HoliRisk - Risk Management Reporting

Diogo Manuel dos Mártires Estevens

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor(s): Prof. José Luís Brinquete Borbinha

Examination Committee

Chairperson: Prof. Paolo Romano
Supervisor: Prof. José Luís Brinquete Borbinha
Member of the Committee: Prof. Miguel Nuno Dias Alves Pupo Correia

October 2017

To perseverance

Acknowledgments

Firstly, I would like to mention and thank Professor José Borbinha for the opportunity to be part of this work and his guidance throughout the all dissertation. I would also like to thank António Higgs and Ricardo Vieira, for their constant help and advice given, in order to overcome technical and theoretical challenges.

Secondly, I would like to acknowledge Eduardo Melo with whom I had the pleasure to work with during this dissertation.

Last but not least, I would like to express my sincere gratitude to all my family and friends for the motivation, support and encouragement without which this accomplishment would not be possible.

Abstract

Over the years and even nowadays, enterprises do not follow standards, when it comes to Risk Management, and end up using spreadsheets to identify, register and manage risks. These spreadsheets are then used to manage the risks of the company, risks of the departments and even risks regarding activities, since they may have a different manner on how to be managed. Thus, it can be difficult to integrate the risks altogether and consequently, the vision regarding said risks becomes fragmented and dispersed throughout several contexts.

HoliRisk is a web-based platform that aims to support the risk assessment process of any organization, independently of its focus, objectives or context. HoliRisk makes it available a holistic centralized view of risks where it is possible to register risks regarding the enterprise as a whole, its departments or even activities, in an integrated and centralized manner.

Even though HoliRisk is a holistic risk assessment tool where it is possible to register risks and see all the data inserted, there was no actual component that analysed the data and provided useful information regarding the existing data. This work aimed to implement that data analytics component, named Reporter, allowing the creation of multiple types of reports using multiple risk assessment techniques. This Reporter's objective is to improve the way that HoliRisk helps any enterprise to thrive, giving its users the knowledge and information necessary for success.

Keywords: Risk Management, Risk Assessment, Data Analytics, Reporting, Information

Resumo

Ao longo dos anos e até hoje, empresas não respeitam os padrões definidos quando se trata de Gestão de Riscos, o que leva à utilização de folhas de cálculos para identificar, registar e gerir riscos. Essas folhas de cálculo são usadas para gerir os riscos da empresa, os riscos dos departamentos e até mesmo riscos relativos a atividades, uma vez que podem ser geridos de diferentes formas. Assim, torna-se difícil integrar todos os riscos existentes e consequentemente, a visão sobre esses riscos torna-se fragmentada e dispersa em diversos contextos.

O HoliRisk é uma plataforma web que visa apoiar o processo de apreciação de risco de qualquer organização, independentemente do seu foco, objetivos ou contexto. A HoliRisk disponibiliza uma visão holística e centralizada dos riscos, onde é possível registar riscos relativamente à empresa como um todo, seus departamentos ou até mesmo atividades, de forma integrada e centralizada.

Apesar de a HoliRisk oferecer uma ferramenta holística de apreciação de risco onde é possível registar riscos e visualizar todos os dados inseridos, não existia nenhum componente que analisasse os dados e fornecesse informações úteis relativamente aos mesmos. Este trabalho teve como objetivo implementar esse componente de análise de dados, chamado Reporter, que permite a criação de múltiplos tipos de relatórios usando múltiplas técnicas de avaliação de risco. O objetivo deste Reporter é melhorar a forma como a HoliRisk ajuda qualquer empresa a prosperar, dando aos seus utilizadores o conhecimento e a informação necessários para ser bem sucedida.

Palavras-chave: Gestão de Risco, Avaliação de Risco, Análise de Dados, Informação

Contents

List of Figures

Acronyms

API Application Programming Interface

CRUD Create, Read, Update, Delete

DFS Depth First Search

DOM Document Object Model

ERM Enterprise Risk Management

GRC Governance, Risk Management and Compliance

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

ISO International Organization for Standardization

JSON JavaScript Object Notation

MVC Model View Controller

PDF Portable Document Format

PUR Pizzeria Under Risk

SUS System Usability Scale

UEQ User Experience Questionnaire

UML Unified Modelling Language

UX User Experience

Chapter 1

Introduction

Every enterprise sets targets that need to be reached, in order to be successful. Therefore, it is paramount to manage possible risks that could compromise the achievement of those goals. Formerly, companies would use spreadsheets to identify and register risks and the manners in which to proceed for each one of them. This process is poor, incomplete and is used throughout the company, by departments, projects and even activities making it difficult to integrate all the risks together. Therefore, the vision of the possible risks of the enterprise becomes fragmented and spread throughout various contexts.

Over time, standards were created to identify the proper way to handle and implement risk management in an enterprise. The ISO 31000 [1] emerged establishing Risk Management standards, and most companies tried to adapt using what they had: spreadsheets. Other companies adapted by recurring to outside development or available software. The outside development is done on demand exactly as the customer requests and probably no other company would be able to use. On the other hand, there is Risk Management oriented software that is implemented in a specific way and the customer has to adapt to it, making it difficult for some of the involved to get what they want if something different was desired. These two options seemed to open an opportunity for the development of a software that could be incorporated into whatever organization and be flexible enough for the company to change it if necessary.

1.1 Problem Description

A Risk Management platform named HoliRisk was created in order to be possible for any enterprise to use and configure, instead of having to adapt completely to an existent application or having to define the requirements of a new one. HoliRisk is a web-based platform that allows a holistic view of risks in an organization rather than a fragmented view where risks are dispersed across multiple contexts. Even though the HoliRisk platform is a more flexible and centralized solution where it is possible to register and visualize data, it lacked a data analytics component that could analyse the available data, with the purpose of obtaining valuable business information and provide a better support for decision making.

1.2 Objectives and motivation

This dissertation is meant to provide extend the HoliRisk platform with a module that provides better way to analyse existing data. Thus, the data can be used by a data analytics or reporting module that is going to be addressed as Reporter throughout this document. Besides that, the data being used as the basis for this document is the Pizzeria Under Risk (PUR) which describes a simple scenario regarding risk management of a pizzeria, further detailed in appendix A. The Reporter aims to provide the possibility of creating multiple reports which can lead to multiple conclusions, faster answers regarding the business and better decision making, that could mean the difference between the success or failure of a company.

1.3 Contributions

This work aimed to incorporate a component into the HoliRisk platform that feeds on its data, in order to produce business intelligence that can be used by risk management specialists to take the right decisions at the right time. In the data analytics component added, the Reporter, it is possible to create five types of report (Overview, Risk Matrix, Filtered List, Fishbone and Combined) by configuring them and exploring the respective results. Furthermore, the component is a strong base from where multiple other reports can be created, with the same mission: for the organization to thrive making the right decisions at the right time.

The Reporter interface proved to be useful and friendly in an end-user perspective. The main commentaries of the users were that it was very intuitive even for the users that were not until then familiar with the risk management concept.

This work also contributed with an update to the user manual already created for the HoliRisk platform, with the addition of the Reporter functionalities. Since the manual was already written in Portuguese, it was also carried on in the same language and is available in the appendices.

1.4 Document Structure

The rest of this dissertation is organized in more six chapters and four appendices. Chapter 2 presents some related work to the detailed problem. Chapter 3 states the problem analysis, the platform on which this dissertation would be built on and the requirements that need to be addressed and implemented. Then, in chapter 4 will be depicted the proposed solution. Chapter 5 describes the solution implementation in detail. Chapter 6 describes the evaluation method used, as well as the results of the user testing and respective conclusions. The last chapter concludes the dissertation with an overall discussion of the work done and the future work proposed.

Finally, in the appendices it is available the Reporter User Manual, an example scenario used throughout the dissertation and the guideline and survey presented in the evaluation chapter.

Chapter 2

Related Work

Every business has risks associated and it is necessary to manage them, e.g. if the *chef* of a restaurant were to catch a cold, this would be a problem that the restaurant should anticipate beforehand and have a substitute ready, or establish that the *sous-chef* would substitute the *chef* while he is unavailable. The prevention of this risk or any other requires using a variety of techniques to assess them and come up with a solution. Since there is a large number of possible techniques to be used, an enterprise can use the techniques that better apply to its objective.

Each company has been managing risk as they pleased, some with spreadsheets, others just handling risk when something happens. Therefore, International Organization for Standardization (ISO) standards were created and should be the point of reference for every company that is adopting risk management. This dissertation will be focused and comply with some of the standards created by ISO, mainly ISO 31000 and 31010.

