IMPLEMENTATION OF END-TO-END INCIDENT MANAGEMENT PROCESSES IN A UNIFIED IMS

Ásgeir Jónasson
Master of Science
Computer Science
June 2015
School of Computer Science
Reykjavík University

M.Sc. PROJECT REPORT

ISSN 1670-8539
Implementation of end-to-end incident management processes in a unified IMS

by

Ásgeir Jónasson

Project report submitted to the School of Computer Science at Reykjavík University in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

June 2015

Project Report Committee:

Hannes Högni Vilhjálmssson, Supervisor
Associate Professor, Reykjavik University

Marta Kristín Lárusdóttir
Assistant Professor, Reykjavik University

Guðbrandur Örn Arnarson, MBA
SAR coordinator, Icelandic Association for Search and Rescue
Implementation of end-to-end incident management processes in a unified IMS

Ásgeir Jónasson

June 2015

Abstract

Information technology plays a key role during large scale search and rescue operations, such as those carried out by the Icelandic Association for Search and Rescue (ICE-SAR). Systems used by incident managers must support their processes and actions, and enable fast and efficient information gathering and distribution.

In this project, two novel additions to the incident management system used by ICE-SAR and the Civil Protection in Iceland, were developed.

The first is a new system called Preparedness, which aids in the collaborative creation and management of preparedness plans. Through integration with an incident management system, Preparedness furthermore allows incident managers to initiate a preparedness plan within the same system as is used to carry it out, increasing speed and accuracy in the plan’s execution.

The second is a search area and evacuation management component, an addition to ICE-SAR’s and the Civil Protection’s incident management system. Creating and managing search areas and evacuation zones is now, for the first time, possible from within the system. This allows for a connection between areas, and the tasks, groups and individuals within the incident management system. This enables a plethora of new features around search area and evacuation management to be implemented.

These two additions bring an already successful incident management system to a new level, and make it a complete solution for incident management.
Verkferlar viðbragðsáætila sameinaðir í einni heildstæðri hugbúnaðarlausn

Ásgeir Jónasson

Júní 2015

Útdráttur

Upplýsingatækni spilar lykilhlutverk í stórum aðgerðum í leit og björgun, líkt og þeim sem að Slysavarnarfélagið Landsbjörg sinnir. Sá hugbúnaður sem aðgerðastjörnendur nota þarf að styðja við verkferla þeirra, og ýta undir hraða og skilvirka öflun og dreifingu upplýsinga.

Verkefnið felst í að þróa tvær nýjar viðbætur við aðgerðastjórnunarkerfi Lands-bjargar og Almannavarna Ríkislögreglustjóra.

Fyrsta viðbótin er nýtt kerfi er kallast Viðbragð, sem verður notað við samvinnu um gerð og umsjón viðbragðsáætlana. Í gegnum tengingu við aðgerðastjórnunarkerfið, býður Viðbragð ennfremur upp á að ræsa viðbragðsáætlun innan þess kerfis sem notað er til að framkvæma hana, sem eykur hraða og nákvæmi í framkvæmd áætlunarinnar.


Þessar tvær viðbætur færa aðgerðastjórnunarkerfi sem þegar hefur sannað sig vel, upp í nýjar hædir, og gerir það að heildstæðri hugbúnaðarlausn fyrir aðgerðastjórnun.
Tileinkað Danna bróður
Án þín sem fyrirmyndar hefði þetta verkefni aldrei orðið til
Acknowledgements

I want to thank the following people for supporting me throughout my M.Sc studies and the project:

Steinunn Kristín Jóhannsdóttir for supporting me, in all areas of my life, for the entire duration of this project.

Jónas Guðmundsson, Harpa Ásgeirsdóttir, Daníel Páll Jónasson and Heiðrún Dóra Jónas-dóttir, for moral support, and keeping me dedicated throughout.

Hannes Högni Vilhjálmsson for his excellent guidance, support and fantastic and inspirational ideas.

Guðbrandur Órn Arnarson for the opportunity to work for SAReye from the start, and his excellent vision in past, present and future projects.

Björn Þór Jónsson and the Computer Science office for the Alan Turing grant, and for the support throughout my studies.

Sigrún María Ammendrup and Kaelene Elizabeth Marie Spence for being there whenever I had any questions.

The Icelandic Association for Search and Rescue and the Civil Protection in Iceland, for believing in me and the rest of the team that worked on the first SAReye IMS.

Kári Tristan Helgason, Ingólfr Eðvarðsson and Trausti Sæmundsson, for the collaboration during the summer of 2013.
# Contents

List of Tables xii

1 Introduction 1

2 Background 5

2.1 The beginning of SAREye 5

2.2 Original problem domain: Large scale search operations and their volunteer needs 5

2.3 A new beginning: The SAREye IMS 7

2.4 SÁBF: Foundation of the SAREye IMS 8

2.5 Implementation of SÁBF in the SAREye IMS 10

2.5.1 Framgangur (e. Progress) 11

2.5.2 Skjáir (e. Screens) 13

2.5.3 Lotur (e. Rounds) 13

2.5.4 Summary and non SÁBF related features 14

3 Preparedness 17

3.1 Background 17

3.1.1 Structure and content of preparedness plans 17

3.1.2 How preparedness plans are written 19

3.1.3 Problems with the writing 19

3.1.4 Potential solutions to the problems 20

3.1.5 Thesis contribution 21

3.2 The technology stack 22

3.3 General design principles 23

3.4 Writing a preparedness plan 24

3.4.1 Sign in and registration 24

3.4.2 Permissions 25

3.4.3 Creating a preparedness plan 26
3.4.4 Adding editors and authors ................................................. 27
3.4.5 Writing a chapter .......................................................... 29
3.4.6 Viewing chapter versions and changes made by other authors. ... 30
3.4.7 Defining tasks and communications methods ...................... 34
3.4.8 Publishing a preparedness plan ........................................ 36
3.4.9 Examples from the PDF .................................................... 41
3.4.10 The preparedness plan’s version history in the changelog ...... 43
3.4.11 Viewing recent changes to preparedness plans .................... 43
3.5 Preparedness: A tool for emergency responders ....................... 44
3.6 Preparedness: The integration with the SAReye IMS ............... 45
3.7 Evaluation ................................................................. 49
  3.7.1 Users ................................................................. 49
  3.7.2 Environment and duration ........................................... 49
  3.7.3 Methods ............................................................. 49
  3.7.4 Data gathered ....................................................... 49
  3.7.5 Measurements ....................................................... 50
  3.7.6 The steps performed by users ..................................... 50
  3.7.7 Discussion with Eggert Magnússon ............................... 51
  3.7.8 Discussion with Víðir Reynisson ................................. 52
  3.7.9 Results and possible improvements ................................ 53
  3.7.10 Changes to make .................................................. 53
4 Search area and evacuation management .................................. 55
  4.1 Background .................................................................. 55
    4.1.1 Thesis contribution .................................................. 56
  4.2 Overview ..................................................................... 56
  4.3 Map and tile layers ...................................................... 57
  4.4 Drawing areas ............................................................ 58
    4.4.1 Hand drawing areas ............................................... 58
    4.4.2 Hand drawing hasty search ...................................... 59
    4.4.3 Drawing using KML and GPX .................................... 61
    4.4.4 Code for the four methods of drawing ....................... 62
  4.5 Probability circles ....................................................... 64
  4.6 Connection between tasks and areas ................................ 67
    4.6.1 Task names and area identifiers ............................... 67
    4.6.2 Associated tasks and areas and their statuses .............. 69
  4.7 Statistics ..................................................................... 69
  4.8 Calculating resource needs ............................................. 72
4.9 Instructions ................................................................. 72
4.10 Personalized mobile view ............................................ 73
4.11 Evaluation ................................................................. 76
   4.11.1 Method .............................................................. 76
   4.11.2 Timeline of the development and evaluation ......................... 76
   4.11.3 Summary ............................................................ 78

5 Conclusions 81
  5.1 Lessons learned ....................................................... 81
    5.1.1 Personal lessons learned ......................................... 81
    5.1.2 Technical lessons learned ....................................... 81
    5.1.3 Success of thesis contributions .................................. 82
    5.1.4 Future work ....................................................... 82
    5.1.5 Final words ...................................................... 83

Bibliography 85

A 87
List of Tables

A.1 Results from the Preparedness usability testing 87
Chapter 1

Introduction

When it comes to search and rescue, speed is key. So is accuracy, efficiency, cooperation, experience, planning, and knowledge. ICE-SAR, the Icelandic Association for Search And Rescue, and the Civil Protection in Iceland, rely heavily on information technology to facilitate these key dynamics in their operations.

Because of the vast and variable user base of ICE-SAR’s and the Civil Protection’s systems, they use must be easy to use. Because of the complexity and importance of the work their incident managers do, the systems must facilitate accurate and comprehensive information gathering and distribution. Because of the sheer number of volunteers that come into play during critical rescue operations that may span multiple days, the systems must support the cooperation of hundreds of people. For preparedness sake, the systems must facilitate foreseeing future disasters, and in the unfortunate event that they do occur, must enable plans that have been created to speed up the execution of critical tasks within operations. Because the field of search and rescue is continuously improving, and rescuers are always looking to better themselves (they are very self-criticizing), these systems must facilitate learning from mistakes and successes. Systems should therefore keep track of as much information as possible about when decisions were made, who made them, and when incidents critical to operations took place, and when they were logged by incident managers.

Some of the aforementioned IT needs of ICE-SAR and the Civil Protection have been met in the past, to a varying degree, and some not at all. SAReye, the company behind the solutions in this report, was formed to create a solution to one especially tough to solve operational issue. The yet unsolved issue is one that becomes apparent in large scale search operations, when thousands of volunteers are needed. The sheer vastness of the search areas makes it almost impossible for them to be covered in a few hours. An idea for
a solution was to use a combination of areal photography with drones, image processing and crowdsourcing to solve this problem. That solution however, is mostly beyond the scope of this report, but since it is the starting point of SAReye’s journey, it will get a high level overview in the background chapter.

As will be covered in this report, we do believe we have met many of the unmet needs of ICE-SAR and the Civil Protection, and largely improved the solutions to needs met in the past, both through previous work, discussed in the background chapter, and this thesis’ contributions.

When SAReye was formed, ICE-SAR were on the third iteration of their Incident Management System (hereafter IMS), called Aðgerðagrunnur. This system had its merits, but a number of problems plagued it. As Guðbrandur Örn Arnason, the CEO of SAReye and SAR coordinator at ICE-SAR, stated in May 2013:

It [the IMS] is painfully slow, non-extensible, virtually immutable, expensive to maintain and not particularly user friendly. Furthermore, the people at ICE-SAR feel that those working on previous iterations have never quite understood their needs.

In the summer of 2013, a team capable of making an IMS system that met all of ICE-SAR’s needs came together. SAReye had contacted me, and I formed the team consisting of top students from Reykjavík University. We had a team of developers all well versed in current web development technology, all interested in and passionate about ICE-SAR (I have myself been saved by ICE-SAR’s volunteers from a life-threatening situation), and very dedicated to meeting all their needs.

Perhaps more importantly, Guðbrandur Örn is an incident manager at ICE-SAR. He became the product owner of the SAReye IMS. We therefore had expert knowledge on both ends, the technology and the problem domain. The feedback loop during development was therefore very short, and we could be sure that our implementation was fulfilling ICE-SAR’s needs.

The result was a new IMS system for ICE-SAR being launched in the beginning of November 2013. The system had been tested by ICE-SAR with operations being run in parallel in the old IMS and new SAReye IMS. After a few operations, the old system was put into storage mode and the new SAReye IMS took over. Since then, over 2400 of ICE-SAR’s operations have been executed with the aid of the SAReye IMS. It has been very well received, and since 2013, the Civil Protection in Iceland, the Icelandic Red Cross, and Landsnet, a company that operates Iceland’s electricity transmission grid, have begun using their respective instances of the SAReye IMS.
We did not, however, stop there. Our vision is to integrate as many of the information technology needs of incident managers as possible into a single system, to increase efficiency and accuracy in their work. This is where my M.Sc project comes into play.

In this report, we will first take a detailed look at a new system I have developed as part of this thesis, called Preparedness. Preparedness was designed for the Civil Protection, and integrates with their instance of the SAReye IMS. Furthermore, we will go through a completely new system component of the SAReye IMS. The component I have designed and implemented is called Search area and evacuation management, that aids ICE-SAR in search operations, and the Civil Protection in evacuations and disaster management.

Preparedness is a tool to aid in the creation, maintenance and release of preparedness plans. A preparedness plan is a document that describes a disaster that may strike at any time, and how different organizations should respond in the unfortunate event that it does. Disasters described and planned for can be, for example, plane crashes, forest fires, eruptions and earthquakes. The plans are rather lengthy, about 45 to 125 pages long, and include chapters on, among other things, topography, management hierarchy, communication, and tasks that need to be carried out.

