to be addressed. The notes were also used in designing the prototype and later in writing the thesis report.

Finally, in addition to the observation and interviews, some formal and informal documents about the emergency response work (mainly Swedish emergency response) have been studied. These documents consist of the formal documentation from Swedish Emergency Management Agency in addition to the event logs from incident management systems or from incident site.

In addition, articles, scientific papers and several books in geo-visualization techniques, the ISO/IEC 9126 and software quality management have been also studied.

3.2 Data Analysis

The data gathered at this stage are used to define the prototype specification and the users' requirements. Therefore, the collected data by different techniques has been classified and analyzed. All the notes and photos taken during the observation were reviewed and studied several times. The data collected in the observation was used to study current situation of the information system and technologies used at the SOS center. It gave an initial understanding of the current situation and some ideas about the areas for investigation.

At the same time, the field notes from interviews were studied and classified to be analyzed and processed. In fact, data analysis was a parallel activity during the interviews. Following steps were taken to analyze the field notes from the interviews.

Firstly, the notes were classified in different themes. The main themes are: existing functionalities and their weaknesses, desired requirements and features, user's expectation of the prototype being developed, the most important quality attributes in such application and the priority of those attributes.

After each and every interview the data from that interview were analyzed based on the different defined themes. Classifying the data into different themes helped to see the strength and weakness of the interviews in covering all areas and consequently it leads to the next step to revise the interview questions.

Secondly, the interview questions were revised after each phase in the interview to get more information from the interviewees and cover areas needed to be investigated more.

This was mainly involved the changes in the interview questions and leading the interview into the desired direction.

In sum, the outcome of the data analysis was the specification the user's needs and desires in the format of functional and non-functional requirements. These requirements shape the whole system and its behavior. The next section gives an overview of these outcomes.

4. Prototype Specification

One of the preliminary steps toward developing a software product is requirements specification. By specifying the requirements and user's needs the further steps would be more cleared.

As it was previously mentioned in the research methodology section, the specification of the prototype and its requirements were done by conducting formal and informal interviews by domain experts.

The purpose of specifying requirements is to define the boundaries of the prototype. It also leads to complete and better understanding of overall system and its characteristics as well as stakeholder's expectation of the final prototype. Two different types of requirements are specified: functional and non-functional requirements which are briefly described below.

4.1 Functional requirements

Functional requirements are the functionalities that the prototype is intended to perform. Functional requirements are "statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations" (Sommerville, 2007). The table4-1 bellow presents the functional requirements and brief description of each.

FR-01	**Setup connection with Google Map API**
Importance	Essential
Description	Initialize and make connection to Google map
FR-02	**Select a location/area on map**
Importance	Essential
Description	The prototype shall let the user to select a location/area on the map using provided tools
FR-03	**Draw lines and polygons on the map**
Importance	Desirables
Description	The prototype should let the user to use drawing tools polygon and line to define a space on the map
FR-04	**Define a situation on the map**
Importance	Essential
Description	The prototype shall let user to define a situation on the map and add description to that situation
FR-05	**Store situation on database**
Importance	Essential
Description	The prototype shall store the drawn situation and its description to the database
FR-06	**Import Event Logs**
Importance	Essential
Description	The prototype shall import event log in the format of excel file and store it in database table and shows the content to the user
FR-07	**Geo-tag event log entries on the map**
Importance	Essential
Description	The prototype shall let the user to select each event log entry and assign it to a defined situation on the map and geo-tag that event log entry
FR-08	**Interaction between Map panel and Situation Panel**
Importance	Desirable
Description	When a situation on the map is selected, all the event log entries assigned to that situation should be loaded on situation panel and vice versa.

Table 4 1 Functional Requirements

4.2 Non-functional requirements

Nonfunctional requirements are sometimes categories as usability, performance and security and etc. they are in fact, "constraints on the services or functions offered by the system as timing constraints, constraints on the development process, standard, etc" (Sommerville, 2007). The table 4-1 bellow demonstrates the non-functional requirements assigned for this prototype along with their short description and importance.

N-FR-01	**The prototype should be run in different browsers**
Importance	Essential
Description	The prototype should be able to work with the most popular browsers (e.g. Internet Explorer, Mozilla Firefox, etc.)
N-FR-02	**Functionality**
Importance	Desirable
Description	The prototype should provide the functionality which is supposed to
N-FR-03	**Reliability**
Importance	Desirable
Description	The prototype should provide its services under the defined conditions
N-FR-02	**Usability**
Importance	Essential
Description	The prototype should be easy to use and easy to learn
N-FR-03	**Scalability**
Importance	Essential
Description	The prototype should be scalable. Means a developer should be able to easily add new functionality to the existing prototype.