This chapter will introduce the risk management concept, its process and respective risk assessment techniques. Only the techniques that were used and implemented are depicted in more detail.

2.1 Risk Management Core Concepts

Risk management begins with the possibility of an **event** occurring, i.e. a change in a set of circumstances [2]. An event can create the effect of uncertainty on a business' objectives and that effect is called **risk**. Thus, risks have to be managed by defining activities to control that uncertainty and to guarantee that objectives are achieved. These measures that modify the risks are called **controls**.

Every event that may impact an organization's objectives must be taken into account when managing risks. To deal with these risks, the Australian and New Zealand AS/NZS 4360 standard was created in 2004 [3]. Based on that efforts, the ISO 31000 was created, in 2009, and is now the risk management international standard.

ISO 31000 was created so that risk management can be focused on an organization as whole or on smaller parts such as its activities, processes or resources [4]. Organizations should manage risk by firstly identifying it, secondly by analysing it and finally by evaluating whether the risk should be handled

and/or treated [1]. When adopting this standard, a number of **principles and guidelines** need to be followed in order for risk management to be effective, and a systematic and logical **process** has to be adopted. This standard can to an entire organization or to specific functions, projects and activities.

The adoption of this risk management standard enables an organization to improve in multiple manners, such as by increasing the likelihood of achieving objectives; improving the identification of opportunities and threats; improving stakeholder confidence and trust; establishing a reliable basis for decision making and planning; allocating and making use of the resources for risk treatment in an effective way and improving incident prevention and management; improving organizational learning and resilience [1].

In order for this improvement to be possible, each risk must be associated with a person or an entity that is accountable and that has the authority to act upon it. This person or entity, the so-called Risk Owner [1], becomes responsible for the risks and for taking actions, making decisions towards a better solution.

2.1.1 Principles and Guidelines

ISO 31000 contributes with principles and guidelines on risk management and can be applied to any risk, regarding any industry and can be implemented throughout the life of an organization and to specific strategies and decisions, operations, processes, functions, projects, products, services and assets [1].

In order for risk management to be effective, an organization should comply with the principles of the standard ISO 31000: create and protect value; be part of organizational processes, be part of decision making; explicitly handle uncertainty; be systematic; be structured; use the best available information; be tailored; take human and cultural factors into account; be transparent and inclusive; be dynamic; be iterative; be responsive to change; and facilitate continual improvement of the organization [1].

Although this standard provides generic guidelines, it is not supposed to promote uniformity on risk management across organizations, i.e. it must adapt to each specific organization, its particular objectives, context and whatever target risk management standard is focused on[1].

2.1.2 Process

The process of risk management should be a part of management, culture, practices and adapted to the business processes of the organization [1]. The activities of this process are represented in Figure 2.1 and detailed below.

Communication and consultation take place with stakeholders, internal or external to the organization. This stage should be developed early on, to address issues related to the risks themselves, causes, consequences and measures to control the risks. Also, communication and consultation should be effective in order for the stakeholders and those responsible for implementing the risk management process, to understand the basis on which decisions are made and the reasons why specific actions are required [1].

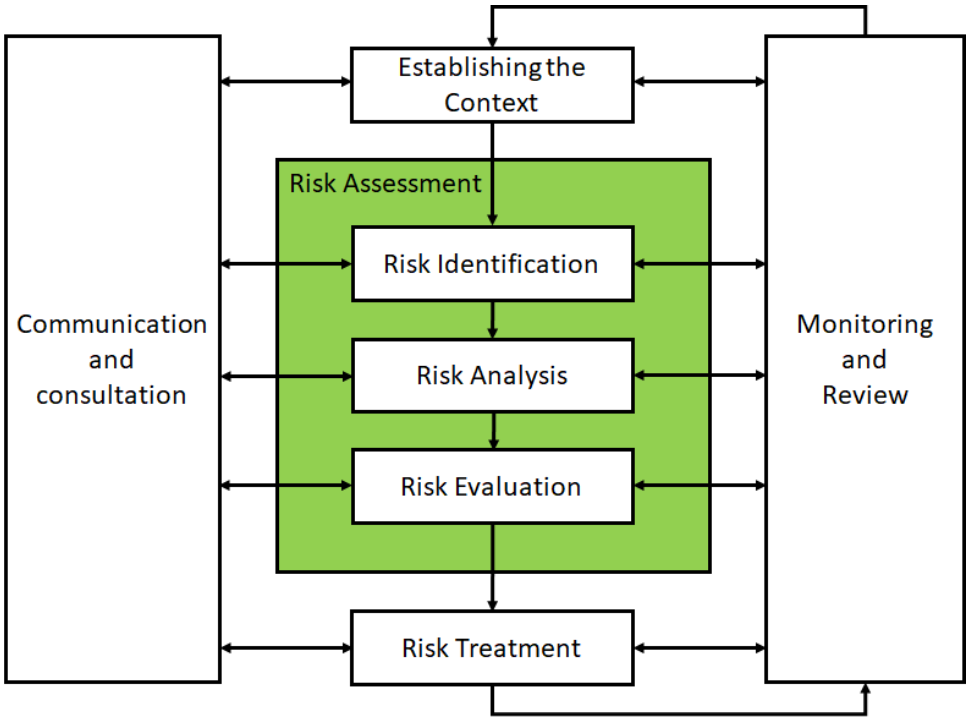


Figure 2.1: Risk Management Process [1]

Stakeholders are the ones that make judgements about risks based on their perceptions and consequently must be identified, recorded and taken into account in the decision-making process.

This step is important so that everyone understands the risks, decisions regarding them and the direction that risk management is supposed to take. Thus, it has to happen regularly and, as can be seen in Figure 2.1, it should happen on all the stages of the risk management process.

Establishing the context of a company is another risk management step and its purpose is to define the company's objectives, identifying external and internal parameters that influence how risk should be managed, as well as setting the scope and risk criteria [1]. Risk criteria is the reference against which the importance of the risk is evaluated, e.g. it is defined that a risk is critical if it involves the loss of more than a specific amount of money and when the risk's possible loss is defined, it is compared to this risk criteria (in this case, the limit value defined) and established if it is critical or not to be treated [2]. For any organization, it is important for the context to be well defined in order for decisions to be made accordingly, reason why stakeholders must be consulted in this stage.

Risk assessment is composed of three steps that correspond to risk identification, analysis and evaluation, as can be seen in Figure 2.1. As the names suggest, it is necessary to first identify the risk, secondly to analyse it and lastly to evaluate said risk.

Since the main focus of this dissertation is regarding risk assessment techniques, the detail on risk assessment itself will be in section 2.2 taking into account the ISO 31010 [5].

Risk treatment corresponds to the selection of one or multiple paths or measures that can be taken to modify risks, and then implement them. These paths are called controls. On this step, treatments are created, decided if the respective residual risk levels are tolerable and generate new risk treatments if necessary.[1]

Monitoring and Review is another important step in order to maintain risk management always up to date, checking it periodically or in an *ad hoc* manner. The points to look out for, according to the ISO 31000 are [1]:

Identify emerging risks;

Ensure that controls are effective and efficient in both design and operation;

Obtain further information to improve risk assessment;

Analyse and learn lessons from events (including near-misses), trends changes, successes and failures;

Detect changes in the external and internal context, including changes to risk criteria and the risk itself which can require revision of risk treatments and priorities.

Everything has to be recorded in order to have a historical path that can be evaluated and used to, for example, not repeat the same mistakes as before or to amplify decisions that had a good response at the time.

2.2 Risk Assessment

As mentioned before, Risk Assessment is part of the Risk Management process (see it in Figure 2.1 highlighted with green) and is being detailed in this chapter, starting with an overview and then with the depiction of the techniques chosen to be explored in this dissertation.

2.2.1 Overview

Risk assessment improved the way to understand if risks can compromise the achievement of objectives of an organization and if the controls in place are adequate and effective. This information is the basis for decision-makers and responsible parties to decide about the most appropriate approach to be used to deal with the risks. Therefore, the output of risk assessment is considered an input for the decision-making processes of the organization [5].

In order to assess risk, it is necessary to go through three stages, shown in Figure 2.1:

1. Risk identification
2. Risk Analysis
3. Risk Evaluation

Risk identification is the step one and where risks should be identified, their causes and respective consequences. A comprehensive identification of risks is paramount to generate a list of risks that are going to be analysed and evaluated. The ones that are not identified will never be analysed and evaluated and it can become a problem for the organization that is not handling them [1].