Preparedness facilitates management of authors and editors and acts as a version control both for smaller changes to individual plans and entire versions of preparedness plans. Most importantly though, it integrates with the SAReye IMS and enables incident managers to use preparedness plans as a template for operations, something that has not been done before. For those that the plans concern, it significantly speeds up the process of looking up what to do during disasters, information that, until now, has been ‘hidden’ inside these lengthy preparedness plans that are not always readily available. Before Preparedness was created, responders needed to find a version of the relevant preparedness plan, find the chapter that described what their organization’s role was during the disaster and then find their specific task. During a disaster, this is considered as a too time consuming process, especially if the responders need to download and search through a 45 to 125 page document.

Search area and evacuation management is a new component within the SAReye IMS. Before this component came into existence, incident managers used other software solutions to create areas on maps (for evacuation or search). The result from their work had no connection to a system that is actually used to manage work around the areas, such as task assignment and completion, or assigning responders to the areas.

When areas are drawn on maps in the Search area and evacuation management component of the SAReye IMS, tasks are automatically created for each of them. These tasks
can have groups assigned to them, and they can be marked as completed or in progress within the *SAReye IMS*. This allows incident managers to observe, on maps, the progress of searches or evacuations as tasks are assigned and completed. Useful information, such as sizes of areas, sizes of searched and unsearched areas, workload of individual rescuers, probability circles, and more, is readily available within the *SAReye IMS*.

We will begin this report by covering the background, the work that had been done at *SAReye* before this project started. This includes the idea *SAReye* was formed to realize, and the main functionalities of the *SAReye IMS* system. After that we will, in chapters 3 and 4, discuss the backgrounds, implementations, and evaluations of both thesis contributions, the new *Preparedness* system and the new component in the *SAReye IMS*, the *Search area and evacuation management*. At the end, we will review the progress so far, and discuss the potential for future work.
Chapter 2

Background

2.1 The beginning of SAReye

SAReye started working on solutions for incident managers in the year 2013. It all began with SAReye winning the Icelandic entrepreneurship contest Gulleggið (e. Golden egg) ¹, for the proposed solution to an especially hard-to-solve operational issue, described below. Shortly after, proposing the same solution, SAReye got accepted into Startup-Reykjavik, a 10 week, mentorship driven seed stage investment program, held by Arion Bank, Innovit and Klak ².

2.2 Original problem domain: Large scale search operations and their volunteer needs

The problem that SAReye aimed to solve originally becomes apparent in large scale search operations. One of the first steps taken during such operations is determining the so called probability circles. The probability circles are centered around the initial planning point (IPP), and are determined using statistical information from the ISRID database, and academic analysis by Robert J. Koester ³. The circles are drawn for 25%, 50%, 75% and 95% probabilities, along with a 300 meter circle around the IPP. The 25% circle, for

¹ Gulleggið is hosted by Klak Innovit ehf. Gulleggið website http://www.gulleggid.is.
example, represents the area within which 25% of those found were located (Sigurðsson, 2007, p. 38-44).

More often than not, the 95% circle is extremely large. When the circle’s radius is around 18 kilometers, the search area spans 1000 square kilometers. To search this area in 8 hours takes 2,495 volunteers. These searches are not a once in a blue moon occurrence, and obviously, in a country with a population of just over 320,000 people, getting 2,495 volunteers is hard. It would furthermore be preferable to be able to search the area more than once, and no less importantly, to be able do so in less than 8 hours.

Figure 2.1: *Probability circles from an ICE-SAR search operation, north of Mýrdalsjökull. The 95% circle has a radius of 19.5 kilometers, and as can be seen, it was not possible to plan search areas for the entire circle, so the areas must be prioritized.*

The solution **SAReye** proposed was to use a combination of areal photography, crowd sourcing and automated image analysis to solve this problem. The idea was, and is, that drones fly over and photograph the entire 95% circle. The images are then sent through an automated image analysis that searches for any type of anomalies in the images, and they are then prioritized according to the findings. The images are then analysed further through crowd sourced efforts. The images most likely to contain clues or people, become the first ones put through further human analysis. Any interested individual would be able to participate in these searches by going through and marking these images. Those images marked as of interest, would then be sent to incident managers for further examination.

The **SAReye** team put some effort into realizing this idea, mostly during the 10 week session at StartupReykjavik. The idea however, was eventually put on hold. There are technical issues, mainly due to the fact that in many of operations in which this technology is needed, the weather conditions are awful. This causes problems for smaller drones, that

---

4 Example provided by Guðbrandur Órn Arnarson, and is calculated using ICE-SAR’s calculator for volunteer needs.
already have a limited flight range. There are also inherent problems with some types of
terrain, in which spotting clues can be difficult or impossible. These are likely manageable
or even solvable problems (better drones, infrared cameras), but the main reason for the
change of plans was that there were more pressing IT needs at ICE-SAR that SAReye
wanted to tend to. These needs were relevant to all operations at ICE-SAR. This original
idea, while still relevant today and having the potential to save lives, usually only comes
into play in a handful of operations a year. It will have to wait for further financing, and
for the more affordable technology to catch up with SAReye’s ideas.

2.3 A new beginning: The SAReye IMS

In 2013, more pressing, and more manageable software related problems at ICE-SAR be-
came apparent, than that of bettering search methods during large scale search operations.

ICE-SAR discussed with Guðbrandur Örn the possibility of SAReye making a new ver-
sion of their IMS, called Aðgerðagrunnur. A team was formed around building the new
SAReye IMS 5, while other members of SAReye would participate in StartupReykjavik for
10 weeks, working on the marketing for the SAReye IMS and the crowd sourcing idea,
the requirements analysis for the SAReye IMS and a prototype for the crowd-sourcing and
image analysis. Towards the end of the 10 week session, the focus moved entirely onto
the SAReye IMS.

After ten weeks of programming, and having the new IMS tested with operations being
run in parallel in the old and new system, the SAReye IMS was launched and became active
in the beginning of November 2013. Since then, over 2400 of ICE-SAR’s operations have
been executed with the aid of the SAReye IMS.

One might ask why ICE-SAR did not simply try to use an out of the box software solution
for their needs. There are several reasons for this. The first being that their previous
version was just that. It was originally an out of the box solution, albeit modified for their
needs, made by an American software company. Modifications where very expensive, the
software was becoming very dated in terms of web application technology and incident
managers at ICE-SAR had never felt that the developers understood needs. The system
was also not particularly easy to use, and painfully slow. Getting SAReye to make them
a new version was the perfect opportunity to have a group of developers implement the
software, where a team member and product owner, is an incident manager at ICE-SAR.

5 This team, as mentioned in the introduction, was composed of students from Reykjavík University.
In addition to myself, the team members where Kári Tristan Helgason, Ingólfur Eðvarðsson and Trausti
Sæmundsson, with Guðbrandur Örn guiding the implementation process as a Product Owner.
There was no reason to worry about miscommunication, or misunderstandings when it came to ICE-SAR’s needs. Furthermore, SAReye being placed in Iceland had the potential to make the feedback loop much tighter.

According to Guðbrandur Örn, using an out of the box solution unchanged was, as well, never an option. There were issues related to the structure of a potential system, which had to be tailored specifically to the operational management framework used at ICE-SAR (the SÁBF framework, discussed in section 2.4). There is a need for Icelandic weather forecasts, and reporting specific to ICE-SAR. Most importantly though, the system had to sync with ICE-SAR’s database of volunteers, rescue units and vehicles, and maintain relationships among volunteers and vehicles and their units. This database is quite complex, and the system had to facilitate storing vast amounts of information about each of the around 4200 volunteers and 300 vehicles. No out of the box software solution was ever going to facilitate these needs, so ICE-SAR were always going to need a custom solution, and SAReye had the best team for the job.

2.4 SÁBF: Foundation of the SAReye IMS

The SAReye IMS is built upon a strong foundation. It is designed to facilitate the operational management framework SÁBF, used by ICE-SAR, the Civil Protection in Iceland, the Icelandic Red Cross and most other response teams in Iceland.

SÁBF is based upon the Incident Command System (ICS), and the National Incident Management System (NIMS). FEMA, one of the agencies that uses ICS and NIMS, defines ICS thusly:

*The Incident Command System is a management system designed to enable effective and efficient domestic incident management by integrating a combination of facilities, equipment, personnel, procedures, and communications operating within a common organizational structure. ICS is normally structured to facilitate activities in five major functional areas: command, operations, planning, logistics, Intelligence & Investigations, finance and administration (FEMA, 2008).*

ICS is a framework that is constantly being improved. Post-mortem learning from the 9/11 events, led to the creation of NIMS, as an add-on framework on top of ICS.

Information about NIMS, the improvement process of the ICS, and their connection to the background of the SAReye IMS, was supplied by Guðbrandur Órn Arnarson.
To give the SAReye IMS context and explain why it was designed as it is, an explanation of the structure of SÁBF is in order.

SÁBF is a unified management framework for all emergency response teams in Iceland, and the acronym stands for Stjórnun (e. Command), Áætlanir (e. Planning), Bjargir (e. Resources), and Framkvæmdir (e. Execution). Due to the high scalability of the SÁBF framework, it can be used for planning and executing any operation, regardless of its size or nature (Almannavarnadeild, 2010). An incident can escalate from one incident manager to thousands of responders, within the framework, and then back down to one incident manager. The basic idea is that every operation has the same units of work, and that each person taking part in the operation falls into one unit at a time. Therefore, each person knows exactly what his or her responsibilities are. Guðbrandur Örn Arnarson helped in compiling the following definition of the SÁBF framework:

- **Stjórnun (e. Command):** The tasks at hand must be managed, overseen and their completion ensured. Operational managers in regional command are first and foremost leaders, are responsible for making sure that others carry out their tasks, and should be able to delegate responsibility. Managers must oversee operations and ensure that work is done professionally and according to plans.

- **Áætlanir (e. Planning):** The role of those taking part in planning, is to plan for how to solve the tasks at hand. In search operations, the role of planning is twofold. First, there is information gathering and research and then there is search area planning. In search operations, information gathering and research involves taking interviews with relatives and family members, gathering information about the search area, weather conditions, how knowledgeable or prepared those lost are, and the last known usage of their credit cards and mobile phones. The information gathered is taken into consideration when organizing the search effort.

- **Bjargir (e. Resources):** The role of those in charge of resources, is to ensure that there is enough of them; be it vehicles, tools, manpower, food, facilities or anything else needed. Those assigned to resources must think ahead, as volunteers must be replaced when they rest in between tasks.

- **Framkvæmdir (e. Execution):** The plans and resources must be utilized to carry out the tasks at hand. Resources are assigned to groups, which are in turn assigned one or more tasks. Volunteers are given reports about the operation and the tasks at hand. After task completion the volunteers are debriefed.

---

7 Explanation of the scalable nature of SÁBF supplied by Guðbrandur Örn Arnarson
The exact tasks carried out by each work unit may vary between operations, depending on their nature. The high level definitions of their responsibilities however, stay the same.

2.5 Implementation of SÁBF in the SAREye IMS

Now that we have covered the foundations of the structure of the SAREye IMS, we can delve into the implementation, and take a look at how SÁBF is facilitated in the solution. We will use the search for Nathan Foley Mendelssohn as an example operation, displayed in figure 2.3. Note that although the IMS is in Icelandic, all images from the system have headings and headers in English.

The front page shows all operations since the beginning of the week before. Operations can also be marked as ‘stuck’ to the front page, which ensures that they are displayed there at all times.

![Figure 2.2: The front page of the SAREye IMS on March 20th, 2015.](Image)

By glancing over the front page, incident managers can gain insight into ongoing and recently completed operations. During exceptionally bad storms in Iceland, for example, there can be around a dozen ongoing operations, and incident managers may need to consult the front page to gain a better overview of the situation as a whole.

An operation’s name very briefly describes what has happened and what needs to be done. The operation’s severity, number of sufferers, and information on active and total number of volunteers quickly indicate its scope. Operations that have been finished have two timestamps, so one can quickly see which ones are finished and which are not. Finally,
we want most of the operations to be associated with at least one map coordinate. Map coordinates are assigned to events and tasks (described in the next section), and incident managers can quickly see which operations have yet to have coordinates assigned to them.

In the following sections we will take a look at the other main system components in the SARey IM: Framgangur (e. Progress), Lotur (e. Rounds) and Skjáir (e. Screen). A quick overview of other smaller components will also be given at the end.

2.5.1 Framgangur (e. Progress)

When an operation is clicked, we move on to Framgangur (e. Progress), shown here in figure 2.3. Framgangur is used both to gain an overview of an operation, and to aid in its execution.

![Figure 2.3: Framgangur view for the search operation for Nathan Foley Mendelssohn.](image)

A multitude of different events take place throughout the execution of an operation. Events need to be logged, not only for overview purposes, but also to keep track of decisions made, to enable lessons to be learned from the experience. In our example here,
we can see events indicating volunteers leaving for duty, a volunteer being in an accident, and a file with a GPS track. Events can also contain information about where the person was last seen or some other new information about the lost person. Events can be marked as important (highlighted), private (only visible to admins), or as one of the operation’s important events, that is, on the operation’s timeline.