Table 4-1 Non-functional Requirements

4.3 Application Scenario

This application scenario illustrates how the prototype can be used. It is an example of how using this prototype can ease the processing of event logs.

Considering an incident has been happened and a commander at emergency center wants to get more information about it through this prototype. Either the incident has been finished or is an ongoing one there might be some logs/reports from incident site in the textual format which are sent to the emergency center.

However, based on the incident situation, which is finished or ongoing or new, there are two alternative ways of treating incident on the map. In case of a finished incident the user can just select an existing incident and open it. He/she then needs to import the event log related to the selected incident. Then by choosing the event log entry/s the user can assign them to the area which is defined as the situation area. By this action we assign the event log to the relevant situation.

The second alternative is when the situation is ongoing or new one. In that case the user need to first draw the situation on the map. The drawing tools can be used to draw a line or polygons, depending on the area which the situation has been happened there; and the next step is as defined in the first case.

The commanders or decision makers can just browse the map and track the situation and all the responds which are assigned to it. They are not those who may use the second alternative to define a new situation.

5. Quality Modeling of the Prototype

The first step to model the quality of a software product is identification of that software, i.e. the type of the software and the domain it's going to be used as well as its characteristics. This will contribute to defining the needs of different users interested in that software.

The current work contributes to quality modeling and evaluation of an interactive web application supporting crisis management by applying ISO/IEC 9126 quality standard. The prototype is supposed to be used in the crisis management activities and the intended users are commanders and decision makers at command and control center and management group at County Administration Board in Gothenburg.

5.1 Quality Attributes of the Prototype

Defining the quality attributes in the current work was challenging due to the nature of the context in which the prototype can be used and the type of the prototype itself. Not only defining the right attributes to cover both aspect of the prototype (safety-critical interactive web-application supporting crisis management activities) is important, but also the priority of the attributes should be also taken into consideration. Previous researches in the field (Miyoshi and Auma, 1993) have shown the advantages of keeping the number of key quality factors between three and eight. Choosing too many attributes will end up with conflict between attributes. It also makes the evaluation process more complex and difficult.

Basically, interviews with domain experts and end users were also done to choose and prioritize the quality attributes. Literatures about safety-critical application as well interactive web applications were reviewed. Based on the gained knowledge and by applying the ISO/IEC9126 the quality attributes were chosen and applied.

In the section 2.5 a review of the ISO/IEC 9126 quality model and all the characteristics and its sub characteristics was given. Among the characteristics presented in the ISO/IEC 9126 standard the most relevant factors are selected to apply to the prototype. The table 5-1 shows the correlation between quality attributes and the application domain (both safety-critical and interactive systems are considered). According to the ISO/IEC 9126 the usability and reliability are characteristics which are in turn divided into sub-characteristics, however the security is a sub-characteristic of functionality.

According to the table 5-1, there are three quality attributes that should be considered: Usability, Reliability and Security. In order to ensure if the desired quality has been achieved, we must measure it. There are relevant metrics which can be used to measure these quality attributes. The rest of this section gives definitions of the following selected quality attributes (most descriptions and definitions are taken from the ISO/IEC the 2001 edition):

- Functionality: Are the required functions available in the software?
- Usability: is the software easy to use?
- Reliability: how reliable is the software?

Functionality

"Functionality is expressed as a totality of essential functions that the software product provides" (ISO/IEC, 2001). Functionality is one of the most important and essential factors of any software product. It describes the functions that the system is supposed to perform. It is self evidence that the software functionality is the basis of other characteristics. For instance, we cannot measure usability or reliability of software which doesn't function correctly.

Functionality is defined by its sub-characteristics: Suitability, accurateness, interoperability, compliance and security. However, due to the nature of safety-critical software the security is the most important factor that should be selected. Suitability is another sub-characteristic which is essential to ensure the software functionality.