After identifying the risks, one must then analyse them, hence the second step: **risk analysis**. This analysis of the identified risks demands a better understanding of the risks. Should the risk be treated? What treatment strategies and methods would be the most appropriate in order to deal with the risk? What type of risk is it? What level of risk is it? These questions are the ones being answered in this stage.

In order to answer these questions, one must consider the events that correspond to risks, the likelihood of the risks and the possible consequences. The latter, depending on if they are positive or negative consequences, must be dealt with by defining controls that may prevent, drop down the likelihood of the risk or that minimize the effect of the risks' consequences. These controls have to be maintained and reviewed since their effectiveness and efficiency might change and, in that case, a change of approach might be in order [1].

Risk analysis output is to be used as an input to the next step of the risk assessment: **risk evaluation**. According to [1], risks are evaluated taking into account which need treatment and the priority for treatment implementation. It involves comparing the level of risk found during the analysis process with the risk criteria established when the context was defined, which will establish if there is the need for the risk to be treated.

2.2.2 Techniques to explore

Risk assessment process can be performed or assisted using tools and techniques. Depending upon the needs of the organization, more than one method of assessment may be used. These tools and techniques are being depicted on Figure 2.2 for each step in the risk assessment process [5].

There are multiple tools and techniques and therefore only a selected few will be detailed further: brainstorming, check-lists, cause-effect analysis and consequence/probability matrix. The techniques to be explored are related to the report types that the stakeholders wanted to be implemented into the HoliRisk platform. Those report types are introduced and detailed in section 5.3 and use directly or adapt to the techniques explored below.

Brainstorming is a technique used loosely as being any type of group discussions. However, according to [5], the true meaning lies in particular techniques to try to ensure that people's imagination is triggered by the thoughts and statements of others in a group. The input for this technique corresponds to a group of knowledgeable people of the organization, system, process or application being assessed. The process of brainstorming can be a formal or an informal one. In one hand, the informal is less structured and more *ad-hoc*. On the other hand, the formal process is more structured and the participants have to be prepared in advance and the session has a defined purpose and outcome with a means of evaluating ideas. Thus, it may be necessary a facilitator that would guide the group through their thoughts into the outcome necessary.

Check-lists correspond to a technique used to check that everything has been covered. It can also be used as part of other risk assessment techniques. These check-lists can be lists of hazards, risks or control failures that have been developed from experience by the result of a previous risk assessment.

Tools and techniques	Risk assessment process				
	Risk Identification	Risk analysis			Risk evaluation
		Consequence	Probability	Level of risk	
Brainstorming	SA[1)]	NA[2)]	NA	NA	NA
Structured or semi-structured interviews	SA	NA	NA	NA	NA
Delphi	SA	NA	NA	NA	NA
Check-lists	SA	NA	NA	NA	NA
Primary hazard analysis	SA	NA	NA	NA	NA
Hazard and operability studies (HAZOP)	SA	SA	A[3)]	A	A
Hazard Analysis and Critical Control Points (HACCP)	SA	SA	NA	NA	SA
Environmental risk assessment	SA	SA	SA	SA	SA
Structure « What if? » (SWIFT)	SA	SA	SA	SA	SA
Scenario analysis	SA	SA	A	A	A
Business impact analysis	A	SA	A	A	A
Root cause analysis	NA	SA	SA	SA	SA
Failure mode effect analysis	SA	SA	SA	SA	SA
Fault tree analysis	A	NA	SA	A	A
Event tree analysis	A	SA	A	A	NA
Cause and consequence analysis	A	SA	SA	A	A
Cause-and-effect analysis	SA	SA	NA	NA	NA
Layer protection analysis (LOPA)	A	SA	A	A	NA
Decision tree	NA	SA	SA	A	A
Human reliability analysis	SA	SA	SA	SA	A
Bow tie analysis	NA	A	SA	SA	A
Reliability centred maintenance	SA	SA	SA	SA	SA
Sneak circuit analysis	A	NA	NA	NA	NA
Markov analysis	A	SA	NA	NA	NA
Monte Carlo simulation	NA	NA	NA	NA	SA
Bayesian statistics and Bayes Nets	NA	SA	NA	NA	SA
FN curves	A	SA	SA	A	SA
Risk indices	A	SA	SA	A	SA
Consequence/probability matrix	SA	SA	SA	SA	A
Cost/benefit analysis	A	SA	A	A	A
Multi-criteria decision analysis (MCDA)	A	SA	A	SA	A

[1)] Strongly applicable.

[2)] Not applicable.

[3)] Applicable.

Figure 2.2: Tools and techniques used for risk assessment [5]

Each element of the list must be handled by a person or a team, going through the list and reviewing whether items on the check-list are present or have been dealt with [5].

Figure 2.3 exposes an example of a check-list[1] that handles the possible risks that a project manager has to worry about when working with a software customer. This check-list serves to analyse if the customer is ready to be involved in the development process and to understand if there is too much risk to work with the said customer.

(C) Customer Related Risks

Following generic risks are associated with different customers

Sr.	Check Point / Defect Statement	Check Mark (√) the Appropriate Column	
		Yes	No or N/A
1)	Have you worked with the customer in the past?		
2)	Does the customer have a solid idea of what is required? Has the customer spent the time to write it down?		
3)	Will the customer agree to spend time in formal requirements gathering meetings to identify project scope?		
4)	Is the customer willing to establish rapid communication links with the developer?		
5)	Is the customer willing to participate in reviews?		
6)	Is the customer technically sophisticated in the product area?		
7)	Is the customer willing to let your people do their job–that is, will the customer resist looking over your shoulder during technically detailed work?		
8)	Does the customer understand the software engineering process?		

Note: If the answer to any of these questions is "No," further investigation should be done to assess the risk.

Figure 2.3: Check List Example

Cause-and-effect analysis is a method to identify possible causes of an event that is not desirable. An effect can have a number of causes which may be grouped into different categories. An example of a fishbone is shown in Figure 2.4[2] where the line in the middle is the starting point to understand it. *Shorted Motor Coil Causing 23% Failure Rate on Cycle Destruct Test* is the end result or the effect that one is trying to understand, and the arrows pointing to it are the lines corresponding to the causes. Therefore, the main line has six possible causes/categories that affect the line in the middle: the man-power, the methods, the machines the metrics, the materials and the minutes. And these causes can also have causes of their own, and so on.

Consequence/probability matrix is a technique that can be applied in all steps of the risk assessment process. According to [5], this technique is used to rank risks, sources of risk or risk treatments on the basis of the level of risk. It is used to define which risk need further analysis, which risks need treatment first, or which need to be referred to a higher level of management.

A consequence and probability scales must be done in order to define the three dimensions of the matrix, the abscissas, the ordinates and the severity of the risk (see Figure 2.5 [3]). In order to apply this

[1] `http://m.softwaretestinggenius.com/?page=details&url=risk-assessment-and-analysis-checklist` - accessed 05-September-2017

[2] `http://fishbonediagram.org/example-1-poor-product-quality/` - accessed 05-September-2017

[3] `https://safeworkpro.com/risk-assessment/what-is-a-risk-assessment-matrix` - accessed 05-September-2017

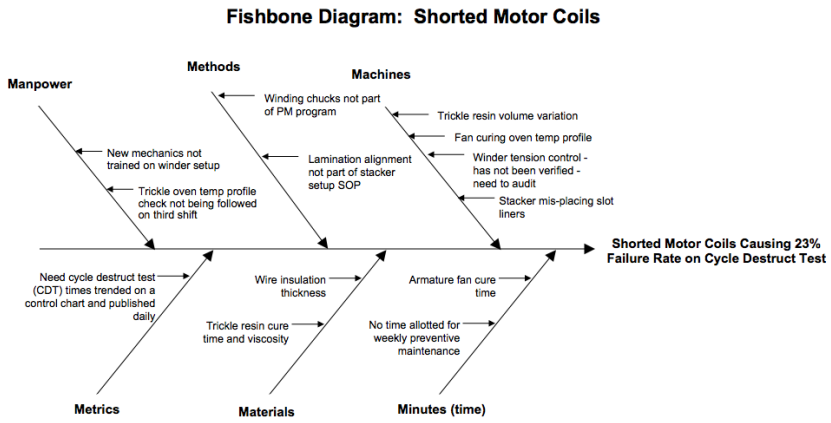


Figure 2.4: Fishbone Diagram Example

technique and use it in a tool, it needs people, ideally a team, with expertise and as many data as is available to help judge the respective consequences and probabilities.