*Groups* are containers around resources, both volunteers and vehicles. In large scale search operations such as this one, volunteers from rescue units from all over the country participate in the search. These volunteers are assigned to groups, which are in turn assigned to tasks. Incident managers can see how many groups are available for assignment, out of the total number of groups. Furthermore, for each group, they can see how many active resources there are, and what the group’s status is; *Available for task assignment, Working, Not available/resting or Inactive*. In the box to the right, titled Resources, incident managers can see a summary of how many resources there have been, and how many are active now. Each of the volunteers should have seen at least the *basic information*, which sums up what they need to know about the operation. In search operations such as this one, this can be basic information about the lost person, when he or she became missing, where they are from, what may have happened, and how they are dressed.

*Tasks* for searching specific areas, or performing other duties, start rolling in as the operation is executed. Each task will have one or more of the *Groups* assigned to it, and each *Group* will be assigned one or more of the tasks. For each task, we can see the number of active volunteers in the groups assigned to it, and the task’s status; *Finished, In progress or Not in progress*. Finally, we can see each task’s priority, which is important, as search areas can be of varying importance. These priorities can be quickly changed using the up and down arrows.

For a visual and geographical overview of the ongoing tasks and logged events, there is a *map*, displaying the operation’s region, that shows icons for each of the *events* and *tasks* that have coordinates assigned. These can be created by clicking anywhere on the map as well. Below, incident managers can see the region’s weather forecast as well, fetched through the *Icelandic meteorological office’s XML service*. For the map, it is possible to choose between *Google Street Map*, and *Ískort*, with the latter showing much more detailed terrain and geographical information.

The hierarchy is closely related to the SÁBF framework. It is displayed here because it is important to get information quickly on who to contact with different issues.

Finally, *communication methods* list how to contact, for example, different units or incident managers at the headquarters or those in the vehicles.
2.5.2 Skjáir (e. Screens)

In the view Skjáir (e. Screens), one can view most of the components found under Framgangur (e. Progress), with the addition of Tasks in progress.

![Figure 2.4: Screens page showing the map, with Ískort selected.](image)

Screens use the whole screen for a single component, for better visibility and increased clarity. Skjáir are used for showing information on large screens in the ICE-SAR command center, and the mobile command center, Björninn.

2.5.3 Lotur (e. Rounds)

Through Lotur (e. Rounds), the SAREye IMS facilitates planning, the way it is done at ICE-SAR and the Civil Protection. Rounds have specific start and end times, they cannot overlap, and at each point in time, only one round is active. Some operations may only have one round, called first response, while others, such as the one currently revolving around the eruption at Bárðarbunga, has over 150 rounds.

Each round can have goals, weather forecasts, risk assessments, and results from a status meeting associated with it, but it is not a requirement.

The rounds also serve as containers for groups, incidents and tasks, as each can be associated with a specific round. If these entities are created at Framgangur, they automatically
belong to the active round. However, under *Rounds*, they can be created for specific rounds, and doing so for groups and tasks is the primary use of the *SAReye IMS* for planning. While tasks are being carried out for the current round, planning is being done for the next round in line.

Each round can have a specific hierarchy, and the hierarchy for the currently active round is the one displayed at *Framgangur*.

### 2.5.4 Summary and non SÁBF related features

It should now be clear how each of SÁBF’s units can use the IMS to their advantage when carrying out their work. We will not go into detail about the other components seen in the images above, except *Areas*, which will get a dedicated section later on. However, for completeness, here is a short summary of each of these components:

- **Setup** is the form for the operation, where the basic information, severity, region and so fourth are entered.
- **Checklists and files** groups together all files that have been uploaded for individuals tasks and events, and checklists that incident managers have created.

---

Figure 2.5: *Rounds page of the search operation for Nathan Foley Mendelssohn.*
• **Change log** lists every single creation, edit, deletion and attribute change (coordinates, descriptions, priorities etc), that have been made since the start of the operation, along with the responsible incident manager’s name.

Here are a few other features and designs of the *SAReye IMS*, that are also, strictly speaking, not a part of *SÁBF*:

• Reports for chosen periods of time, for individual volunteers, vehicles and rescue units. Lists of operations participated in are available for each unit, volunteer and vehicle, and they are exportable to spreadsheets.

• Visualization on maps, where we show all coordinates for all operations in a given time period, with coordinates shown either individually along with relevant information, or in the form of heatmaps.

• Locking operations with SHA-2 encoded passwords. We keep a list of those who have entered the password correctly, and that list is emptied on password change.

• In-system messaging, mainly used to report erroneous logging.

• View history, showing who opened which operation and when.

• Tables on *Framgangur* and *Skjáir* are updated automatically every 15 seconds, with the newest tasks, groups and events.

• All post requests for the forms on *Framgangur* are done with AJAX, so the page is never unnecessarily reloaded.

We have now covered the beginning of *SAReye* as a company, from its proposed solution to an yet unsolved operational issue, to its pivot into creating an incident management system, as well as the background and features of the resulting *SAReye IMS*. In the next two chapters we will cover the backgrounds, implementations, and evaluations of both the new *Preparedness* system, and the new *Search area and evacuation management* component in the *SAReye IMS*.

To clarify, before we move on to the next chapter, a diagram showing the relationship between the frameworks and systems discussed in this chapter is presented below.
Figure 2.6: The SÅBF operational management framework is the foundation of the SAR-eye IMS, and is itself based on NIMS and ICS. The two thesis contributions are the search area and evacuation management component, which is a part of the SAR-eye IMS, and Preparedness, an entirely new system from which the SAR-eye IMS requests and receives data.
Chapter 3

Preparedness

This chapter describes the background, implementation and evaluation of the new Preparedness system. Note that even though Preparedness is in Icelandic, all images from the system will be in English, with the exception of user generated content.

3.1 Background

There are situations that can arise, where it is not feasible to decide what to do, once a crisis is already underway. These situations, or disasters, require planning ahead. They require preparedness.

3.1.1 Structure and content of preparedness plans

The Civil Protection in Iceland is responsible for writing preparedness plans for a multitude of possible disasters. These include plans for disease outbreaks, forest fires, plane crashes, eruptions, earthquakes and maritime accidents. There is a full list of plans on their website. All of their preparedness plans follow the same chapter structure:

1. Inngangur (e. Introduction)
2. Staðhættir (e. Topography)
3. Skilgreiningar (e. Definitions)
4. Boðun (e. Callout)

List of Almannavarnir’s preparedness plans: http://www.almannavarnir.is/default.asp?cat_id=89
Chapter 7 usually describes some specific scenario, for example, a plane crash at sea in a plan for a specific airport, or something that is specific for the type of disaster, for example an evacuation plan, which is only found in a handful of plans.

Each plan defines three potential phases, Övissustig (e. Uncertainty), Hættustig (e. Alert), Neyðarstig (e. Distress). As an example, here are the translated definitions of these three stages for the plan Landsáætlun um heimsfaraldur influénsu (e. Nationwide plan for an influenza pandemic)\(^2\):

- **Övissustig**: A new influenza virus subtype has been detected in humans, but no known human-to-human transmission, except in exceptional cases of very close contact.

- **Hættustig**:
  - Small, localized clusters in humans caused by a new influenza virus subtype which does not appear well adapted to humans.
  - Larger clusters in humans, but still localized. Evidence of the virus having become increasingly better adapted to humans, but not to the extent of posing substantial risk of a pandemic.

- **Neyðarstig**: Pandemic: Increasing and sustained spread of human-to-human transmission in general population.

The plans also list all emergency responders that should respond to the crisis. The preparedness plan for the influenza outbreak lists 48 responders and divisions within them, and these range from financial institutions to the Icelandic Coast Guard.

\(^2\) Preparedness plan can be found here: http://www.almannavarnir.is/upload/files/31032008_influenza_áætlun_utgafa_1.pdf
Tasks to carry out are defined for each of the responders, for each of the three phases. In the case of the influenza plan, there are 144 sets of tasks defined for these three stages, each containing from 1 to 13 tasks. In total, there are well over 400 tasks defined.

Figure 3.1: Task list in Icelandic for the health care sector, specifically the Icelandic surgeon general.

### 3.1.2 How preparedness plans are written

The preparedness plans are edited and managed by the Civil Protection. At the time of writing, there are three project managers at the Civil Protection. They are educated in police work, and have received basic training in project management. Each project manager spends around 300 hours on each preparedness plan. The plans are on average 74 pages long, and their lengths range from 45 to 125 pages. They require the joint effort of many institutions, municipalities, organizations and other entities. Therefore, they can have between 20 and 30 authors.

### 3.1.3 Problems with the writing

A number of problems have plagued the writing of preparedness plans. No clear methodology is used, and the procedure is unclear as work processes are not well defined (Pálsson, 2013).

According to Guðbrandur Örn, and what was confirmed by Viðir Reynisson and Eggert Magnússon from the Civil Protection in Iceland, at the beginning of writing a preparedness plan, in general, there is a skeleton created. Then the plan is sent to some of the
authors, and they write their part. Then it is sent to the next set of authors, and they write their part. It is therefore a linear process. Writing the plans in this manner is cumbersome, very time consuming and does not allow for much overview of the writing.

Furthermore, there can be hundreds of tasks defined for the emergency responders, and since their definitions only exist within PDF and Word documents, they are unstructured. This means that external system cannot easily, or at all, fetch these tasks for further use. Therefore, the list of tasks in the preparedness plans are completely segregated from the systems that are used to carry them out, one of which is the SAReye IMS. This means that all the hundreds of tasks may need to be put by hand into the SAReye IMS, a very time consuming and error prone process.

This also means that when disaster strikes, there is no solution available to responders for quickly fetching their individual tasks, for the specific phase. This means that they have to either have the corresponding preparedness plan handy, which is highly unlikely, or download it as a whole from the Civil Protection’s website, find the appropriate chapter and then their list of tasks.

Finally, there is no version history of individual chapters and sections of the preparedness plans, and difficult so see who wrote what, when the preparedness plans are complete. There is also no easy way for the editors to approve of a new version, and have it available at a central location instantly.

### 3.1.4 Potential solutions to the problems

When this M.Sc. project started, and at the time of writing, no software solutions dedicated to the writing of preparedness plans existed. And such solutions would have had to be tailored to the needs of the Civil Protection, or at least be flexible enough to allow for their structuring and formatting of plans.

There are software solutions that would enable a more distributed approach to the writing of preparedness plans. Examples include wiki systems, Google Drive and Microsoft Word Online. These solve the problems to a varying degree, but none completely.

- **Google Drive** is excellent for real time collaboration. The version history is a bit flawed in that the versions are created automatically. The authors have no control over what constitutes a new version of the document.

- **Wiki systems** are abundant and some offer good collaboration mechanisms.
• *Microsoft Word Online* offers great real time collaboration of authors, and minor and major versioning of documents.

For the purpose of simply writing preparedness plans, *Microsoft Word Online* would probably have been one of the top choices, due to good versioning and collaboration functionalities.

However, none of these offer solutions to all of the problems. There would still be no way of efficiently fetching tasks for the purpose of executing them with the aid of an *IMS*, or to display them quickly to emergency responders. Editor’s would also not be able to accept new plans or versions thereof, for immediate availability online.

### 3.1.5 Thesis contribution

This thesis introduces a novel solution that integrates with the *SAReye IMS*, and solves the problems mentioned above. The solution offers the following novel features, which will be explained in more detail in the remainder of this chapter:

- When creating operations within the *SAReye IMS*, incident managers can choose to use a preparedness plan from *Preparedness* as a template. The name of the chosen preparedness plan becomes the operation’s name, and all tasks and communication methods defined for the preparedness plan are fetched from *Preparedness*, created in the *SAReye IMS* and assigned to the new operation.

- We enable distributed work processes for preparedness plan creation, with authors and editors being assigned to each plan, who can then efficiently work together in writing them.

- Our versioning functionality behaves like a version control system. Authors submit changes to a specific chapter, and these changes can be reviewed at any point in the future. When ready, the editors of a preparedness plan can simply accept its current version. When they do, the current version will automatically become the active version of the plan, and as such, will be instantly available online in PDF and HTML format. It will furthermore become the version fetched by the *SAReye IMS*, if the preparedness plan is put into action there.

- A simple but effective interface was created for responders to search for a specific preparedness plan, and a phase within it, and see the tasks they are expected to carry out.
It is important to note that the project started with Guðbrandur Örn Arnarson’s initial wireframes and sketches for the *Preparedness* system, although the design has changed considerably since. No code for *Preparedness* had been written before the start of this project.

Before we delve into the implementation of the features listed above, we will first go over the technology stack used for development, and general design principles.

### 3.2 The technology stack

During development, *Heroku*[^3] was used for hosting. To set up an instance of a free *Heroku* server, which is fast enough for development purposes, takes well under an hour, and deploying changes after that is quick and easy, and can be done from the developers own terminal with a simple command.

Since the back-end for the **SAReye IMS** is written in *Rails*, a decision was made to use *Rails* as well for *Preparedness*. We have found that development is extremely rapid in *Rails*; the *Ruby* programming language and the *Rails* framework are very intuitive in nature, and there is a plethora of extensions available written by the *Rails/Ruby* community.

*Preparedness* uses a *PostgreSQL* database[^4]. We used *PostgreSQL* for the **SAReye IMS** because it is free and easy to use, it is used by many start-ups today (*Instagram*, *Plain Vanilla* etc), and it has great community support. For those reasons, as well as for keeping consistency and the fact that is is the default choice when hosting on *Heroku*, using *PostgreSQL* for *Preparedness* seemed like the obvious choice.