Quality Factors	**Usability**	**Efficiency**	**Maintainability**	**Portability**	**(Functionality) Security**	**Reliability**
Application domain						
Safety-Critical Systems	**D**	**-**	**-**	**-**	**R**	**R**
Interactive Systems	**R**	**-**	**-**	**-**	**-**	**-**

Table 5-1 Correlation between quality factors and the application domain (http://www.iste.uni-stuttgart.de/se/links/checklists/download/Qu_Factors.html)

Legend:

R: Quality Factor is required

D: Quality factor is desirable, but not required

-: Quality Factor is not relevant

Sub-characteristics:

- *Suitability*
  Definition: Appropriateness (to specification) of the functions of the software (ISO/IEC, 2001).
  Metric: comparing the number of defined requirements with the number of completely implemented requirements to measure the suitability of the prototype

- *Security*
  Definition: relates to unauthorized access to the software functions.
  Metric: by using number of tools as metric the security can be achieved,

Reliability

Reliability is the "capability if the system to maintain its service provision under defined conditions for defined periods of time" (ISO/IEC, 2001). Reliability also has sub-characteristics: Maturity, Fault tolerance and Recoverability.

Sub-characteristics:

- *Recoverability*
  Definition: Ability to bring back a failed system to full operation, including data and network connections.
  Metric: keeping track of all associated defects during the implementation and testing. Then it is possible to check the total number of recoverable defects/ KLOC.

Usability

ISO/IEC 9126 presents five sub-characteristics for usability which three of them are listed here: understandability, learnability and operability. However, the attributes that a product needs to assure usability is highly depend on the nature of its users, the tasks and the environments that the software is going to be used.

The usability of a software product can be defined from different perspective depending on different audience like the end user, managers or the software developers. In the present work the suability is viewed from the end-user's point of view which has more focus on the product. In the 2001 edition of ISO/IEC 9126 usability is defined as "the capability of the software product to be understood, learned and liked by the user, when used under specified conditions."

Sub-characteristics:

- *Understandability*

  Definition: Determines the ease of which the systems functions can be understood, relates to user mental models in Human Computer Interaction methods.

  Metric: to evaluate the understandability an experiment designed and ran with users with different background.

- *Learnability*

  Definition: Learning effort for different users, i.e. novice, expert, casual etc.

  Metric: to evaluate the learnability an experiment designed and ran with users with different background.

- *Operability*

  Definition: Ability of the software to be easily operated by a given user in a given environment.

  Metric: to evaluate the operability an experiment designed and ran with users with different background

6. Quality Evaluation

This section illustrates how an experiment was designed and ran to evaluate the prototype usability. First, the aim of the experiment will be explained and the variables are defined. Then an overview of the subjects and their background will be given.

Experiment tasks and procedure will be explained and finally, the results are discussed.

6.1 Experiment Design

A controlled experiment was conducted to evaluate the usefulness of the prototype. The main goal of the designed experiment is to analyze the user's interaction with the prototype and measure the usability with respect to the user's knowledge about the crisis response domain as well as the user's skills in using such application.

In the experiment different variables are defined. The variables are of two types: independent which can be changed by the experimenter purposely and dependent variable that will change according to the change of the independent variable. I have defined the variables as bellow:

- **Independent Variable:** Usability of the prototype during performing tasks
- **Dependent Variable 1:** How easy is the system to understand and interact?
- **Dependent Variable 2:** How easy the tasks can be done using the prototype?
- **Dependent Variable 3:** how much time the users save when using this prototype to monitor the event logs?

6.2 Subjects

The subjects are selected among different groups of users based on their skills. The subjects in this experiment are divided in the two following groups:

- Duty officers at County Administration Board in Gothenburg
- Students in Software Engineering program

First group are those who have good knowledge of crisis response activities and are involved in these activities. They are familiar using computer system. This group of subjects is domain expert and they use similar applications in their job. However, they don't use such system in daily basis or in each and every crisis which might happen. They have administrative roles and they use the system to monitor the situations and the taken action by different organizations.

The second group is students in software engineering program (Master and Bachelor level). They are selected among the students who have no knowledge about the crisis response work but they are good in using computer systems.

The combination of these two different groups of users will serve to give more detail data to analyze and compare in order to achieve my goal. For instance, if subjects are asked to import an event loge and try to assign it to an existing situation, it might take different amount of time for each group. Performing such task needs to import an Excel file and interact with the map to assign it to a situation. Therefore, the users with different knowledge of crisis response activities and users with different computer skills may spend different amount of time to perform it.

Performing the mentioned task can be seen from two different perspectives in respect to the user's skills. Users with good insight into the task itself (crisis response domain expert) are more likely to figure out how it should be done. Performing the same task by a student who has no idea about the crisis response activities but a good knowledge and computer skills might be done spending different amount of time.