		Consequences				
		Insignificant (1) No injuries / minimal financial loss	**Minor (2)** First aid treatment / medium financial loss	**Moderate (3)** Medical treatment / high financial loss	**Major (4)** Hospitable / large financial loss	**Catastrophic (5)** Death / massive financial loss
Likelihood	**Almost Certain (5)** Often occurs / once a week	Moderate (5)	High (10)	High (15)	Catastrophic (20)	Catastrophic (25)
	Likely (4) Could easily happen / once a month	Moderate (4)	Moderate (8)	High (12)	Catastrophic (16)	Catastrophic (20)
	Possible (3) Could happen or known it to happen / once a year	Low (3)	Moderate (6)	Moderate (9)	High (12)	High (15)
	Unlikely (2) Hasn't happened yet but could / once every 10 years	Low (2)	Moderate (4)	Moderate (6)	Moderate (8)	High (10)
	Rare (1) Conceivable but only on extreme circumstances / once in 100 years	Low (1)	Low (2)	Low (3)	Moderate (4)	Moderate (5)

Figure 2.5: Consequence/probability matrix

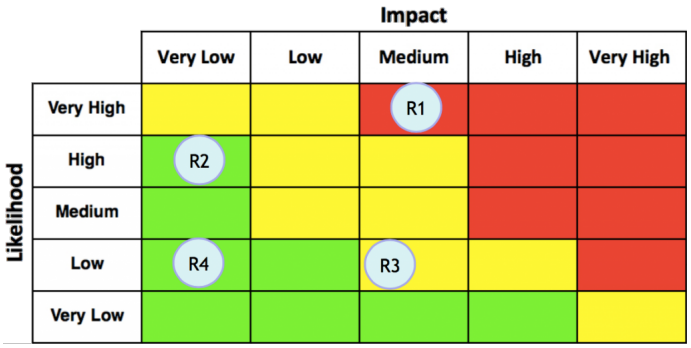


Figure 2.6: Consequence/probability matrix results

The output of this technique is a ranked list of risks with significance levels defined, see Figure 2.6 [4]. An association can be made between level of risk and treatment, e.g. low level risks can be directly associated with the decision of not treating the risk.

[4] `https://www.cgerisk.com/solutions/risk-register-and-hazard-identification` - accessed 05-September-2017

2.3 Risk Management Tools and Techniques

Multiple tools exist nowadays to allow companies to manage their risks by being able to register and controlling them. Some tools are oriented towards the registering of risks, others in allowing the user to establish tasks to be resolved, there are many different ways to build a risk management application. There are even tools which full purpose is not to manage risks but can be used to that end, such as spreadsheets, and therefore may not be the best ones to do it. Tools oriented towards risk management may have different focuses such as to define a single way for the company to manage risk, limiting the decisions of the company. It is also possible to develop tools that centre its focus only on the enterprise as a whole and not on its departments or activities, making it harder to manage the risks since it is not well designed or because one can not have the level of detail required.

Out-of-the-box tools have a way of handling risk management, i.e. the company developing the tool defined a process that follows the regulatory requirements and forces it into its customers as being the best way. Furthermore, if the company wants to use the tools in a different or needs the tool to be changed, it is necessary for the developing company to use its developers to change and adapt to the needs of the customer. This flow is not an optimal one since each company may have multiple different ways to want to implement the risk management process and that is why other tools allow the user or enterprise to create and establish the manner in which the risk management process should be built and conducted.

The tools to be presented are:

Resolver ERM[5];

EQMS[6];

Logicgate ERM[7].

Resolver is a software that offers a Governance, Risk Management and Compliance (GRC) enterprise solution. The risk management solution is labelled Resolver Enterprise Risk Management (ERM) and is an out-of-the-box solution that is made available to any enterprise.

Each enterprise needs to ask for a demo to see the out-of-the-box solution where the risk management steps are made available: risk identification, risk analysis, risk assessment, risk treatment, risk monitoring, reports, test (see Figure 2.7 [8]).

For each step, there is a possible solution and the existent configuration is the one made available on the demo. If the client wants to change any, it would be necessary for the Resolver's developed to change it as per the client's requirements.

Therefore, even tough the Resolver ERM does offer multiple steps regarding the risk management process, it is not a flexible software.

[5] http://www.resolver.com/apps/enterprise-risk-management-software/ - accessed 05-September-2017
[6] http://www.eqms.co.uk/grc/ - accessed 05-September-2017
[7] https://www.logicgate.com/solutions/enterprise-risk-management/ - accessed 05-September-2017
[8] http://www.businesswire.com/news/home/20170315006142/en/Resolver-Launches-Newest-Cloud-Based-Enterprise-Risk-Management/ - accessed 05-September-2017

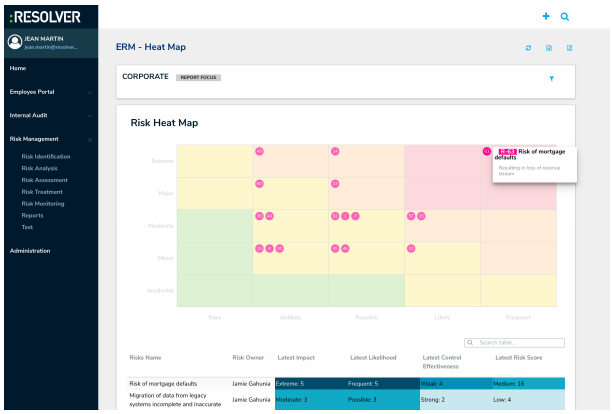


Figure 2.7: Resolver ERM Interface

EQMS is another out-of-the-box tool that is centred on risk, i.e. it is possible to create risks and all the possible relations with them. The configuration is made by the company developing the software and therefore, it is not possible to be changed by the user of the tool. In Figure 2.8 [9], it is possible to see that the tool is completely focused on the creation of risks and where the possible connections are, for example, likelihood, impact and controls. It is possible to configure those connections in a risk but is not possible to configure the risk itself, e.g. it is not possible to define that risk is also related to an event, instead of having only a description.

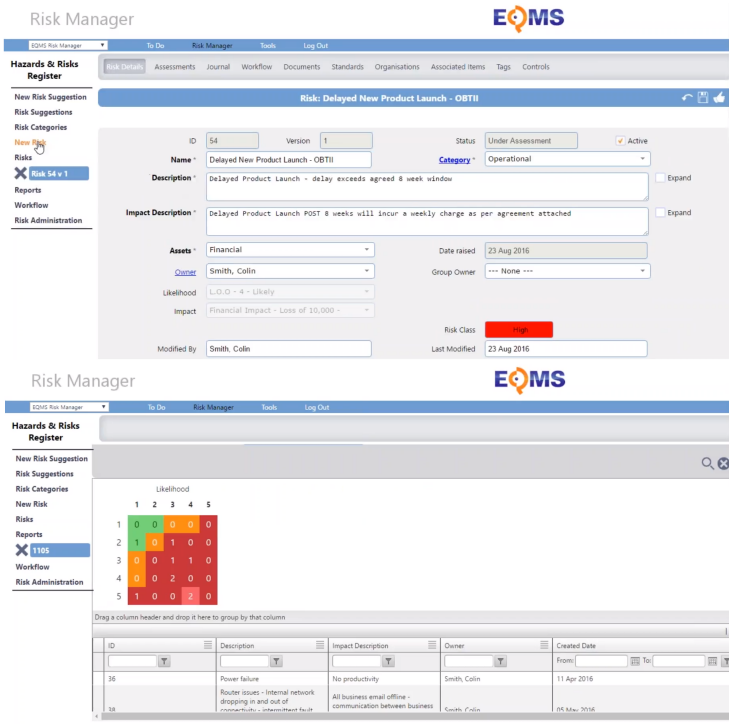


Figure 2.8: EQMS Interface

Once inside the EQMS application there is a check-list is shown in the beginning as a TO-DO list where there is a list that the user must complete or verify and another type of report corresponding to

[9] `http://quality.eqms.co.uk/blog/author/michael-ord/` - accessed 10-September-2017

a pie chart which is not a good chart to convey information[10]. It also available a risk matrix where it is possible to go through the risks and check the respective severity in the matrix.

It is a risk centred software which does not mean it will not do what is supposed to but means that it is not flexible enough for every company's objectives.