The front-end is written in *HTML5*, *CSS3* and *JavaScript*. *Bootstrap*[^5]; an *HTML*, *CSS* and *JavaScript* framework, and *jQuery*[^6], a *JavaScript* library, were extensively used to speed up the development. *Bootstrap* was exceptionally helpful in that regard.

[^3]: Heroku is a cloud application platform that supports multiple web development frameworks. Heroku’s website: https://www.heroku.com/home
[^4]: PostgreSQL is a powerful, open source object-relational database system. Website: http://www.postgresql.org database
[^5]: Bootstrap makes front-end development faster and easier. Website: http://getbootstrap.com
[^6]: jQuery is a fast, small, and feature-rich JavaScript library. Website: https://jquery.com
3.3 General design principles

Every word that is displayed within the system, except user generated ones, is translatable. The *I18n Ruby gem* 7, which ships with *Rails*, provides a framework for translating applications, and is used for this purpose, for content that is rendered using *Ruby* as well as *JavaScript*. A `.yml` file is used to define the translation for each word, using a specific file for each language. We have for example the files `is.yml` and `en.yml`. Although the possibility to change languages has not been enabled, implementing multi-language support from the start was done so that the migration to a more general purpose software would be easier, although the software is still slightly tailored to *Civil Protection in Iceland*. Implementing multi-language support in a system as an after thought would have been very difficult, as we have learned with the *SAREye IMS*.

![Figure 3.2: A small part of the is.yml file for Preparedness.](image)

Other design principles were keeping repetitions in the code to a minimum, and separation of concern. The *Model-View-Controller* design pattern was followed to the letter. No logic can be found in any of the views, and any functionality that is used more than once is implemented in the models, so that different functions in the controllers can share the functions implemented there. Partial views are used extensively, for all content that is rendered in more than one place. In fact, one of the principles during development was that if three or more lines of code were used more than once, they were put into a function. This has resulted in a rather compact code base for the entire system, and using descriptive function names and using the *MCV pattern* has kept the code readable.

As a rule, *CSS* and *JavaScript* code was completely kept out of the views. These are all kept in separate *CSS* and *JavaScript* files, so that *JavaScript* functions and *CSS* definitions can be reused.

Finally, most of the operations available to the user, whether loading or saving content, is done using *JavaScript*. This means that there are fewer page refreshes within the system. Requests are rather sent using *AJAX* 8, and the user is notified that the operation was a success through simple success messages appearing in the user interface. This makes for a very nice and smooth user experience.

7 RubyGems is a package manager for the *Ruby* programming language
3.4 Writing a preparedness plan

In this section we will now go through the journey of writing a single preparedness plan using Preparedness. We will use one of the older plans in our example, specifically Eldgos undir Eyjafjallajökli, and go through how that plan might have been written in our system.

We will cover everything from the registration process to publishing multiple versions of our preparedness plan. In between we will be covering the following:

- Creating preparedness plans.
- User permissions and roles.
- Adding authors and editors to preparedness plans.
- Writing a chapter.
- Viewing chapter versions and changes made by other authors.
- Defining tasks and communication methods.

Afterwards, we will take a look at a few examples from the PDF that is generated for our preparedness plan.

The first set of functionalities we cover are the basics of user authentication and permissions, to set up a basis for functionality discussed later on.

3.4.1 Sign in and registration

Devise\(^9\), an open source authentication solution for Rails, was chosen to handle authentication, registration and sessions. The SAREye team had successfully used Devise before for the SAREye IMS. Plugging in and configuring Devise for a Rails project is easy. It is modular and flexible and enables easily restricting certain actions within the system to, for example, only admins. Devise takes care of the work regarding user registration, sign in, session management, password resets, etc, within Preparedness. The only work that was needed was customizing the sign in, registration and password retrieval forms, and designing the underlying user model used by Devise.

Using emails addresses is necessary for password retrieval and possibly, in the future, sending users summaries of recent activities. Therefore, email addresses are used for

\(^9\) Devise on GitHub: https://github.com/plataformatec/devise
registration and sign in. They are also unique, and as such can be used for identification. Personal information entered during registration is limited to a phone number and full name.

![Preparedness registration form](image)

Figure 3.3: *The registration form.*

Preparedness plans are generally not confidential, and anyone can therefore register and view preparedness plans, but all other actions are restricted until the user is given further permissions.

### 3.4.2 Permissions

At first, registered users can only view preparedness plans and download them as PDFs. Admin rights can be assigned to other users by admins only (at first, only one admin exists), and admins are the only users that can create preparedness plans. They can also create emergency responders and call levels, and types and locations that can be assigned to preparedness plans, all of which we will cover later.

Apart from the admin role, which is system wide, we have two other possible roles for users. These are *editor, author* and *creator*. Editing, authoring, and creators rights are given to users on a per preparedness plan basis. The role of an editor, within a specific preparedness plan, is fourfold:

- Assigning authors to the plan
- Accepting a current version of the plan once one is completed
- (Possibly) publishing the plan and
- Writing the plan
An *author* of a preparedness plan only has the ability to write the preparedness plan. Therefore, editors are authors as well. The plan’s *creator* has all the rights of an editor, with the added rights for adding editors, and changing the name of the preparedness plan.

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Authors</th>
<th>Editors</th>
<th>Creator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write in all chapters</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Add authors</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Add editors</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Change plan's name</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Figure 3.4: A table showing the permissions for each of the three roles.

The permissions are structured so that those working at the *Civil Protection in Iceland* will mostly be creating and editing plans, and author rights will be given to those whom the plan concerns. These can be, for example, police or other authorities in the area that the disaster described in the plan may affect. Those that have assigned authors roles, are usually only responsible for writing a few paragraphs or maybe a single chapter.

### 3.4.3 Creating a preparedness plan

First, let’s take a look at the front page of *Preparedness* as it is before we create our preparedness plan.

Figure 3.5: The front page of Preparedness, before we create our plan.

The front page contains a list of preparedness plans, filterable on type and location. The location we need for our preparedness plan, *Eyjafjallajökull*, has not been created yet, so we begin by adding it. To do so, we go to the admin panel, which is accessible through the dropdown in the header, or by clicking the link on the left side on the front page. The latter brings us straight to the desired function.
After adding our location, we go back to the front page, and now we are ready to create our preparedness plan. To create the plan, we click the *New preparedness plan* button.

The specific chapter, as mentioned in 3.1.1, is a chapter that is reserved to handle something that is only needed in some subset of plans, depending on their subject. We choose our type and location, and since we are writing about a possible eruption, we choose *Rýmingaráætlun (e. Evacuation plan)* as a specific chapter, and save.

### 3.4.4 Adding editors and authors

After creating our preparedness plan, we get to the empty canvas for our preparedness plan.
Figure 3.8: *Empty editor for the first chapter, Introduction, in our new preparedness plan.*

We can pick which chapter we want to contribute to, and we can see the, now empty, change log for the chosen chapter. The creator of a plan will automatically be added as an editor for it, and as creator, the first task is adding a few additional editors to help out, who will in turn add a few authors as well. To add editors and authors, we go to the editors panel.

Figure 3.9: *Editors panel, where editors can, among other things, add authors.*
In figure 3.9 we can see that a couple of editors have been added. By adding editors, we have added a few authors as well. A couple of other authors have been added as well, so we can start writing the preparedness plan.

### 3.4.5 Writing a chapter

The text editor in figure 3.8 is called *Summernote*[^10]. *Summernote* is a WYSIWYG editor that encodes formatted text using HTML. Because of the fact that *Summernote* encodes the formatted text in *HTML*, we can simply save the *HTML* buried within the editor when the user clicks *Save*, and display it again by inserting the *HTML* into the editable region of the *Summernote* editor.

![Figure 3.10: A very small example of HTML within Summernote.](image)

The *Summernote* editor in *Preparedness* has been translated to Icelandic using a `.yml` file similar to the one we saw in figure 3.2. The *Summernote* editor has most of the formatting options found in most word processing software, including adding images and tables as well as undoing and redoing actions.

A few things to note:

- The save button in figure 3.8 is disabled until we edit the chapter, and it is disabled again if we change the chapter and then backtrack the changes. This is done to prohibit creating two identical versions of the chapter.
- Second, when we save our changes, a new entry is added at the top of the change log without a page refresh.
- If we edit the chapter, and then click on one of the other chapters without saving, a modal opens asking us whether we want to save or discard the changes just made.

[^10]: *Summernote* is a WYSIWYG editor on Bootstrap. Website: [http://summernote.org//](http://summernote.org//)
3.4.6 Viewing chapter versions and changes made by other authors.

Let's assume that we and the other authors have completed writing the chapter *Staðhættir*.

Figure 3.12: *Six revisions where made to Staðhættir. For each revision in the changelog we can view that version or the set of changes made.*

On the right, in figure 3.12, we can see the chapter’s change log. We can see each change that has been made, and who made it and when. The current version is loaded when a chapter is selected.
There are two possible actions for each change in the changelog. The first one is viewing the chapter as it was after the chosen change. We can do this by clicking View version. This loads the chosen version into the Summernote editor. We do this, rather than displaying the version in a modal, so that we can base a new version of the chapter on some arbitrary old version. This effectively enables backtracking changes. We can select an old version, make changes, and save again, resulting in a new current version. In figure 3.13 we have selected version four of the chapter, as can be seen by the green highlighting on that version.

Figure 3.13: Version four of Staðhættir selected.

The second option is to see the changes that the author made.

Figure 3.14: Viewing the changes made in iteration six of the Staðhættir chapter
By clicking View changes, we open up a modal that displays how the chapter changed between the chosen version and the one before it. This can be seen in figure 3.14.

Text that was added in the version is highlighted in green, and text removed is highlighted with red and a strike-through. The following is a brief explanation of how this is implemented.

Each time a change to a chapter is saved, the entire content of the Summernote editor, which is represented in HTML, is saved to the database and associated with a version number. In figure 3.15 we can see a bit larger example of this HTML content than before.

![Figure 3.15: HTML within the Summernote editor](image)

Saving the entire HTML inside Summernote is done mainly so that we can load the version again into the Summernote editor, to enable the See version feature discussed above. If we would only save the text to the database, by first removing the markup, we would loose any formatting done by the author.

To show the changes made in a specific version, we fetch the version that the user asked for, along with the version before it (if one exists, the changes in the first version are of course the first version itself). Then a tool called HTMLDiff is used to compared the two versions. HTMLDiff is a Ruby gem that can be used to compare HTML, and the best tool found. It focuses only on content when comparing HTML. The HTMLDiff comparison method returns HTML that highlights content added and deleted, in green and red.

11 HTMLDiff on GitHub: https://github.com/myobie/htmldiff
Writing a *HTML diff tool* from scratch was not an option, simply due to the scope of the project. Writing a good one would have taken weeks. It must be mentioned though that even though *HTMLDiff* was by far the best tool found in an extensive search online, there are a couple of drawbacks.

The first one is that the tool only detects content changes, so changes in for example font sizes are not detected. Optimally, we would like to be able to show formatting work that authors went through, especially if they were cleaning up content written by others. However, the content itself is the most important part of the work, and the changelog is there to answer the question *Who wrote that particular part?*, more so than who decided the spacing or font sizes. Furthermore, as stated, writing a diff tool from scratch would not have been feasible, and no good tool was found that detected content changes, probably because finding a good way to display them is not small task either.

A slightly bigger issue is that *HTMLDiff* did not detect any changes that had to do with images. For the most part, that is fine, and it is hard to convey to the user that a picture was aligned differently or re-sized. We would however like to know if the author deleted or added images. Therefore, before comparing the *HTML* of the two versions, the text *Picture <number>* is inserted before each image, where *<Picture 2>* would indicate that the image is the second image in the chapter. If the second picture was deleted in the version the user wanted to see the changes to, then the text "Picture 2" will be missing in that version when it is sent to *HTMLDiff*. A *Picture 2* highlighted in red will therefore appear in the diff along with the deleted image, indicating that the author deleted it. Images added will be accompanied by a green *Picture X* above them in the diff. This can be seen in figure 3.16.

![Figure 3.16: Showing addition of an image.](image-url)
3.4.7 Defining tasks and communications methods

Now that we have covered how the \textit{diff} functionality is implemented, we can move on to defining tasks for emergency responders. This, as well as defining communication methods, is the main connection between \textit{Preparedness} and the \textit{SAReye IMS}.

![Figure 3.17: The code that handles fetching the diff of two chapter versions. Notice the part that adds an identifier above images, so that additions and deletions of images can be seen.]

![Figure 3.18: Authors can choose between creating tasks for emergency responders, or work on the free text. The free text is the same thing as in all other chapters, and will be rendered in the PDF above the tasks table.]

Defining tasks for emergency responders

In figure 3.18 we can see the user interface for defining tasks for emergency responders. Let’s assume that the emergency responders we need have not been created yet, and that we need to add a new phase to use for our preparedness plan. We can add new ones by clicking the dropdown at the top of the page, and going to the admin panel, displayed here below.