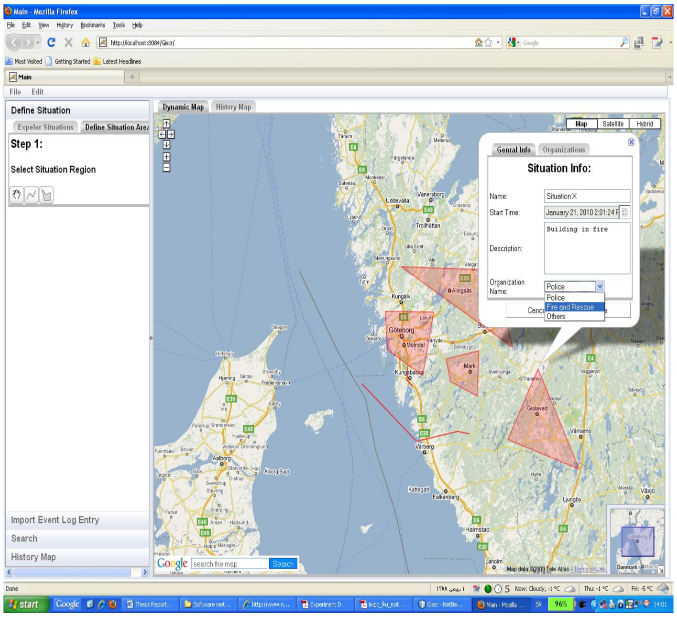
6.3 Experiment Tasks and Procedure

The experiment consists of three phases. Before the participants began their sessions, in the first phase, a brief introduction was given about the prototype and purpose of the experiment. In second phase, participants were asked to complete predefined tasks using the prototype and finally in the third phase participants were asked to talk about their experience using prototype. They were asked to diagnose the problems and give suggestions to improve the prototype usability.

Bellow is the description of the tasks and the procedure the participants are supposed to perform.

Define a Situation: In this task the users are asked to define a situation by drawing it on the map and filling needed information to be saved in the database. This is quite simple task which user needs to choose the right tool either polygon or poly line to draw a situation. It also involves interacting with the map to zoom in/out to find the exact location of the situation.

After locating the situation on the map the user should add the situation name and description and also define the involved organization and save them. Figure 6-3-1 shows a snapshot of the page to perform this task.



Situation Info:

Name: Situation X

Start Time: January 21, 2010 2:01:24 P

Description: Building in fire

Organization Name: Police

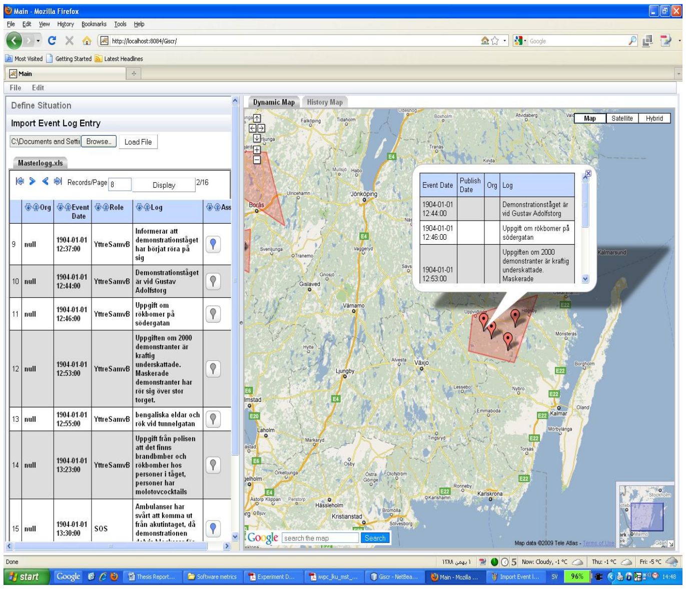
Police
Fire and Rescue
Others

6-3-1: Define a situation on map

Import an Event Log: In this task the user is supposed to import an event log which is an Excel file. The process is quite simple and easy to perform.

Participants are supposed to open the appropriate tab on the left menu and use import wizard to import an event log. Figure 6-3-2 presents the task.

Assign event log entries to a situation: To perform this task the user should have defined a situation first or he/she should have chosen a situation among the existing situations. Then the event loge should be imported. As it is shown in the figure 6-2-2, after importing the event log all the event log entries are listed in the left panel. The user should select one or more event log entries and then assign them to a location in the selected situation area.