Logicgate is another GRC solution where the risk management is also centred on risk such as the EQMS and also an out-of-the-box software. From Figure 2.9 [11] it is possible to configure a workflow of to handle risks, e.g. it is possible to configure when an expert is supposed to review the risk and update it if necessary. This software allows the registration of risks and makes it possible for the user to automate a workflow of tasks to handle risks. Since it is risk-oriented, it is not possible to define other concepts. The only concepts to be used are the ones provided and no other can be created unless a change is requested, meaning extra cost.

There are a few report, for example assignments where it is possible to expose the collaborator the is assigned to specific activity or workflow. Another report exposes risks depending on their status, e.g. one can have a report to show all the risks that have not yet been handled.

There are multiple tools that were not mentioned but in order not to repeat the same advantages or disadvantages, only these selected three were introduced.

In sum, the existent risk management tools are oriented towards a specific mindset of risk management but it is not flexible enough to adapt to every organization request. Risk management can be implemented using these tools but only if the organization chooses to adapt to it. Reports are available on these tools and since the data is always designed in the same manner, it becomes easier to develop, e.g. a risk can be configured to have a likelihood and a consequence and nothing else can be changed. Restricting the tool so that the company has to follow its flow becomes easy to control and to develop but not as easy for the company to adapt to it. This lack of flexibility affects also the reports that are available since, if the manner in which the data is defined is restricted, so will the reports.

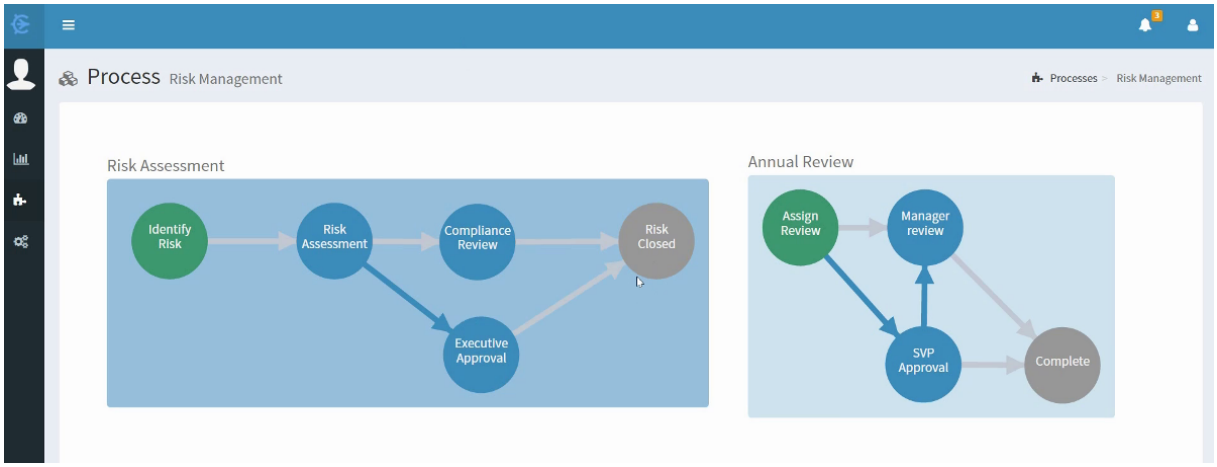


Figure 2.9: Logicgate Interface

[10] http://www.businessinsider.com/pie-charts-are-the-worst-2013-6 - accessed 08-September-2017
[11] https://www.getapp.com/it-management-software/a/logicgate/ - accessed 08-September-2017

2.4 Summary

This chapter was meant to introduce all relevant information to understand the basis on which the HoliRisk application was developed on. It was introduced the standards for risk management, mainly its process and the techniques used in the risk assessment step.

Tools that could be used for the purpose of managing the risks of an enterprise were also introduced, depicted and reviewed.

Chapter 3

Problem Analysis and Requirements

This chapter introduces the HoliRisk platform and respective problem and afterwards, the requirements the problem of this dissertation are presented.

3.1 Problem context and motivation

Risk management is a crucial process to achieve success in an organization. However, the implementation of this process in organizations is mainly supported by spreadsheets whether in departments, projects or even activities making it difficult to integrate. Thus, the vision of the risks becomes fragmented and spread through various contexts. The same happens with risk management oriented applications which are not flexible and restrict the user to manage risk in the manner available on the platform.

ISO 31000 emerged in 2009 and with it arose the opportunity to create a flexible application that could be used to implement risk management in any organization, independently of its focus or context. Thus, INESC-ID[1] took advantage of that opportunity and developed a platform named HoliRisk. João Edmundo was the one to take the first step in HoliRisk development, which was continued by the work of Carlos Martins [6]. This dissertation's purpose is to continue and improve the work done before and coordinate it with the work of others, specifically the work of Eduardo Melo [7].

HoliRisk is a generic platform to support phases of the risk assessment process in different organizational contexts. It allows a holistic view of risk through a centralized and configurable risk register, rather than a fragmented view where risks are dispersed across multiple contexts. Given that the objective was to manage risk, independently of the context of the organization and in a holistic way, the creators chose HoliRisk as the application's name.

After developing the core of HoliRisk, it became obvious that there was a gap in the needs of an organization. That gap was related to acquiring knowledge or business intelligence from the data inserted and available on the application. Therefore, a Reporter module to be introduced in order to be possible to create multiple reports that can lead to multiple conclusions and faster answers regarding each business.

[1] https://www.inesc-id.pt

3.2 HoliRisk

As mentioned before, HoliRisk is a platform which purpose is to manage possible risks in an organization. In order to manage these risks, the ones responsible for the company's risk management must insert all the relevant data, structured according to the business it regards to, into the application.

The main goal of HoliRisk is to be a platform where it is possible to configure the manner in which risks are managed. Firstly, only authenticated users should be able to use the platform. Secondly, it should be possible to define multiple contexts or domains, e.g. the company as a whole or specific department, and to configure the data that would populate those domains.

3.2.1 Use Cases

In the beginning of the development process of the HoliRisk platform, there were decisions made regarding its use cases. Those use cases can be seen in Figure 3.1. The implementation of these use cases would then be the base over which this dissertation would be built upon.

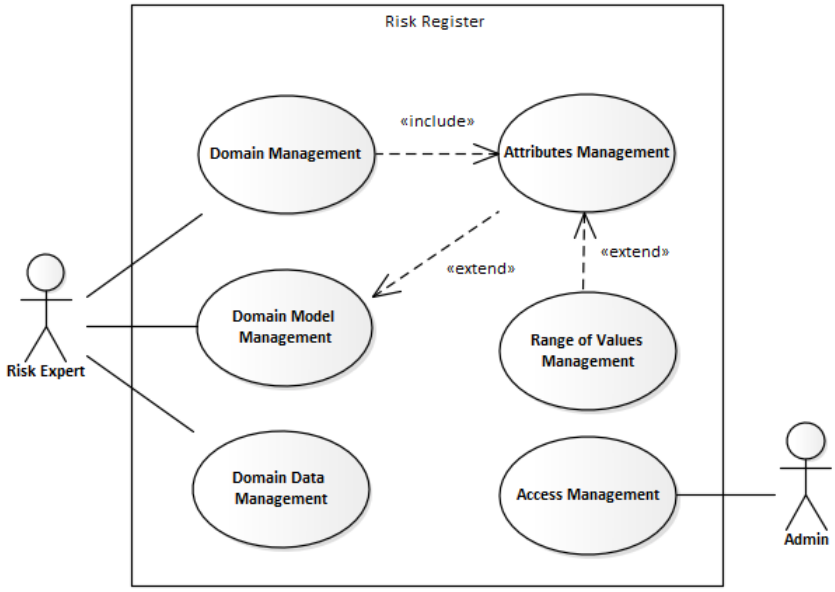


Figure 3.1: HoliRisk Use Cases

The **Access Management** component was introduced because of the ISO 31004 [8], that states that the information associated with a risk is sensitive and that is why it is important to guarantee confidentiality, security and privacy of the collected information stored on the application. In order to do that, it was implemented an access management to control who accesses HoliRisk. The application access is divided into two different steps: **user registration** and **user authentication**. The first step is when the user does not yet exist on the platform and has to register into it, by setting a pair username/password and providing a correct email. Once registered, at the authentication phase, the user must enter the correct pair username/password to get access to the respective previously created domains or to public domains already created on the application.