![Screenshot of user interface for defining tasks for emergency responders]

**Figure 3.19: Now we have added Landhelgisgæslan as an emergency responder.**

We add the missing emergency responders and phase and go back to our preparedness plan. Now we can start defining tasks for our emergency responders.

Each task will have a description and is assigned to a particular emergency responder. Each task will also belong to a certain phase. Finally, we can choose whether the task should go to the SAReye IMS when the preparedness plan is initiated there. Some tasks are not large or specific enough to warrant their own task in the IMS.

When an incident manager initiates the plan in the IMS, he or she picks a phase for the incident. All tasks that are defined for the chosen phase and marked for migration to the IMS, are fetched and added to the operation that is automatically created. We will cover this migration in a bit more detail later on, but for now, it suffices to say that this saves a lot of time and increases accuracy during incident management, as there can be dozens of these tasks.

When HTML and PDF versions of the preparedness plan are loaded, a table will be constructed and rendered that lists all the tasks. This table is positioned below the free text for the Verkefni viðbragðsáðila chapter.
Figure 3.20: *Now we have defined a few tasks for Vegagerðin for the phase Hættustig.*

**Defining communication methods**

Communication methods can also be defined for the preparedness plan. The communication methods will be fetched when the plan is initiated in the IMS, and just as with the tasks, this saves time, and increases accuracy.

Figure 3.21: *Adding a single communication method. This is exactly the same form as in the SAReye IMS.*

### 3.4.8 Publishing a preparedness plan

Now let's assume that we and all the other authors and editors have finished writing our preparedness plan, and that is is now time to publish it.
To do so, we go to the editors panel. There we can see a list of those editors that have approved the plan as it is now. Note that this list of approving editors is emptied if new changes come in, in any chapter.

![Figure 3.22: All editors have approved of the current version, and the preparedness plan is about to be published](image)

We approve the plan and then click the *Publish* button. All the chapters, and all the tasks and communication methods defined are used to construct a new preparedness plan. The new version appears in the version list on the editors panel, seen here in figure 3.23.

We can choose to view the plan as *HTML* or *PDF*, or view a list of the authors and editors that contributed to it.

![Figure 3.23: The plan has been published.](image)
Figure 3.24: The authors that contributed to this first version of the new preparedness plan.

On the front page, it is now possible to view the newly released, current version of our preparedness plan in HTML or PDF format.

Figure 3.25: HTML and PDF versions of the preparedness plan can now we viewed.

It is important to note that the PDF now available is not actually saved to the database. When a PDF is requested for any version of a preparedness plan, the version’s compiled HTML is used, among other things, to render the PDF. This means that we can change the way we render the PDF at any point in the future, changing for example the front page, the font, margins, colors or anything else that might come to mind later on.

A Ruby gem for Ruby-On-Rails called Wicked PDF 12, which relies on another gem called wkhtmltopdf-binary, is used for the PDF rendering. All that is needed to render the PDF

12 Wicked PDF on GitHub: https://github.com/mileszs/wicked_pdf
after installing Wicked PDF is a HTML version of the preparedness plan, a template view for rendering (with a .pdf.erb ending), and a separate CSS file for any specific styling we might want in the PDF. The pieces of code that handle most of the work for the PDF rendering can be seen in figures 3.26, 3.27, 3.28 and 3.29.

Figure 3.26: The template view for the PDF rendering.

Figure 3.27: The part of the code that handles setting up the PDF rendering.
Figure 3.28: The part of the code that puts together the HTML for the PDF rendering; adding a list of authors and editors to the front page for example.

```html
.container_pdf {
  margin: 0;
  background-color: #FFF;
  width: 300mm;
  height: 300mm;
  text-rendering: optimize-speed;
}

.page {
  display: block;
  clear: both;
  border: 2px solid #FF0000;
  page-break-after: auto;
}

#communications_table {
  margin-left: auto;
  margin-right: auto;
  width: 100%;
  text-align: center;
  background-color: #000;
}
```

Figure 3.29: A small part of the CSS file for PDF specific styling.
3.4.9 Examples from the PDF

The following are a few example pages from the PDF for our preparedness plan. We can see that a front page containing the title of the plan and a list of all the authors and editors, as well as a table of contents, have been prepended to the plan. The tasks and communication methods defined have all been compiled into tables, with the tasks sorted on phases first and emergency responders second. The version history of the preparedness plan appears at the end.

Note that this is only an example, and the preparedness plan has not been entered in its entirety, neither the texts nor the tasks or communication methods.

Figure 3.30: The front page of the generated PDF.

Figure 3.31: The table of contents in the generated PDF.
Figure 3.32: An example of chapters.

Figure 3.33: The tables containing the tasks and communication methods defined.

Figure 3.34: Images in the PDF and the version history of the preparedness plan.
3.4.10 The preparedness plan’s version history in the changelog

When multiple versions of a preparedness plan have been published, one might want to see in the changelog for a chapter, a representation of when each version of the plan was published. It is not only relevant to see all the changes made to a chapter, we also want to be able to see which changes have been made since the preparedness plan was published last, and which changes where made between version $X$ and $Y$. This functionality is shown in figure 3.35

![Figure 3.35: We can see indicators for when versions of the plan were published, intermittently between changes in the chapter’s changelog.](image)

3.4.11 Viewing recent changes to preparedness plans

One feature that has not been mentioned yet, is a tool for viewing all changes made to preparedness plans that one is an editor of. By clicking the dropdown at the top-right corner and then My plans, or the My plans link on the front page, we go to a page that shows all the changes that have been made, to all of the plans that one is an editor of. This page can be see in figure 3.36. By clicking a row in the table, it is possible to see exactly what was changed, both additions and deletions, which works exactly like it does in the changelog discussed earlier. For each change, we can see in which preparedness plan it occurred, in which chapter, and who made the change and when.
3.5 Preparedness: A tool for emergency responders

One of the biggest reasons for defining the tasks for preparedness plan in a more structured manner, was so that we could represent them in a more accessible manner to emergency responders, mainly to speed up the process of them finding out which responsibilities they have during a particular disaster. This functionality is shown in figure 3.37.

Figure 3.36: Changes made to the preparedness plans I am an editor of, for the last month.

Figure 3.37: The emergency responder begins by selecting the relevant preparedness plan. Then the dropdown for phases populates with phases that have tasks defined for them in this preparedness plan. After selecting the phase, the dropdown for emergency responders populates with those that have tasks defined for the chosen phase. Finally, when the emergency responder is selected, a table appears at the bottom, with the defined tasks.
3.6 Preparedness: The integration with the SAReye IMS

When creating a new operation in the Civil Protection in Iceland’s instance of the SAReye IMS, it is now possible to select a preparedness plan as a template, seen here in figure 3.38.

![Figure 3.38: Using a preparedness plan as a template for an operation.](image)

After selecting our preparedness plan, we must select the phase, so that the correct tasks are fetched over from Preparedness. Finally we click Confirm, and the results can be seen in figure 3.39.

![Figure 3.39: Note that the tasks created have the name of the emergency responder responsible prepended in brackets.](image)
An operation has now been created for the chosen preparedness plan and phase. All the tasks and communication methods defined for the preparedness plan Eldgos undir Eyjafjallajökli have been fetched over from Preparedness. The name of the operation created is the same as the name of the preparedness plan.

There are two sides to this functionality; the IMS side and the Preparedness side, and a few steps. The figures referenced in the following list appear below it.

1. When the user opens the modal to create an operation from a preparedness plan, an AJAX request is sent to the SAReye IMS server. The code is in figure 3.40.

2. Then the SAReye IMS sends a request to Preparedness’ API, requesting all the preparedness plans that have been published. The code is in figure 3.41.

3. Preparedness gathers all the published preparedness plans, and goes through all of the tasks for each plan, creating a list of phases that have been used for tasks that should go to the SAReye IMS. Preparedness then responds with the lists of preparedness plans and phases. The code is in figure 3.42.

4. After receiving all the preparedness plans and phases, the SAReye IMS sends the list to the user that requested it originally. Again, the code is in figure 3.41.

5. The users client uses the list of preparedness plans and phases to construct a form. Again, the code is in figure 3.40.

6. When the user has chosen a preparedness plan and phase, a request is sent to the SAReye IMS to created a new operation from the preparedness plan. The code is in figure 3.43.

7. When the SAReye IMS receives the user’s request, it creates a new operation, and the sends a request to Preparedness for all of the tasks and communication methods defined for the preparedness plan, for the given phase. The code is in figure 3.44.

8. Preparedness responds with all the tasks defined for the preparedness plan and phase, that are marked to go to the SAReye IMS. The code is in figure 3.45.

9. The SAReye IMS receives the lists of tasks and communication methods, and creates new tasks and communication methods for each one of those received from Preparedness. The code is in figure 3.46, and it is the rest of the function from figure 3.44.

In section 5.1.4 we will discuss possible extensions to this functionality. Below are all of the figures referenced in the list above.
Figure 3.40: The request from the client to the IMS for preparedness plans, and the code that builds the resulting form from figure 3.38.

Figure 3.41: IMS requesting plans and phases from Preparedness.

Figure 3.42: Preparedness responding with all preparedness plans and phases.

Figure 3.43: Requesting creation of a new operation using the selected preparedness plan as a template.
48 Implementation of end-to-end incident management processes in a unified IMS

Figure 3.44: The creation of a new operation, in the IMS, and the requesting of tasks and communication methods for the given plan and phase.

Figure 3.45: Preparedness responds with all of the tasks and communications defined for the preparedness plan (tasks only for the given phase).

Figure 3.46: The rest of the function from figure 3.44, creating tasks and communication methods in the IMS.
3.7 Evaluation

This section describes the evaluation of Preparedness. The evaluation is based on testing the systems usability. Discussion on the results, general comments from the users and possible steps to improve the system, are given in this chapter as well.

3.7.1 Users

The Civil Protection in Iceland supplied two users for the usability testing. The first user was Eggert Magnússon, the Detective Chief Inspector and a project manager at the Civil Protection. The second user was Víðir Reynisson, department manager at the Civil Protection. Neither of the users had used the system before.

3.7.2 Environment and duration

The usability testing took place at the Civil Protection’s headquarters, and both users used their own work computer, and browser of choice. The process took just over an hour for each user.

3.7.3 Methods

The Think-aloud method was used for the usability testing of Preparedness. Two users were asked to perform 19 steps in the creation of a preparedness plan within Preparedness. The users were asked to think aloud while performing the steps, and were not interrupted while doing so. If something went amiss, a discussion as to why that happened was initiated. In some instances the users initiated a discussion on the problems as well.

3.7.4 Data gathered

During the usability testing, written notes were gathered on the following:

- Whether the user was successful or not, and if not, what steps in the wrong direction where taken.
- Discussions with the users, both during and after the usability testing.
- What users said while they were performing the steps.
3.7.5 Measurements

In the usability testing, the focus was on whether users could or could not finish the steps. However, for tasks completed, notes were also taken on exactly how easy it was for users to complete the steps, and whether there was something that could have gone better.

The emphasis was not on how quickly tasks can be performed. They functionalities of the system are not that many, and users will hopefully move more quickly about the system after using it for a while. Not finding features however, means that users will not be able to learn using them. Therefore the focus was more on whether users could find different components and actions at all to perform the steps.

Evaluating how easy it is for users to cooperate on a preparedness plan, over a long period of time, is out of the scope of this usability testing. That needs to be done in later stages, when users have started using the system on a daily basis. Data on cooperation will be gathered over time, with more experience in using the system, and most likely in a less structured manner.

However, in the discussions, both during and after the usability testing, the users where free and encouraged to comment on the system as a whole, and the process of managing and creating preparedness plans within it.

3.7.6 The steps performed by users

The users were asked to perform the steps in the list here below. A summary of what went wrong or could have gone better, and possible solutions to the problems, are presented in section 3.7.9. In Appendix A, the full results from the usability testing are listed in a table.

Both users were given admin rights upon registration, so they could perform the step of creating a preparedness plan. The step of initiating at preparedness plan in the SAReye IMS was omitted, as it has not been released yet on their server, and as a side note, the UI is very simple.

Here are the steps the users where asked to perform:

1. Go through the registration process.
2. Create a new preparedness plan. User is asked to use a type and location that does not exist yet, and is expected to create both, leading to the next step.
3. Find the correct view to create the aforementioned type and location, and create both.
4. Finish the process of creating the new preparedness plan.
5. Add me as an editor to the new preparedness plan.
6. Add some text of own choosing to the Introduction chapter and save it.
7. View changes made by another author to the chapter Stadhættir.
8. Write a new version of the Stadhættir chapter using an earlier version as a base.
9. Create two tasks for an emergency responder and phase of own choosing.
10. Add free text to the Verkefni viðbragðsadila chapter.
11. Add a new communication method.
12. Publish the preparedness plan.
13. View the PDF version of the newly published preparedness plan.
14. Go back to the front page.
15. Go back to working on preparedness plan. User asked to use links, not the back button in the browser.
16. Go back to the front page. User asked to use links, not back button in the browser.
17. View a PDF version of the plan (should access it from the front page).
18. View changes made to all plans you are currently and editor of.
19. Find a task for a specific preparedness plab, responder and phase in the tasks view.