	Org	Event Date	Role	Log	Ass
9	null	1904-01-01 12:37:00	YttreSamvB	Informerar att demonstrationståget har börjat röra på sig	
10	null	1904-01-01 12:44:00	YttreSamvB	Demonstrationståget är vid Gustav Adolfstorg	
11	null	1904-01-01 12:46:00	YttreSamvB	Uppgift om rökbomer på södergatan	
12	null	1904-01-01 12:53:00	YttreSamvB	Uppgiften om 2000 demonstranter är kraftig underskattade. Maskerade demonstranter har rör sig över stor torget.	
13	null	1904-01-01 12:55:00	YttreSamvB	bengaliska eldar och rök vid tunnelgatan	
14	null	1904-01-01 13:23:00	YttreSamvB	Uppgift från polisen att det finns brandbmber och rökbomber hos personer i tåget, personer har molotovcocktails	
15	null	1904-01-01 13:30:00	SOS	Ambulanser har svårt att komma ut från akutintaget, då demonstrationen	

Event Date	Publish Date	Org	Log
1904-01-01 12:44:00			Demonstrationståget är vid Gustav Adolfstorg
1904-01-01 12:46:00			Uppgift om rökbomer på södergatan
1904-01-01 12:53:00			Uppgiften om 2000 demonstranter är kraftig underskattade. Maskerade

Figure 6-3-6-1 Import an Event log and assign event log entries to a situation

The key point here is that, although all situations are shown on the map but only one situation at a time is active. Therefore, if the user wants to assign event log entries to different situations, he/she should choose the situation and then assign event log entries to it.

6.4 Measurements

The testing is focused on evaluating the user performance. To that end two different metrics were employed. First one is the time that each user spends to complete a task. The taken time is then used to measure how the tasks can be done more efficient using the prototype.

Second metrics is the number and type of the errors which is made by each participant during performing tasks. This metrics is employed to identify problems with the user interface.

To assist the final measurement, for each participant I recorded the following:

- Total time spent on performing each task including the time for finding the right page or right function

- The exact time spent to complete each task
- The time spent to understand how to perform a particular task

In addition to the recorded time for measuring the efficiency, the following error types were considered:

- Failing to find the right function/page
- Failing to interact with drawing tools (find the right tool to use)
- Other errors

The reason for saving these data is to analyze the real time spent in each task for inexperienced and experienced users. For most users performing a task for the first time may take more time than the second time and more. By keeping the time as mentioned above I am able to analyze how easy is the system to understand and remember to do the same task for the second time.

6.5 Task Analysis

Figure 6-5-1 demonstrated the number of errors users made in doing each task.

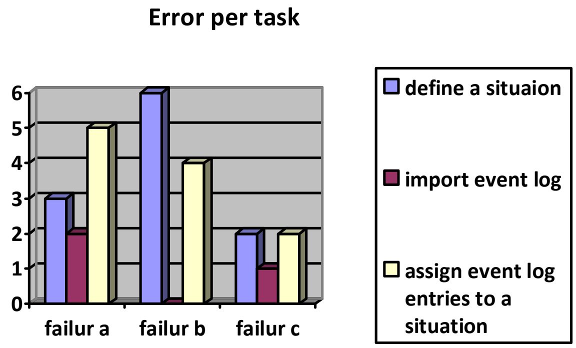


Figure 6.5-1 The figure demonstrates errors per task: (Failure a) Failing to find the right function/page (Failure b) Failing to interact with drawing tools (find the right tool to use) (Failure c) other errors

As it is shown in the figure finding the right page /function was common error in defining a situation. The reason is that to define a situation the user should use drawing tools and it is in a separate tab and most users had difficulties to find it out. The second most common failure was interacting with drawing tools. Apart from choosing the right tool for drawing different types of situations, using the tools was a bit challenging in the beginning.

6.6 Time Analysis

The expected time is calculated by doing the same stuff by me. Since I have done the task so many times and I am the developer of the prototype I do the same task much faster, therefore, by considering some extra seconds for each task (depending on the complexity of the task) I come up with the expected time for each task.