The **Domain Management** component had to manage domains. A domain corresponds to the data stored on the platform regarding a specific context. Authenticated users can create new domains and edit or remove them. These domains can be set to public, shared or private. The purpose of a domain is to register risks and in order to do that, it is necessary to define a domain model that is composed of concepts, relations, attributes and ranges of values. After the domain model is defined, it becomes possible to register the data according to it.

The **Attribute Management** component handles the management of the attributes on a domain. For each domain, it is necessary to define which attributes can be used in the domain model. These attributes are characterized by a name, type and whether it is mandatory. For example, further in the Interface section, in Figure 3.5, the domain model has a concept named Risk, which can have attributes such as Name or severity, as can be seen on the right of the figure.

The **Range of Values Management** component serves to manage the definition of ranges of values. Once defined, they can be associated with an attribute to have a value of the respective range. A range of values can be quantitative, qualitative or a table:

Quantitative values: numerical values that can serve for measurements on quantitative scales. The range of values from 1 to 5, or 1, 4, 5, 6, 8, 10 are two ranges of quantitative values.

Qualitative values: used to define categories. The range of values *high, medium, low* and *masculine, feminine* are two ranges of qualitative values.

Table: sets of sets. The previous sets allow the definition of lists of values. However there was a need to create more complex sets that resemble sets of sets. Examples such as [1, high, 2, medium, 3, down], or [event x, occurs 10 h, event y, occurs 11 h, event z, occurs 12 h] are a representation of tables.

Function management was developed to allow attributes to be defined depending on the values of other attributes. This makes it possible for an attribute of name 'severity' for example to correspond to the multiplication of the values of the attribute 'likelihood' and the value of the attribute 'impact' [7].

The **Domain Model Management** component is needed to manage the construction of a domain model that can be done through a form or in a visual manner, using a class diagram such as Unified Modelling Language (UML) (see Figure 3.5 further in the Interface section). To define the domain model of the risk register, it is necessary to define the concepts, the properties that define the concepts and the relationships between the concepts.

Concepts - A concept defines an entity in the domain model which has associated attributes and can have multiple relationships.

Attributes - During the construction of the domain model, it is necessary to associate attributes with the concepts, in order to define the properties of this concept.

Relationships - Concepts can have relationships between them, defined by the relationship's cardinality.

The **Domain Data Management** component is where the data can be registered according to the domain model that has been previously defined. This data is stored and accessible for further review in the Reporter. Since most of the existing risk data is in spreadsheets, the platform needs to be able to import and export information in that format. Once the domain model is defined, the user can export a worksheet and work on it, in order to be able to import the domain data back into the platform.

3.2.2 Architecture

The use cases were interpreted and implemented using the technology that is going to be detailed in the next section. The implementation was done having in mind the HoliRisk's architecture that can be seen in Figure 3.2. The respective components are access management and risk register.

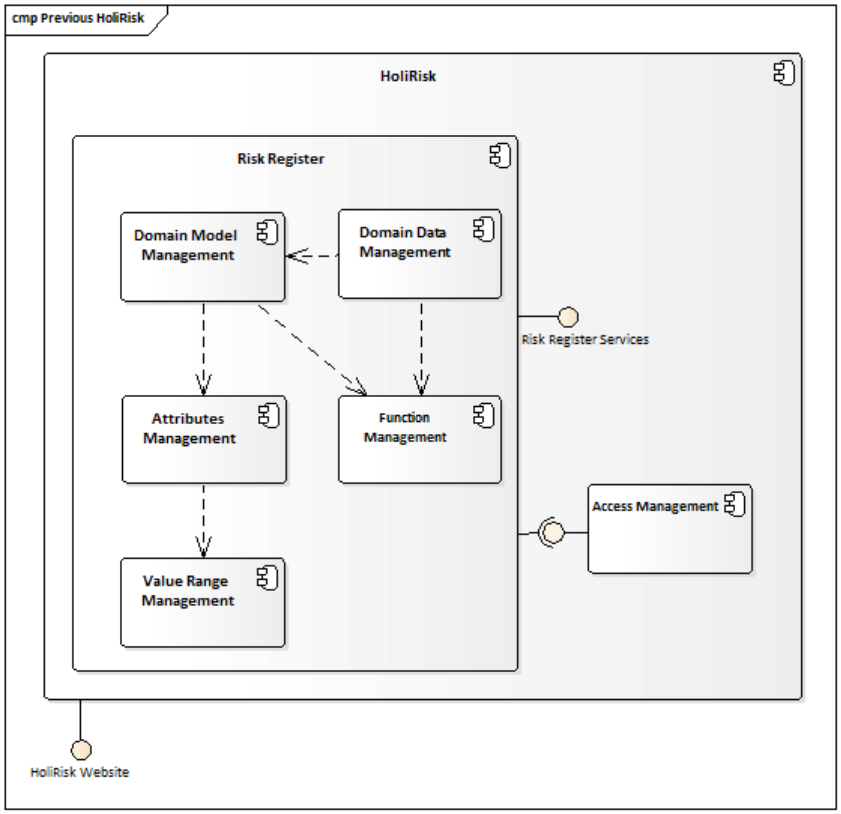


Figure 3.2: HoliRisk Previous Architecture

The access management serves to manage users' access to the HoliRisk platform making it available a service so that, any component such as Risk Register, can confirm that the user has the right credentials.

The risk register component is composed of five components: the Domain Model Management handles the UML definition for each domain; the Domain Attributes Management and the Value Range Management manage the types of attributes and the possible range of values available on the domain, respectively; the Domain Data Management handles the population of the existent Domain Model; and the Function Management handles the possibility to associate a function to an attribute.

The Risk Register provides an interface to be used by other components, such as the Reporter

module implemented in this dissertation that will use it to obtain the necessary data registered.

The HoliRisk, being a web-based platform, provides a web interface for users to access it online via a web browser.

3.2.3 Technology

In order for the platform's development to be flexible and agile it was decided that the development stack to be used would be MEAN[2] which stands for MongoDB[3], ExpressJS[4], AngularJS[5], NodeJS[6].

MEAN is used to build web applications composed of two parties, client and server, that communicate with each other. When people use a web browser, such as Mozilla Firefox or Google Chrome, they are interacting with the client side. The client is responsible for handling the user's input and, when necessary, it requests the server for what is required. For example, a user inserts the profile information and hits the save button. The client, in response to that push of the button, requests the server to save the data, that the user inserted, on the database (located on the server).

The main objective of using this MEAN stack is to build web applications using only JavaScript language[7] and guarantee that the integration between all the technology models used is made seamlessly.

Next will be briefly explained the MEAN four components.

MongoDB is an open source database that is used to persist HoliRisk's data. It is a document-oriented database which means that its structure is defined by JavaScript Object Notation (JSON) documents. This means that, since all stack of development is using solely Javascript and the database's structure is defined and queried using the same notation, the data can flow through the stack in a fast and scalable manner.

NodeJS is the server side of the application. Once more, this environment was developed so that the server side would be in Javascript. NodeJS provides an event-based architecture with an asynchronous API in order to consume few resources and be as efficient as possible. This allows, for example, to create highly scalable server applications, such as real-time web applications [8].

ExpressJS is a web application framework for NodeJS that provides a set of robust features to interact with web applications or mobile devices. This framework provides NodeJS with a communication interface that respects the Hypertext Transfer Protocol (HTTP).

AngularJS is the technology chosen to develop the HoliRisk platform has a human-based web interface. This technology is very extensible and versatile, streamlining Hypertext Markup Language (HTML) pages through controls and extensions to the HTML language, which would otherwise have a more complex implementation and a much higher development cost. In addition, AngularJS is designed to develop applications based on the Model View Controller (MVC) standard, in order to facilitate both development and testing.

[2] http://mean.io/
[3] https://www.mongodb.com
[4] https://www.expressjs.com
[5] https://angularjs.org
[6] https://nodejs.org
[7] https://www.javascript.com/
[8] https://nodejs.org/en/about

3.2.4 Interface

HoliRisk is a web-based platform that can be accessed by multiple users that need only to register themselves. Afterwards, it is possible to log in and take advantage of the risk management platform. Once logged in, it is possible to create and manage multiple domains which can correspond to the company as a whole, to departments such as the human resources department or to activities.

Each domain can be shared with other users by defining it or with everyone by setting it as a public domain. Users can access and edit domains created by themselves, shared with them or public domains.

After defining the name of the domain, it is possible to select it and configure the data that is going to represent said domain and the reports that are going to be used on it. In this case, let us select the PUR domain (see Figure 3.3).

.