After each usability test, a short discussion took place about the overall experience of using the system.

3.7.7 Discussion with Eggert Magnússon

Eggert said the system looked very good, and that it will be a great tool for cooperating with other authors. He was also very excited to see plans initiated in the SAREye IMS, and seeing the defined tasks appear there automatically.

13 There is no link to this view as of yet, so the URL was entered for them. It has not been decided whether or not users should be redirected automatically to this page if on mobile. Or whether the link should be like the other ones on the frontpage, or an icon. Adding one, when decided, will only take minutes.
Eggert mentioned that a more fine grained access control would maybe be beneficial. Some authors would actually have no need to edit certain sections, and it would only confuse them to have so many chapters. Having access control down to a chapter or even sub-chapter level might be beneficial. After a bit of discussion though, Eggert did agree that this might complicate things, and that there might be a lot of work involved in assigning permissions. He also agreed, when it was mentioned, that the functionality that editors have to accept changes would indeed see to it that no changes that should not be there would slip through.

Eggert also noted that Umfang should be added as a categorization for tasks, and that they should be able to create Specific chapters within the system themselves.

Finally Eggert said that he was very excited to start using the system. He mentioned that it would make their job a lot easier, and more accurate, to be able to delegate responsibility to other municipalities in such an easy way.

### 3.7.8 Discussion with Víðir Reynisson

Víðir said that the most important features were getting the tasks to the SAReye IMS, and being able to view them easily on mobile. He said that the tool would greatly simplify the preparedness plans creation and management for them. He said that up until now, there have been big issues with copy pasting between preparedness plans. Published plans regarding for example the town Ísafjörður, would have Vestmannaeyjar mentioned in multiple locations. He said that that time is now over.

We discussed adding Umfang as a categorization for tasks, as that came up in the usability testing with Eggert. Víðir agreed that it was indeed necessary to add that categorization. He did not believe a finer grained access control would be beneficial. He agreed that accepting plans would be enough to ensure quality, and having to assign tens of different permissions would only complicate things. This is in line with what has been discussed with Guðbrandur, and my personal view. Víðir also said that there was nothing wrong with users reading other chapters and maybe pitching in fixing grammatical errors. Furthermore, if someone makes changes, that should not be doing so, it could easily be spotted, and the changes viewed and then removed.

One idea that Víðir come up with was that Neyðarlínan (National emergency number) would add a link to the tasks page in Preparedness, to the SMS they send out when disaster strikes.
Finally Víðir said he was extremely excited to start using the system, that it is brilliant, and would be a revolution in the way they create preparedness plans.

### 3.7.9 Results and possible improvements

All in all, the usability testing went very well. Although there were some issues, most of them are very easily fixable and they are simply good additions to the system. Both Eggert and Víðir were very positive towards the system, and excited to start using it.

Changing the way we handle permissions seems, for now, not something to change without further discussions. The system cannot be too complex, or it will completely miss its mark.

A way to create Specific chapters will be implemented. The reason they were not creatable to begin with was partly because additions of new ones are very rare. Also, it was good to have control over their names, because chapter names are inflected in some texts, such as Breytingar hafr verið gerðar á innganginum ... , where innganginum is an inflection of Ingangur. This however, is not that important. Furthermore, the functionality to define new chapters will at some point be added, and the inflections will no longer be possible.

Now we will go over which changes will be made, and the difficulty of their implementation.

### 3.7.10 Changes to make

The following changes will be made to the system, starting with the less time consuming ones.

**Changes that are easy to make**

- Label or tooltip in *Create preparedness plan* modal that tells the user that if the desired location or type does not exist, then it can be added. Either explain where the user interface to do so is, or provide a link.
- Put focus on title input in *Create preparedness plan* modal upon opening it.
- Split Editors panel into two panels. One for accepting/rejecting the current version, and one for publishing the preparedness plan.
- Highlight the links to the two panels, or otherwise make them stand out more.
• Tooltip to explain the To IMS option in the task creation form.

• Highlight description field for tasks, after one is created. Move focus to that input field upon creation.

• Rename Skilgreina verkefni (e. Define tasks) button to Verkefni viðbragðsadila (e. Emergency responders’ tasks).

• Change rendering of PDF so that in the Emergency responders’ tasks chapter, the name of each responder is in a heading, and a table is rendered for the tasks defined for each responder, below the responder’s name.

• Change the name of the column Útgáfunúmer (e. Version number) in My plans, to Útgáfunúmer kafla (e. Chapter version number).

Changes that are of moderate difficulty

• Adding a way to define Specific chapters. Fix texts that depend on using the correct inflection of a chapter’s name.

• Add Umfang categorization to tasks form, the tables in the PDF and as a filter in the view for responders to see their tasks.

The estimated time to make the aforementioned changes is two weeks.

Now we have covered the background, implementation, and evaluation of Preparedness. In the Conclusions chapter we will discuss further possible future work to be done for Preparedness. In the next chapter, we will cover the background, implementation, and evaluation of the Search area and evacuation management component.
Chapter 4

Search area and evacuation management

In this chapter we will go over the background, implementation and evaluation of the second thesis contribution and a new addition to the SAReye IMS, the Search area and evacuation management component.

4.1 Background

Up until now, there has been no connection between the search areas drawn for rescue operations, or the areas drawn for evacuation, and the IMS that manages them through tasks and groups.

In the past, according to Guðbrandur Órn, search areas have usually been drawn using dedicated GIS software, such as Garmin BaseCamp, ArcGIS, Garmin MapSource, OziExplorer. Sometimes they are even simply drawn by hand on maps and then photographed. All the areas are then exported to a document, and uploaded and attached to an event or task in the SAReye IMS.

The problem with this approach is the lack of efficiency and overview over the areas inside the SAReye IMS. The areas in the files have no real connection to the tasks that are created for them. This means that it has not been possible to easily see which tasks correspond to which areas, or which groups are assigned to which areas. Furthermore, because of this lack of connection between the two, it has not been possible to draw statistics from the SAReye IMS, such as total area size, size of searched or evacuated areas and the volunteers’ estimated contributions to the work done.
Lastly, because the volunteers are only assigned to areas inside the SAReye IMS, not the chosen GIS software, there has been no way of creating personalized views of the areas for volunteers. This means that it has never been possible for volunteers to view, online, the areas assigned to their group. This has been done with printed handouts instead.

### 4.1.1 Thesis contribution

This thesis introduces a novel solution, where search and evacuation areas are now drawn, and saved inside the SAReye IMS, with tasks created automatically for the areas drawn. The tasks are named after their respective area’s number, and this number links the two together, in case the name a task is changed.

For search operations, the probability circles can now be defined and displayed on the map, to aid in the drawing of search areas.

To solve the problem of overview, the areas are color coded depending on their associated task’s status. Furthermore, some basic statistics are available on the areas, and on the volunteers assigned to them.

Finally, to solve the problem of having to hand out printouts, a personalized mobile view is now available to volunteers, where they can see the areas they have been assigned to through their groups. They can zoom in and out on the map, and pick overlays depending on terrain, making the solution much more flexible than printed handouts.

### 4.2 Overview

Now that we have covered the background for the Search area and evacuated management component, we can take a look at the implementation.

After showing some of the basic functionality on a blank example operation, we will use one of ICE-SAR’s practice search operations as an example to show some of the more advanced features.

We will only take a look at how the Search area and evacuation management component is used in a search operation, as the usage is almost identical during evacuations. The only two differences are:

- What the areas are created for; during evacuations they are areas that need to be evacuated and during search operations, they are areas that need to be searched.
• *Hasty search* areas (section 4.4.2) are only relevant during search operations.

In figure 4.1 we can see the, now blank, area management for our example operation.

![Figure 4.1: The empty map before any areas have been drawn.](image)

### 4.3 Map and tile layers

The map in figure 4.1 is a part of the open source *JavaScript* library *Leaflet*. *Leaflet* provides functions similar to those provided by *Google Maps* for displaying data and images on maps. What is does better than *Google Maps* is provide a way to change the underlying tiles for the map. This means that we can use any tile layer and add more in the future. As of now, users can choose between using the *OpenStreetMap* tile layer, and the tile layer provided by *Ískort*, which includes more detailed topology information and more accurate contour lines, albeit only for Iceland.

![Figure 4.2: An example of how detailed the Ískort tile layer is.](image)

*Google* does not provide their tiles for use in other map software.
**4.4 Drawing areas**

There are four ways to draw areas; hand drawing polygons on the map, hand drawing lines on the map, drawing using a KML file and drawing using a GPX file. We first go through how to draw using the these methods, and defer code samples until section 4.4.4.

### 4.4.1 Hand drawing areas

It is important to note that the first implementation of drawing polygons on a map, but not saving or displaying them, is based on a group project. Víðir Orri Reynisson wrote that part. Earlier implementation however used *Google Maps*, and has largely been rewritten for *Leaflet*, though part of the logic is still Víðir’s work. He has given consent for its use.

Areas can be drawn on the map using the mouse. The user must first click *Hand draw areas*. This is done because people are so used to using double-click to zoom in on the map. Double clicking was used to start the drawing process, but now only does so if the user has clicked *Hand draw areas*, if not, double-clicking just zooms in on the map.

After selecting *Hand draw areas* and double clicking on the map, the user can start drawing a polygon, in the following manner: After the first double-click, a line follows the mouse on the map (called rubber banding \(^1\)). The line is ended, and a new one started, by double-clicking again. To close the polygon, the user right-clicks on the map. A line from the location of the mouse back to the starting point is automatically created. In figure 4.3 we can see a few finished areas, and one in progress.

![Figure 4.3: Three areas have already been drawn and one drawing is in progress.](http://en.wikipedia.org/wiki/Rubber_banding)

---

\(^1\) http://en.wikipedia.org/wiki/Rubber_banding
According to Guðbrandur Örn, there is no consensus in GIS software regarding how to start or stop drawing. Single click to start would not have worked, since dragging the map is necessary. The current implementation is mainly a legacy from the group project. This is discussed further in section 4.11.

After we have finished drawing the areas, we can save them to the database. When Save drawn areas is clicked, all of the areas that have been drawn are compiled into a JSON object using JavaScript. While compiling the JSON object, the size of each area is calculated as well, along with its geometric centroid. After the JSON object has been compiled, it is sent to the server, and the search areas are saved to the database. Each area will be composed of a set of points, one for each vertex in the polygon drawn and one for its centroid, the size of the area and the area’s numeric identifier. After saving, the page loads again with the areas drawn on the map. This can be repeated as often as necessary.

Figure 4.4: The sizes of areas, in square kilometers, and their numeric identifiers are displayed at their centroids.

In figure 4.4 we can see how the areas are displayed on the map after saving them. The numeric identifier of an area, as well as its size in square kilometers, is displayed at its centroid.

4.4.2 Hand drawing hasty search

At the start of a search effort, a so called hasty search or reflex tasking is often performed. Small groups quickly search the places where the lost person is most likely to be. ICE-SAR asked for a new way of drawing search areas, specifically for these types of searches.
For *hasty search* areas, they wanted dotted lines to be displayed instead of polygons. This is because these searches typically involve walking lines; the trail the lost person was most likely to be on, along fences, or straight to some landmark.

The process of drawing the *hasty search* is essentially the same as for drawing the polygons. The only difference is that a blue line follows the mouse instead of a red one, and when the drawing is ended, using right-click, there is no extra line automatically drawn back to the beginning. In figure 4.5 we can see a *hasty search* area being drawn, and in figure 4.6 we can see the *hasty search* area after saving the two lines.

![Figure 4.5: When drawing the hasty search, the line is blue instead of red.](image)

The process of saving *hasty search* areas is the same as for the polygons, except that neither the area of the line nor the centroid are calculated. Instead of the centroid, the first vertex the user drew is saved, and used instead to display the area’s numeric identifier.

![Figure 4.6: The hasty search is represented by dotted lines instead of polygons. The numeric identifier is displayed at the beginning of each line.](image)
4.4.3 Drawing using KML and GPX

The incident managers at ICE-SAR have already mapped out search areas for some of the more densely populated areas in Iceland, most notably in and around Reykjavík. Hundreds of areas have been drawn, and it soon became clear that it was absolutely necessary for them to be able to use these pre-drawn areas to save time.

These areas where created in the software they previously used, such as Garmin BaseCamp, Garmin MapSource, OziExplorer. These areas therefore existed in GPX and KML formats. Examples of such files can be seen in figures 4.7 and 4.8

Figure 4.7: Example from a KML file.

```
<trkpt lat="64.1138640" lon="-21.7833370">
  <ele>77.4</ele>
  <time>2014-06-14T10:22:42Z</time>
  <extensions>
    <gpxtpx:TrackPointExtension>
      <gpxtpx:atemp>17</gpxtpx:atemp>
    </gpxtpx:TrackPointExtension>
  </extensions>
</trkpt>
<trkpt lat="64.1138780" lon="-21.7829930">
  <ele>76.0</ele>
  <time>2014-06-14T10:22:46Z</time>
  <extensions>
    <gpxtpx:TrackPointExtension>
      <gpxtpx:atemp>17</gpxtpx:atemp>
    </gpxtpx:TrackPointExtension>
  </extensions>
</trkpt>
```

Figure 4.8: Example from a GPX file.