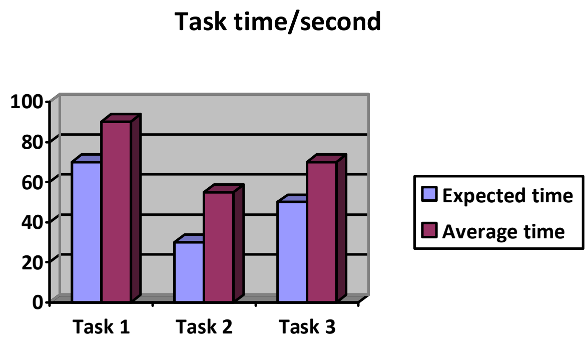


Figure 6.6-1 Time spent per task

As it is demonstrated in the figure 6-6-1 the task time are a bit higher than the expected time. The figure shows that the task one (defining situation) took more time than the other two tasks. Because the Participants spent more time interacting the drawing tools to draw a situation on the map. Importing event logs took less time than other to complete.

6.7 Result of Data Analysis

The observation during the experiment shown that, the most problematic part was drawing a situation on the map. During the experiment some users were mentioned that it is hard using the drawing tool for the first time, however, after couple of times it is easier to use the tools. The second challenging task was to assign event log entries to a situation. Most users made mistakes in performing this task. If the user follows the logical flow from the beginning, defining a situation and importing an event log and then assign it to the situation no mistake was made. The problem occurred when the user wanted to assign the log entries to an existing situation from the past. In this case the user has to change the current situation to the desired one. The observation shown the same task was done faster and easier for the third time and more.

Considering the result of the raw data analysis, I come up with a list of usability issues and suggestions from participants. A usability problem is identified if more than one user finds that problem. However, when one user suggests a feature to improve the usability it was added to the suggestion list and considered if it was relevant. Figure 6.7 gives an example of usability issue. Bellow the list of usability issues and suggestions is given:

Usability Problems:

- Lack of help and user manual:

  User manual or help function is not a requirement of this prototype; however, it is added to this list to be considered as a requirement for future work
- It is hard to distinguish between different types of situation:

  All the situations look the same at first glance. If the user is looking for a specific situation he/she need to search among all exiting situation to find the desired one.
- It is hard to check the situation status on the map:

  It is not possible to distinguish between big or small scale situations by taking a look at the map
- No function to show a past (finished) situation
- Return to the default zoom by a single click: In the current prototype user has to zoom out to come back to the default zoom
- Zoom only in the Gothenburg area when the application starts
- It is hard to put a marker/flack in a sharp position:

Usability Suggestions:

- use a question mark beside each function which includes short description of the function and its usage
- Add different types of flags to distinguish between different type of situation (fire, accident and etc) or even different involved organizations. For instance, by putting a fire flag on a situation it shows that fire and rescue is involved.
- Add different colors to show the status of a situation. For example by putting red color we can stress a situation and present it as a big/important one and by putting yellow or green we can show it is minor one or it has been managed
- By providing a filtering function all the past situation can be shown on the map. The user is also able to define a period of time to filter his/her desired situations.
- Have arrow sign instead of hand sign when assign an event log entries to a situation

The data gathered during the both experiments and interviews with the subjects and domain experts shown that though there was some usability problem, however, the prototype could ease the process of the monitorization of event logs while it reduce the time of processing event logs.

Both groups found the prototype easy to understand and interact. They believed that the prototype is designed in an easy way in which the user can understand the flow of tasks. They also claimed that in the first try there might be some doubt regarding choosing the right page or tool, however, it is not hard to figure out how the tasks should be done.

The duty officers, who are the main audience of this prototype, believed that using this prototype could ease the process of monitorization of event logs. In fact, the prototype provides useful functionalities that they are missing in their current applications. Therefore, integrating these functionalities with the existing applications could reduces the time of processing event logs. It could also reduce the number of mistakes might made by users during the analyzing event logs without using computerized aids.

7. Conclusion

As the need for software products is increasing in all aspect of our daily life, the importance of quality of these products has been also increased. Different quality standards have been proposed to insure the quality of software products. However, different standard can be customized and used according to the type of the software product and the context it can be used.

The current thesis is an effort on quality evaluation of an interactive web application for transformation of logs into visual representations. To ensure the quality of a software product it is essential to ensure the quality of process and the product. In the current work although the quality of both process and prototype has been considered, however, only the quality evaluation of the final product has been presented.

There are challenges in defining and categorizing software product characteristics. The present work basically followed the terminology defined in ISO/IEC 9126 regarding the quality model. In fact, lots of efforts has been put on interviewing domain experts and end user's to come up with accurate characteristics of such product. At the same time a deep literature review was done to study similar products and their characteristics.

Based on the empirical data and by applying the ISO/IEC 9126 quality model, the quality attributes were defined. Accordingly, the most important attributes were chosen to avoid the conflict between them. Usability is one of the quality attributes which was considered. In order to measure the usability two experiments by different groups of participants were run.