Figure 3.3: Domain List

When entering a domain, there are two main possibilities: the configuration/population of the domain and the data analysis of the domain data. Figure 3.4 exposes both, configuration is made on the left and the data analysis, specifically the reporting on the right.

.

Figure 3.4: Domain Main Page

On the domain, it is possible to configure a domain model that represents the domain using UML. The domain data model is defined by configuring the concepts/classes and the cardinality relation between them. This can be seen in Figure 3.5.

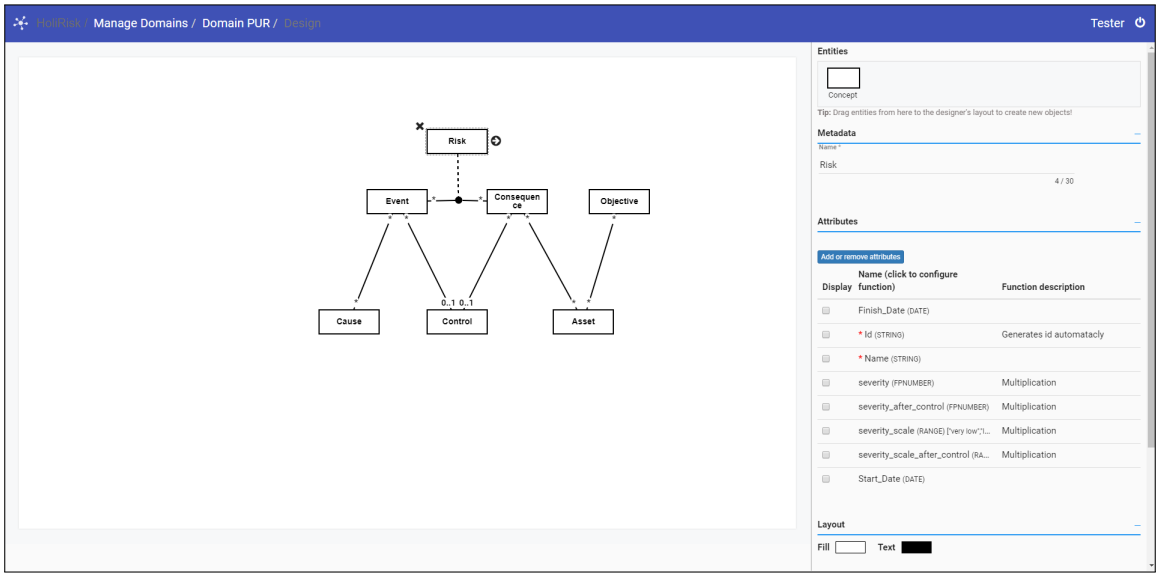


Figure 3.5: HoliRisk Domain Data Model

Each class is represented also by attributes which have to be configured beforehand to be used in the domain model configuration as attribute types (see Figure 3.6).

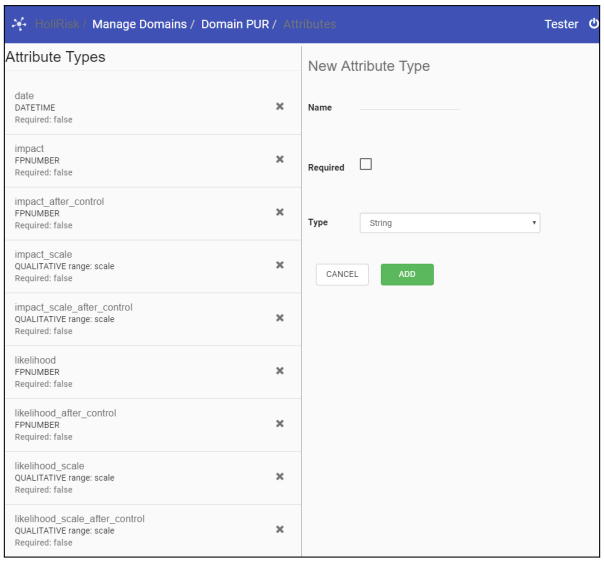


Figure 3.6: HoliRisk Domain Attributes Types

Each attribute can be defined by primitives such as Strings or Floats but can also be represented by ranges (see Figure 3.7). An example of a range is that the value can take a value from 0 to 1 or from Low to High.

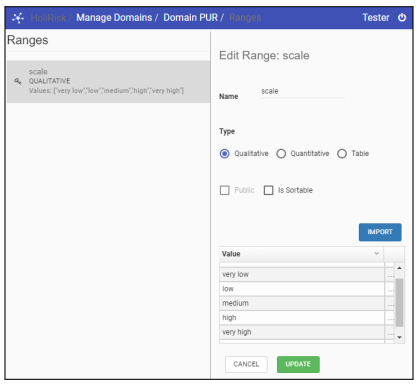


Figure 3.7: HoliRisk Domain Ranges

Once the domain data model is complete, it is possible to populate it (see Figure 3.8). Each class can be instantiated, for example, the class event can be created multiple times depending on the number of possible events are identified that can occur regarding the risk management of the company.

The population of the domain data model is the so-called risk register where it is possible to register risks and the concepts related to it such as events and consequences.

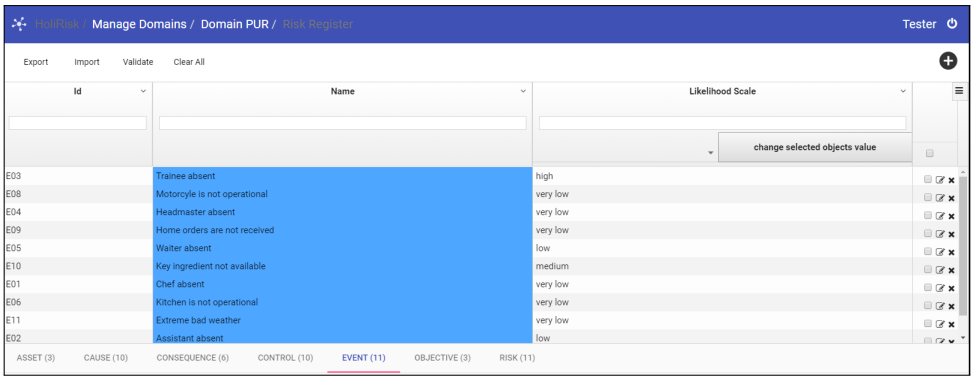


Figure 3.8: HoliRisk Domain Populate

3.3 Requirements

The main problem of the HoliRisk application was the lack of a reporting module, which would allow the users to acquire knowledge, taking into account the data inserted on the platform. Therefore, it was needed a new module that would allow the user to create reports of different types and configure them for the desired purpose. It should also be available the options of update and delete of said reports.

These requirements were defined during meetings with the stakeholders of the platform, and after defining that a reporter was needed and the respective functionalities, it was introduced the idea of two main modes, configuration and exploration modes. On the configuration mode it should be possible only to configure reports and on the exploration mode, it should be only possible to explore them. The introduction of these two main modes regards to the existence of two possible actors that interact with the Reporter on the platform: the Risk Owner and the Risk Expert. The Risk Expert is the on that handles the configuration of the reports and tests the end result to verify that they are correct. On the other hand, there is the Risk Owner that needs only to see and explore the resulting reports that were configured by the Risk Expert, in order to take advantage of them to make decisions regarding the business.

The use cases represented in Figure 3.9 present the main functionalities that should be available and respective actors for the new HoliRisk Reporter.

Figure 3.9: Reporter Use Case Model

This Reporter module should use multiple risk assessment techniques in order to improve the enterprise's risk management. For that to happen it is necessary to create multiple types of reports, that the user can create and manage throughout the risk management process. The report types to be implemented are:

Overview - As the report's name reveals, this report must show the user the overall statistics regarding the number of classes, objects and objects per class inside the respective domain.

Risk Matrix - This report uses the risk assessment technique Risk Matrix where the risks are allocated to a specific position in the matrix, depending upon a function. This function computes the axes and what the coordinated result is.

Filtered List - The filtered list report aims to check the possible relations between the existent data. It should check the existing objects and relations regarding the attributes chosen to be seen in the report's list.

Fishbone - This report uses the Cause and Effect risk assessment technique which intends to show dependencies of attributes. I.e. if an attribute value is created with a function, there are other attributes it can depend on, and those dependencies are the ones to be exposed in this type of report.

Combined - Combined reports aim to show a conjunction of reports already defined. For example, one report can be set to show the overview report and a risk matrix combined into one.