Therefore, a way in which to use these files for drawing was implemented. The user clicks either Draw using GPX file or Draw using KML file, and selects the desired file. As soon as the file has been selected, it is parsed, and all of the areas are drawn on the map.
The reason that the areas are drawn on the map instead of uploading the file, is that is was extremely practical to reuse code already written. By drawing the areas from the GPX and KML files on the map, we can reuse all of the code used for saving the areas, including calculating the centroid and the sizes of the areas. It also nice to be able to draw using the files, and then add a few hand drawn areas afterwards, before saving the areas. In figure 4.9 we can see the map after using the KML in figure 4.7 for drawing, and in figure 4.10 the results after saving the areas.

![Map with drawn areas](image1.png)

Figure 4.9: *The dozens of areas from the KML file have been drawn.*

![Map with saved areas and zoomed in](image2.png)

Figure 4.10: *The areas from the KML file have been saved, and the map zoomed in a bit.*

### 4.4.4 Code for the four methods of drawing

The following figures show some of the code used in the four drawing methods. The first two show code that only handles drawing, and the last one shows how the JSON object for the drawn areas is compiled and then sent to the server.
Figure 4.11: Mouse events for drawing on the map.

Figure 4.12: KML and GPX parsing and area drawing.
Figure 4.13: Compiling the areas drawn into a JSON object and calculating area sizes and centroids.

### 4.5 Probability circles

For almost the entire remainder of this chapter, we will use one of ICE-SAR’s practice search operations, *Leitaræfing Björgunarfélags Hornafjarðar*, to demonstrate features.

To recap, the probability circles are centered around the initial planning point (IPP), and are determined using statistical information from the ISRID database, and academic analysis by Robert J. Koester. They are drawn for 25%, 50%, 75% and 95% probabilities, along with a 300 meter circle around the IPP. The 25% circle, for example, represents the area within which 25% of those found were located. Search areas are drawn on-top of these probability circles, and they are used to prioritise the search effort.
In figure 4.14 we can see the form for creating the probability circles. The IPP is given in WGS84 format, and becomes the center of the concentric circles. The radius of each circle is given in kilometers, but it is not a requirement to enter a value for each one.

After saving the probability circles the page is reloaded and the circles are drawn on the map. The relevant percentage is drawn on each circle, true north of the IPP. Each circle is filled with blue color, which become layered so that the inner most circles are darker than the outer most ones, which makes them easier to distinguish from each other.

Figure 4.14: The form for creating probability circles, and zoomed out view of the probability circles atop of the Ískort tile layer.

Figure 4.15: A closer look at the 300m, 25%, 50% and 75% circles.
Figures 4.16 and 4.17 show the code that takes care of displaying the probability circles.

```python
def search_areas
    @op = Operation.find(params[:operation_id])
    ...
    @ipp_data = []
    if not @op.ipp_data.nil?
        @ipp_data << @op.ipp_data.ipp_lat
        @ipp_data << @op.ipp_data.ipp_lng
        @ipp_data << @op.ipp_data.twenty_five_percentile
        @ipp_data << @op.ipp_data.fifty_percentile
        @ipp_data << @op.ipp_data.seventy_five_percentile
        @ipp_data << @op.ipp_data.ninety_five_percentile
    end
    ...
end
```

Figure 4.16: The server side code that takes care of fetching the probability circles for the operation.

```javascript
function displaySearchAreas()
    var ipp_data = null;
    if (ipp_data.length === 0){
        ipp_data = [];  
        var marker = new L.Marker(geoJSON(ipp_data[0]), parseES6(ipp_data[1])).addTo(map);
        var circle = L.circle(geoJSON(ipp_data[0]), parseES6(ipp_data[1]), 300, 
            {fillColor: "#206890", fillOpacity: 0.3}).addTo(map);
        addCircleLabel(ipp_data[0], ipp_data[1], ipp_data[3], "25%");
    }
    if (ipp_data[2] === 0){
        L.circle(geoJSON(ipp_data[0]), parseES6(ipp_data[1]), ipp_data[2]=1000, 
            {fillColor: "#206890", fillOpacity: 0.3}).addTo(map);
        addCircleLabel(ipp_data[0], ipp_data[1], ipp_data[3], "50%");
    }
    if (ipp_data[3] === 0){
        L.circle(geoJSON(ipp_data[0]), parseES6(ipp_data[1]), ipp_data[3]=1000, 
            {fillColor: "#206890", fillOpacity: 0.3}).addTo(map);
        addCircleLabel(ipp_data[0], ipp_data[1], ipp_data[3], "75%");
    }
    if (ipp_data[4] === 0){
        L.circle(geoJSON(ipp_data[0]), parseES6(ipp_data[1]), ipp_data[4]=1000, 
            {fillColor: "#206890", fillOpacity: 0.3}).addTo(map);
        addCircleLabel(ipp_data[0], ipp_data[1], ipp_data[4], "95%");
    }
    if (ipp_data[5] === 0){
        L.circle(geoJSON(ipp_data[0]), parseES6(ipp_data[1]), ipp_data[5]=1000, 
            {fillColor: "#206890", fillOpacity: 0.3}).addTo(map);
        addCircleLabel(ipp_data[0], ipp_data[1], ipp_data[5], "99%");
    }
}
```

Figure 4.17: The client side code that takes care of drawing the probability circles for the operation. The ipp_data array contains the following data, in the following order: Centroid’s latitude and longitude, and if defined, the radiiuses of the 25%, 50%, 75% and 95% circles.
4.6 Connection between tasks and areas

There is more to the Search area and evacuation management component than drawing on maps. A task is created for each area that is drawn and saved, and a very useful connection is maintained between the two.

4.6.1 Task names and area identifiers

The description of the task created for an area is, by default, Svæði <number> (e. Area <number>), where <number> is the numeric identifier of the area. If the description of the task is changed at some point, we still want to be able to see which area is associated with it. We can do so be clicking the task in the tasks table, either under Framgangur or the tasks area within the Search area and evacuation management. The latter is shown in figure 4.18, and the modal that opens for the task in figure 4.19. Below the task’s description, in figure 4.19, we can see the associated area’s number and size. This number is the one that appears at the area’s centroid on the map.

Figure 4.18: The tasks table at the Search area and evacuation management shows all tasks, not only those that have areas associated with them.
Figure 4.19: A task that is associated with an area, and has had its description changed. Note also that the task’s coordinate is, by default, the associated area’s centroid, unless it is changed.

Figure 4.20: Creating areas and tasks and a connection between each pair.
Finally, we can see the code that takes care of creating all the areas that were sent to the server in figure 4.13, and the associated tasks, here in figure 4.20 above.

### 4.6.2 Associated tasks and areas and their statuses

The most important connection between tasks and areas is through their statuses. Areas can be finished (searched or evacuated) and not finished. As we went through in the background, tasks can be done, in progress (have groups), or not in progress.

When a task is marked as done, then the associated area, if one exists, is marked as finished (searched/evacuated) as well. If an area is marked as finished, through the interface in figure 4.18, then the associated task is marked as done as well.

As we saw in figure 4.15, some of the areas on the map are colored green, some yellow and some red. Each color has a meaning:

- A green area has been finished (searched/evacuated), and therefore its task is done as well.
- A yellow area has a task associated with it that has had groups assigned to it, so the task is in progress.
- A red area is not searched, and has a task associated with it that has no groups assigned.

When an area is restarted, it is marked as not finished (searched/evacuated), and the associated task marked as not done, and when an area is deleted, the associated task is too, and vice versa. Tasks can be marked as done, and then restarted many times. We maintain a count of how many times a task has been finished, which is, according to Guðbrandur Örn Arnarson, very important for analysis and post-mortem purposes.

### 4.7 Statistics

We want to be able to see, roughly, how large a portion of a search or evacuation effort each resource, volunteer or vehicle, has taken part in. Therefore, each time a task that has an associated area is marked as done, or an area is marked as finished, we allocate a portion of the work done to each resource in the groups that were assigned to the task.

As an example, if the area was two square kilometers, and it had two groups assigned, each with four resources, then each resources gets credit for 0.25 square kilometers. This
is only an estimate, but if we start to see that some responders have a load many times that of others, there might be something wrong. Mainly, this functionality was written with future work in mind, and is the first step in an overview of the resources’ workload.

By clicking the Statistics tab, we can see a pie chart that shows the estimated work load of each responder that has taken part, as well as the total sizes of all the areas, and only the finished and not finished areas. This is shown in figure 4.21. In figure 4.23, we can see the code that takes care of rewarding work loads to responders. In figures 4.24 and 4.22 we can see the code that takes care of gathering all this data and displaying it, using the Google Visualization library.

Figure 4.21: The statistics for the work load in the search effort, shown using a pie chart.

```javascript
var drawChart = function () {
  var draw = google.visualization.draw();
  drawchart();
};

var drawchart = function () {
  var data = google.visualization.arrayToDataTable(data.responder_areas);
  var options = {
    chartArea: {width: 256, height: 256},
    pieSliceText: 'value'
  };
  var chart = new google.visualization.PieChart(document.getElementById('pie_chart'));
  chart.draw(data, options);
};
```

Figure 4.22: The client side code that displays the data, partly using the Google Visualization library.
Figure 4.23: Assigning parts of the work load to resources.

Figure 4.24: The server side code that gathers the data for the statistics.
4.8 Calculating resource needs

The resource needs part of the *Search area and evacuation management*, seen in figure 4.25 is used to calculate how many volunteers are needed to cover the 300 meter circle and 25%, 50%, 75% and 95% probability circles respectively.

![Calculator for resource needs](image)

**Figure 4.25:** The calculator for resource needs. The calculations themselves are written by Guðbrandur Órn Arnarson.

The calculations themselves are written by Guðbrandur Órn Arnarson, and are therefore not a part of this thesis contribution, although the design of the view is.

Just to clarify, *range* is how far those searching can see in each direction, and is estimated using a so called *rain dance*. The radiuses of the 25%, 50%, 75% and 95% probability circles are automatically filled in, if the circles have been drawn on the map.

4.9 Instructions

We will not go into any details about the instructions, only note that they are present. The instructions explain how to draw using the four methods covered in this chapter, and they give a quick overview of the connection between tasks and areas.
4.10 Personalized mobile view

Up until now, volunteers have been given printed handouts with the areas they should cover, along with basic information about the operation.

Now, using the mobile version of the SAReye IMS, volunteers can get a personalized view of the areas. In this view, only the areas and tasks that are assigned to the groups that the volunteer belongs to, are displayed. The map works exactly like it does in the desktop version, and the volunteer can zoom in and out on the map. This is a much more flexible, economically friendly (no printing), and faster way of distributing the areas to those responsible. The position of the volunteer is also shown on the map, if the volunteer has GPS turned on on the phone.

The personalized view is shown in figure 4.27, although a volunteer’s position is not shown is this example. The code that takes care of filtering the areas is in figure 4.28. Note that this is not the ICE-SAR operation, but the example operation from before instead, since to view the personalized view one needs to be a registered volunteer and of course assigned to a group in the operation as well.
Figure 4.27: *Mobile view of the areas that have been assigned to the volunteer’s groups.*
Ásgeir Jónasson

Figure 4.28: The code that fetches all the areas and tasks that are assigned to the current user.

This was the last feature of the Search area and evacuation management left to cover, and in the next section, we will evaluate the Search area and evacuation management component by going over the input that has come from users during its development.
4.11 Evaluation

The way the Search area and evacuation management component was developed and evaluated was quite different from the way it was with Preparedness.

4.11.1 Method

Instead of writing a complete tool and then presenting it to the users, the component has been constructed in collaboration with incident managers at ICE-SAR from day one, with features being requested and continuously refined through user feedback, relayed from Guðbrandur Örn.

The component was put into use as soon as it was a viable product. After using the component, both during free time and actual operations, users gave Guðbrandur feedback and ideas for additions, which he then relayed back to me. I implemented new features as soon as I could, with users trying them out the next time they needed to use the component. This circle, and continuous refinement, continued until we had the product we have today.

Although I did not get the feedback directly from users, Guðbrandur made a great job of relaying feedback to me. He was also able to put the requests and comments into language I could understand, and discuss them in the context of the SAReye IMS, of which he has expert knowledge. He knows a lot about the intricacies of the SAReye IMS, and knows what is possible.

This method I have just described made sure that unnecessary features where not present in the Search area and evacuation management component, and that those features implemented where in line with the needs of the incident managers at ICE-SAR.

4.11.2 Timeline of the development and evaluation

The users were first provided simply with a map to draw areas on, they could not even be saved. That version used Google Maps, instead of Leaflet. The page had a disclaimer at the top, that the component was under construction, but that users were welcome to play around, and that any feedback was welcome. The steps in development described in the following paragraphs are shown visually in figure 4.29.

A way to save search areas was implemented, and from there on almost every feature was implemented by request, and according to spec. The two features that were not ideas
from users were statistics on the workload of different volunteers and displaying the sizes of search areas at their centroid.

The first feature requested was the connection between areas and tasks, and the color indicators, green, yellow and red for showing the status of the task associated with an area. Seeing the associated area’s number after clicking a task in the task table was also by request.