First group consists of duty officers at county administrative board and the second group number of students studying software engineering.

The results shown that using the prototype as an additional part of the current applications assists the users to easily monitor and process the event logs while they save more time for interacting with each event log.

Though the developed prototype can be considered as a solution to an existing demand, we believe that it remains some potential for future works and improvements in this area. We believe that using advanced visualization techniques would be a step toward increasing the usefulness of such application in assisting users and decreasing the time spent on monitoring event logs.

Problem #1: ability to zoom on a particular situation on the map, or change the color of the selected situation

Description: usually there is more than one ongoing situation on the map. When a user is about assigning event log entries to a situation he/she needs to select that situation from a drop down list on the left panel. In that case the application will zoom on the selected situation. But, if there were more than one situation on the same location on the map it is hard for the user to recognize the desired situation.

Suggestions: One suggested improvement is to increase the zoom level to just cover the selected situation. It means the map just shows the selected situation with the highest level of zoom which can fit it to the screen. Another suggestion was to change the color of the selected situation, so the user can easily recognize it.

Figure 6.7: Example of a usability problem

8. References

Al-Kilidar, H. Cox, K. Kitchenham, B., (2005), The use and usefulness of the ISO/IEC 9126 quality standard, Sch. Of Comput. Sci. & Eng., New South Wales Univ., Sydney, NSW, Australia

Antonellis P., Antoniou D., Kanellopoulos Y., Makris Ch., Tjortjis Ch., Theodoridis V., Tsirakis N. , (2009), Code4Thought Project: Employing the ISO/IEC-9126 Standard for Software Engineering-Product Quality Assessment, csmr, pp.297-300, 2009 European Conference on Software Maintenance and Reengineering

Cai, G., (2005), Extending Distributed GIS to Support Geo-Collaborative Crisis Management. Geographic Information Science, 11, 1, 4—14

Cheikhi L., Abran A., Buglione L. , (2006), ISBSG Software Project Repository & ISO 9126: An Opportunity for Quality Benchmarking, European Journal for the Informatics Professional, Vol.7, No.1, pp. 46-52 . www.upgrade-cepis.org

Corsby, P.B. , (1979), Quality is free: the art of making quality certain, New York, McGraw-Hill

Deming, W.E. , (1988), Out of the crisis: quality, productivity and completive position, Cambridge Univ. Press

Garvin, D.A., (1984) ,what does “product Quality” really mean? Sloan Management Review 26(1)

Hoyer, R.W. and Hoyer, B.B.Y., (2001), “what is quality”, Quality progress, no.7, pp.52-62

IEEE Standard Glossary of Software Engineering Terminology, (1990), IEEE Std 610.12-1990, Institute of Electrical and Electronic Engineers, Inc., New York, NY, December 10

IEEE, Software Engineering, IEEE Standard Collection, (1994), Institute of Electronical and Electronic Engineers

ISO, (1986), "ISO 8402 Quality Vocabulary," in International Organisation for Standardization. Geneva

ISO/IEC Standard, (1991), ISO-9126 Software Product Evaluation - Quality Characteristics and Guidelines for their Use

ISO/IEC, (2001) , "ISO/IEC 9126-1 Software engineering- Product quality- Part 1: Quality model,"

ISO/IEC, (2002), "ISO/IEC 9126-3 Software engineering -Product quality- part3: Internal metrics,"

ISO/IEC, (2002), "ISO/IEC 9126-2 Software engineering -Product quality- part2: External metrics,"

ISO/IEC, (2002), "ISO/IEC 9126-4 Software engineering -Product quality- part4: Quality in Use metrics,"

Landgren, J., http://jonaslandgren.blogspot.com/2007/03/common-operating-picture.html, Last viewed 2010-05-25

Keim D. , Andrienko G. , Fekete J.-D., Görg C. , Kohlhammer J. , and Melançon G. , (2008), Visual analytics: Definition, process, and challenges,

Khosravi, K., Gueheneuc, Y., (2004), On Issues with Software Quality Models. In: The Proceedings of the 11th Working Conference on Reverse Engineering. GEODES 2004, pp. 172–181. IEEE Computer Society Press, Los Alamitos, Kitchenham, B., Pfleeger, S., (1996), Software quality: the elusive target. IEEE Software 13(1),

Kyng, M., Nielsen, E. T., and Kristensen, M., (2006), Challenges in designing interactive systems for emergency response. In Proceedings of the 6th Conference on Designing interactive Systems

MacEachren A.M. and Kraak M.J., (2001), "Research Challenges in Geovisualization," Cartography and Geographic Information Systems, vol. 28, no. 1,

MacEachren, A. M., Gahegan, M., Pike, W., Brewer, I., Cai, G., Lengerich, E., and Hardisty, F., (2004), Geovisualization for Knowledge Construction and Decision Support. IEEE Comput,

MacEachren, A. M., (2005), Moving geovisualization toward support for group work., in Exploring Geovisuali-zation Dykes, J., MacEachren, A. and Kraak, M. J. (eds.) Elsevier, 445-461,

MacEachren A M, Cai G, (2006), Supporting group work in crisis management: visually mediated human – GIS – human dialogue Environment and Planning B: Planning and Design 33(3) 435 – 456,

Miyoshi, T. and Azuma, M., (1993), An Empirical Study of Evaluating Software Development Environment Quality, IEEE Transactions on Software Engineering, pp. 425-435,
Peng, W., Li, T., and Ma, S., (2005), Mining logs files for data-driven system management. SIGKDD Explor. Newsl. 7, 1

Schoitsch E., (1988), Software safety and software quality assurance in real-time applications. Part 1. Software quality assurance and software safety (concepts and standardization efforts),

Schafer, W. A., Ganoe, C. H., and Carroll, J. M. ,(2007), Supporting Community Emergency Management Planning through a Geocollaboration Software Architecture. Comput. Supported Coop.

Sheppard, Stephan R.J., Blatchford David R., Visual information systems and GIS in crisis management

Sommerville I., 2007 Software Engineering,Chapter 4,Software Processes,page 71, eight ed. Pearson Education Limited,

Thomas, J. J. and Cook, K. A., (2005), Illuminating the Path: The Research and Development Agenda for Visual Analytics, IEEE, Los Alametos, CA.

Tomaszewski, B.M., Robinson, A.C., Weaver, C., Stryker, M. and MacEachren, A.M., (2007), Geovisual Analytics and Crisis Management. In: B. Van de Walle, Burghardt, P.andNieuwenhuis, C. (eds.): In Proc. 4th International Information Systems for Crisis Response and Management (ISCRAM) Conference, Delft, the Netherlands

Appendix-A

This appendix contains the list of characteristics and sub-characteristics of ISO/IEC 9126 quality model which is used and applied to the current work.

Characteristics	Sub characteristics	Definitions
Functionality	Suitability	This is the essential Functionality characteristic and refers to the appropriateness (to specification) of the functions of the software.
	Accurateness	This refers to the correctness of the functions, an ATM may provide a cash dispensing function but is the amount correct?
	Interoperability	A given software component or system does not typically function in isolation. This sub characteristic concerns the ability of a software component to interact with other components or systems.
	Compliance	Where appropriate certain industry (or government) laws and guidelines need to be complied with, i.e. SOX. This sub characteristic addresses the compliant capability of software.
	Security	This sub characteristic relates to unauthorized access to the software functions.
Reliability	Maturity	This sub characteristic concerns frequency of failure of the software.
	Fault tolerance	The ability of software to withstand (and recover) from component, or environmental, failure.
	Recoverability	Ability to bring back a failed system to full operation, including data and network connections.
Usability	Understandability	Determines the ease of which the systems functions can be understood, relates to user mental models in Human Computer Interaction methods.
	Learnability	Learning effort for different users, i.e. novice, expert, casual etc.
	Operability	Ability of the software to be easily operated by a given user in a given environment.
Efficiency	Time behavior	Characterizes response times for a given thru put, i.e. transaction rate.
	Resource behavior	Characterizes resources used, i.e. memory, cpu, disk and network usage.
Maintainability	Analyzability	Characterizes the ability to identify the root cause of a failure within the software.
	Changeability	Characterizes the amount of effort to change a system.
	Stability	Characterizes the sensitivity to change of a given system that is the

Characteristics	**Sub characteristics**	**Definitions**
		negative impact that may be caused by system changes.
	Testability	Characterizes the effort needed to verify (test) a system change.
Portability	Adaptability	Characterizes the ability of the system to change to new specifications or operating environments.
	Installability	Characterizes the effort required to install the software.
	Conformance	Similar to compliance for functionality, but this characteristic relates to portability. One example would be Open SQL conformance which relates to portability of database used.
	Replaceability	Characterizes the *plug and play* aspect of software components, that is how easy is it to exchange a given software component within a specified environment.