Next the functionality to see an area’s size on the map was implemented, along with the statistics for total-, searched- and unsearched- area sizes.

After a rather long break in development, the map for the *Search area and evacuation management* was rewritten to use the *Leaflet* framework, and the *Ískort* tile layer added.

The *probability circles* have been drawn and used extensively in other map software solutions used by ICE-SAR, and they therefore asked for a way to draw them on the map. Since they only require a radius for each percentage and a centroid, only asking for those values was necessary.

Drawing using *KML* and *GPX* files were feature requests that came about when *ICE-SAR* incident managers realized that a lot of work has gone into creating areas already, using other software solutions, work that they wanted to be able to use.

The first implementation of the *probability circles* had the form above the map. This however caused a confusion, as users drew areas, and then used the save button for *probability circles* to save the areas. This caused a page refresh, to save the circles, and the loss of their work.

Therefore, a new interface was designed, where each feature is behind a separate tab. Instructions, which before appeared in a modal, got their own tab. Before, all the features shared the same space.

According to Guðbrandur Örn, incident managers have used past version of the resource calculator extensively, as it can be found in an excel file, and on the web. However, it has never been available before inside their map software, and it was deemed a good idea to have it there, so as to remind incident managers to check their resource needs estimates.

The last drawing functionality to be implemented was the ability to draw hasty search areas, and the last functionality to be implemented was the personalized mobile view.

One particular feature should maybe have gone through a more structured usability testing. As mentioned in section 4.4.1, it would have been nice to put more thought into how to start and stop drawing. However, as mentioned, there is no consensus on how to do
this, so any method would need to be learned. Furthermore, there is a limited number of possibilities, and using only the mouse, the only other possibility would have been to swap double-click and right-click. In any case, it would still be interested to research what feels most natural to users. Changing this implementation now that all users at ICE-SAR have learned to use the tool, does not seem practical.

Figure 4.29: Timeline depicting the development of the Search area and evacuation component.

4.11.3 Summary

Since its debut in September 2014, when it was only a map for drawing areas on, Guðbrandur has been collecting feedback from incident managers on the Search area and evacuation management component, and relaying it to me along with feature requests. Therefore, the component has been constructed in collaboration with incident managers from day one, and has almost all the features they need, and none that they don’t. Both me and Guðbrandur Örn have been very satisfied with this approach. It is Guðbrandur’s view that I learned just enough to understand their needs, but in the meantime, approached the problem with a fresh perspective and was able to solve some problems in a better way than what they are used to do.

Being an outsider to the problem domain, with guidance from Guðbrandur, has led to a more general, and abstract tool. This is the reason behind the component also working very well for evacuations. It has now also started to be useful for Landsnet’s operations,
who also have an \textit{SAReye IMS}, for example when marking danger zones, and searching for ruptured power lines. In Guðbrandur’s words, I knew just little enough about the problem domain, to not be stuck inside the incident managers’ mindset, and be able to think more abstractly about the problem. At the same time, Guðbrandur was able to make sure we still had a product that met the needs of the incident managers atICE-SAR.
Chapter 5

Conclusions

In this chapter, we will discuss lessons learned, both technical and personal, the current status of the projects and how successful they were, and the possibility for future work.

5.1 Lessons learned

5.1.1 Personal lessons learned

It has been an extremely valuable lesson to take a software project, from just an idea for a solution to a problem, to a full-fledged system. Making technical decisions, gathering feedback, and not being able to rely on anyone else during the development has been a great personal lesson, and confidence inspiring.

5.1.2 Technical lessons learned

Making all technical decisions has of course also come with many technical lesson learned. From the start, I have wanted to keep the code clean, and to do as much as possible by the book. It has sometimes been difficult not to opt for the easy way, but always ended up paying off, and affording another lesson each time.

Since there has been limited time to implement features in both thesis contributions, this masters project has been a great lesson in utilizing libraries written by other developers, and connecting together tools to make something that’s greater than the sum of its parts.

The usability tests for Preparedness were also extremely valuable. I have done usability tests similar to this one in classes at Reykjavik University, but it was not until using
think-aloud on a project that one is so invested in, that I realised its full potential. It was especially valuable to see the smaller issues, the ones that might be annoying to users, but take only minutes to fix. I would definitely not have discovered some of them on my own.

5.1.3 Success of thesis contributions

The Search area and evacuation management component has been used at ICE-SAR now for a few months, and the incident managers are very happy with the results. The component has been used in high stakes search efforts, where there was a very real danger. Trust in the component is therefore not lacking. According to Guðbrandur Órn, the incident managers at ICE-SAR are very happy with the component, and no less so are the volunteers that can get personalized views of search areas.

Both Víðir Reynisson and Eggert Magnússon were very excited to start using Preparedness, and Víðir called it a revolution in their creation and management of preparedness plans. They have already bought the system from SAReye. They will soon get a seminar in how to utilize the system to its fullest potential, and afterwards will start writing preparedness plans using it. Hopefully though, they will not initialize many of the plans in the SAReye IMS.

5.1.4 Future work

Preparedness

Apart from the changes mentioned in section 3.7.10, there are couple of more extensions that can be made.

One is to enable hiding of preparedness plans on the front page. This would be a rather simple to implement feature.

Another type is to make it possible to have more data come over from Preparedness to the SAReye IMS. The biggest improvement here would be to put the Search area and evacuation management component into Preparedness. This would enable pre-drawing evacuation areas.

Finally, at some point, it should definitely be possible to create chapters dynamically. This would also enable SAReye to sell the software to other institutions and companies, without any further changes to the functionality.
Search area and evacuation management

One possible addition to the Search area and evacuation management component would be to enable editing of areas. Currently, it is possible to just delete an area and draw a new one. The only problem with this is that the associated task gets deleted as well. This could be averted by offering the choice of not deleting the associated task, and then manually connecting the task and a new area. Then a new area could be created and connected to the old task that had its area deleted. In any case, this is not a high priority request, especially not when considering it is rather time consuming to implement.

5.1.5 Final words

In February 2008, me and three friends of mine were rescued by ICE-SAR at Mt. Esjan during exceptionally bad weather conditions.

The situation was very dire, and we are extremely thankful today for the help that we got.

Early this year, one of the most satisfying moments of my professional career, and my life, took place when the ICE-SAR incident managers utilized my component in a search effort, at exactly the same place where I had been lost. They were able to successfully retrieve the injured and lost person.

I felt that I had finally been able to pay them back for their rescue.
Bibliography


Appendix A

If the user performed a task without problems, then the relevant field says *Nothing to report*. If there was a problem, then it is mentioned, and if a discussion followed, it is outlined in the same field. In the rightmost column, there is a possible solution to the problem, if one exists and is feasible.

Table A.1: Results from the Preparedness usability testing

<table>
<thead>
<tr>
<th>#</th>
<th>Step</th>
<th>Eggert Magnússon</th>
<th>Víðir Reynisson</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Registration</td>
<td>Nothing to report</td>
<td>Nothing to report</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Create a new preparedness plan, for a type and location that does not exist yet.</td>
<td>Did not think that it was possible to create new location and type, only selecting existing ones. In the end, had to tell him that it was possible to create new ones. Tried writing in the type dropdown to create a new type. Did not figure out that he had to close the modal to get to the create type/location page, only tried manipulating inputs in the modal in multiple ways. In the end, had to tell him to close modal. Then he found the link.</td>
<td>Nothing to report</td>
<td>Label or tooltip, inside the modal, explaining where to add the desired type/location that doesn’t exist would probably suffice.</td>
</tr>
<tr>
<td></td>
<td>Creating type and location.</td>
<td>Nothing to report</td>
<td>Nothing to report</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------</td>
<td>-------------------</td>
<td>-------------------</td>
<td>---</td>
</tr>
<tr>
<td>3</td>
<td>Finish process of creating a new preparedness plan</td>
<td>Discussion: Asked whether he could create new Sértekur kaflí (e. Specific chapter) to choose from. When told no, he said that it should probably be possible.</td>
<td>Started typing without clicking the title input field.</td>
<td>Discussion on possibility of creating new specific chapter is in section 3.7.9. Focus should on title input.</td>
</tr>
<tr>
<td>4</td>
<td>Add me as an editor to the plan</td>
<td>Did not find the editor’s panel. Tried going to the header dropdown and to my plans. Also tried the front page and clicking the plan there and different chapters inside the plan’s editor. Gave up and had to be told where to do this. Discussion: Said that this tab looked too much like the chapter tabs, and should stand out.</td>
<td>Found Editors panel, and Authors right away, and how to add authors, but took a bit of time finding the Editors switch, to add editors instead of authors.</td>
<td>Highlight Editors panel in some way, or at least style differently from chapter tabs. Have separate panels for adding authors/editors and publishing, where those functions stand out more.</td>
</tr>
</tbody>
</table>
Table A.1: Results from the Preparedness usability testing

<table>
<thead>
<tr>
<th></th>
<th>Add text to Introduction and save</th>
<th>Nothing to report</th>
<th>Nothing to report</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>View changes made by another author in Staðhættir.</td>
<td>Nothing to report</td>
<td>Nothing to report</td>
</tr>
<tr>
<td>7</td>
<td>Write a new version of the chapter with an earlier version as a base.</td>
<td>First tried opening modal displaying changes View changes and clicking the text there. However, he found the right way soon after.</td>
<td>Nothing to report</td>
</tr>
<tr>
<td>8</td>
<td>Create two tasks for emergency responders</td>
<td>Thought at first that he could create multiple tasks with newlines (or bullet points) in description field. <em>Discussion:</em> after figuring it out thought it was actually a fine implementation. Mainly he was used to writing many tasks at once in bulleted lists in Word. Thought that the structure of the form was clear and the To IMS checkbox as well, but suggested a tooltip there just in case. Suggested keeping To IMS checked.</td>
<td>Thought he needed to press Define tasks again after saving one, to define a new one. Suggested having the description field highlighted after save. Mentioned that the To IMS field would probably almost always be ticked, so it should stay on, which it does not now. User also went back to Free text, and then mentioned that Skilgreina verkefni (e. Define tasks) might confuse users that only want to view the tasks.</td>
</tr>
</tbody>
</table>
Table A.1: Results from the Preparedness usability testing

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Add free text to <em>Verkefni viðbragðs-aðila</em></td>
<td>Nothing to report</td>
<td>Nothing to report</td>
</tr>
<tr>
<td>11</td>
<td>Add a communication method</td>
<td>Nothing to report</td>
<td>Nothing to report</td>
</tr>
<tr>
<td>12</td>
<td>Publish the plan</td>
<td>First went to My plans, Admin panel, About me, and eventually got back to the Editors panel, and was able to publish the plan.</td>
<td>Nothing to report.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Highlighting <em>Editors panel</em>. Possibly splitting up into <em>Authors/Editors panel</em> and <em>Publish panel</em>.</td>
</tr>
<tr>
<td>13</td>
<td>View PDF version of the newly published plan.</td>
<td>Found PDF quickly. Discussion: An idea was to display the tasks not in a table but rather with a hierarchy of headings, like they use now. Was not certain though, but said that at least each emergency responder should be a heading, with a table below for each one, and to remove the responder’s column instead. There was also a discussion here that there was one more categorization missing for tasks, called <em>Umfang</em> in Icelandic.</td>
<td>Emergency responder column should come first. Discussion: Mentioned Egger’s ideas on having a heading for each responder, and then a table below each responder. Said that it might be good, but not necessary, but making the emergency responders column be the first one would be much better than the current implementation. When prompted, agreed that <em>Umfang</em> was indeed missing.</td>
</tr>
</tbody>
</table>
Table A.1: Results from the Preparedness usability testing

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>Go back to the front page.</td>
<td>Nothing to report</td>
<td>Nothing to report</td>
</tr>
<tr>
<td>15</td>
<td>Go back to working on preparedness plan.</td>
<td>Nothing to report</td>
<td>Nothing to report</td>
</tr>
<tr>
<td>16</td>
<td>Go back to the front page.</td>
<td>Nothing to report</td>
<td>Nothing to report</td>
</tr>
<tr>
<td>17</td>
<td>View a PDF version of the plan.</td>
<td>Nothing to report</td>
<td>Nothing to report</td>
</tr>
<tr>
<td>18</td>
<td>View changes made to all plans you are editing</td>
<td>User had found this page before, so he found it immediately</td>
<td>Found the page quickly. Discussion: Almost all columns are clear, although first intuition was that Chapter version, which is not as clear in Icelandic, being Útgáfumúmer (e. Version number), was construed to mean the relevant plan’s version, not the chapter version</td>
</tr>
<tr>
<td>19</td>
<td>Find a task for a specific responder on /verkefni ¹.</td>
<td>Found the page to be excellent, only Umfang was missing as a task filter.</td>
<td>Thought the page was very good. Agreed Umfang was indeed missing.</td>
</tr>
</tbody>
</table>

¹ There is no link to this page as of yet, so the URL was entered for them. It has not been decided whether or not users should be redirected automatically to this page if on mobile. Or whether the link should be like the other ones on the frontpage, or an icon. Adding one, when decided, will only take minutes.
Implementation of end-to-end incident management processes in a unified IMS