The Risk Matrix, Filtered List and Fishbone were set as requirements to implement three risk assessment techniques: consequence/probability matrix, check-lists and cause-and-effect diagram, respectively. The Overview is for the user to have a report already created that shows the overall information of the domain in question. The Combined report was set as requirement in order for the user to be able to compare reports already created, for example joining a Overview and Risk Matrix report.

A good report interface should be as flexible as possible and auto-explanatory, by being user-friendly and easy to use. Therefore, it is important for everything to be consistent and make every functionality visible to the users [9].

As said before, a report has two different modes, the configuration mode and the exploration mode, where each one of them offers different functionalities. In the context of an organization, there can be two types of actors that will interact with the Reporter: the risk owner and the risk expert.

The risk expert has to be a knowledgeable collaborator that is aware of all the information in the application and the way it is structured in order to be able to configure any type of report. On the other hand, the risk owner is the one responsible for the risk and the one to analyse the resulting reports, previously configured.

When playing a risk expert role, the user can create new reports and verify that the result is the one expected by viewing them. Besides, the risk expert can also edit or delete previously created reports. In case of another user have changed or created a report, it is possible to refresh the existent reports, guaranteeing that it is all up to date.

When in a risk owner role, the user can only view/explore the existent reports or refresh them.

3.4 Summary

HoliRisk is a platform built to manage the possible risks of a company. In order to get intelligent information from those risks, it is necessary to create a report phase with multiple techniques in order to better understand what can happen to the company. Furthermore, it is important to understand that there are two actors, the risk expert and the risk owner, in order to separate the ones that configure a report, and the ones that extrapolate knowledge from said report.

Chapter 4

Proposed Solution

This chapter aims to expose the proposed solution to solve the problem at hand. The solution's architecture and technology used for its development will be detailed.

4.1 Development Method

This dissertation was developed following the **prototyping** method which means to build a robust prototype in a structured and refined manner [10]. The prototyping methodology focuses on building an early approximation of a final product which will be tested and reworked until a complete prototype is finally achieved. This method is an iterative, trial and error process that occurs between the developers and the users which will reduce time and costs since changes that are detected later in the development process can have high costs.

Besides the cost and time advantages, this process prevents misunderstandings between the client and the developer making the final product more likely to satisfy the look, feel and performance desired by the client. Therefore, this method consists of three mains steps, as can be seen in Figure 4.1:

Analyse and Design;

Development;

Evaluate.

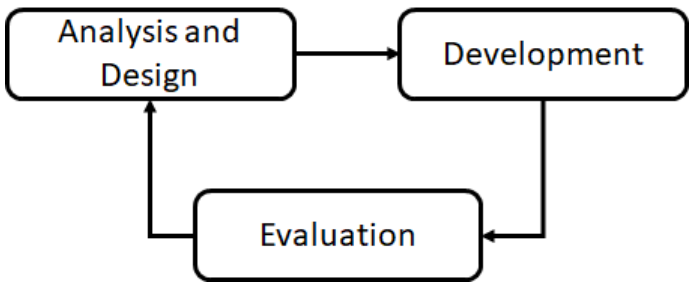


Figure 4.1: Prototyping Development

The phase to **analyse and design** is when the state of the prototype is analysed and new objectives or updates on the existent objectives are defined. Besides that, the behaviour is analysed and decided whether or not it is as expected or if it should be changed and improved.

The **development** phase serves for the developers to implement newly defined objectives or update any implementation due to a decision made while designing, where it was decided that the implementation should be done in a different manner.

In the **evaluation** phase is made the evaluation of the last prototype. In this phase is verified that the objectives established in the first phase are the ones that were implemented in the prototype. This verification is made by the execution of tests in the interface, using it in different browsers to guarantee compatibility, making performance tests and allowing the users to test the platform as well.

The first (analyse and design) and last (evaluation) phases happened with the members of INESC-ID and a client interested in using the application, the stakeholders of this platform. These meetings happened once every two weeks in order to guarantee that there was enough time for development but also fast enough for any misleadings on the development to be corrected and oriented towards the right path if needed.

As referred previously, following this approach it is possible to obtain rapid feedback from the users, even though the application is not entirely complete, since there were multiple iterations. Each iteration produces a new prototype that corresponds to the refinement of the previous one, with the additions, corrections or updates of functionalities.

4.2 Architecture

This dissertation was based on the HoliRisk previous version, presented in section 3.2, therefore its architecture was augmented with the addition of the Reporter component that is presented in Figure 4.2. Since part of the architecture was introduced in the section 3.2.2, this section will only address the components regarding the Reporter itself.

The Reporter component interacts with the previous system by using the services provided by the Risk Register to obtain the necessary data of the domain and by using the access management to verify the user's permission. Furthermore, the implementation of this Reporter component was had by reference, the use cases defined as requirements in the section 3.3 where it shows two actors, the Risk Owner and the Risk Expert, and six use cases, View Report, Refresh Reports' Data, Create Report, Edit Report, Delete Report and Delete All Reports. The implementation was complete and the Reporter is now composed of three main components:

Report Management;

Report Data Management;

Data Analytics Engine.

Figure 4.2: HoliRisk Architecture

All three of the main Reporter components are going to be detailed further but it is important to mention that the Report Management and the Report Data Management components have a more complex inner architecture that needs to be addressed. Since the architecture is the same for both components, only one will be depicted.

Report Data Management is the one to be depicted, in Figure 4.3, exposing a layered architecture. The top layer corresponds to the Presentation Layer which corresponds to the interface and what is viewed by the user on the web browser. The Application Layer corresponds to the logic behaviour of the application, where the commands from the layer above are carried on. The Access Database component handles the bridge between the front and the back end of the platform, i.e. the connection between the application and the database. Lastly, the Database Layer handles the Create, Read, Update, Delete (CRUD) operations while storing the data on the database, specifically the reports for the Reporter component.

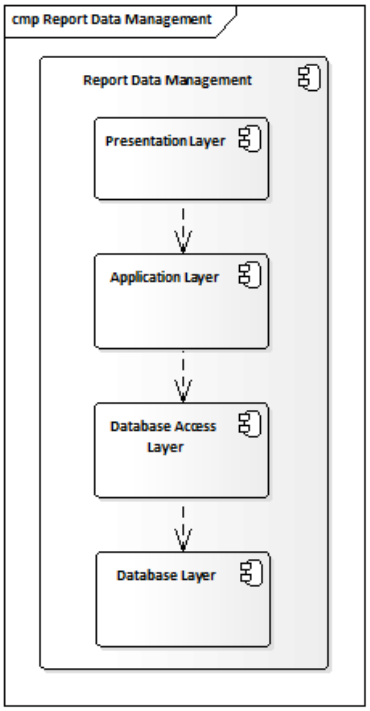


Figure 4.3: Report Data Management Component

4.2.1 Report Management

The report management component aims to manage the life cycle of a report, i.e. the creation of any type of report, its edition, its exploration and its disposal.

Once in the Reporter, there is a list of reports available that the user can go through. Every single report can be viewed or explored, hence the View Report use case that allows the user to select a report and work on it as a Risk Owner (see Figure 4.4 to verify the Risk Owner's possible use cases)

and make important decisions regarding the business itself or as a Risk Expert (who can use every available functionality on the Reporter) and configure the reports in such a way that the Risk Owners have the right information and knowledge to make the right decisions.

As said above, the Risk Expert is able to Edit Report, which is one of the use cases available for the Reporter. However, the Risk Owner is not meant to do the configuration and that is why the respective actor can only do two things, View Report and Refresh Reports' Data. The latter serves to refresh or update the data regarding the reports, i.e. if a Risk Expert is making changes to a report and the Risk Owner wants to have the most updated version of it, it is possible to refresh the reports' data and make sure that the data it is being used is up to date.

The Risk Owner can also delete a single report one by one or delete them all without needing to go through them all. There is only one report that must and always is present which is the Overview Report. This report will be introduced next.

Furthermore, the Risk Expert in order to do all of the presented use cases, Edit Report, Delete Report, Delete All Reports, View Report and Refresh Reports' Data, it is obviously necessary for a use case to be Create Report. This use case is sort of four types of use cases since there are four types of reports that can be created by the Risk Expert.

There are five types of reports: Overview, Filtered List, Risk Matrix, Fishbone and Combined. The Overview report is created by HoliRisk and there is only one of its type since it presented the overall count of the domain data, i.e. the name of the domain, the number of classes of a domain, the number of objects of a domain and the number of objects per class.

The Filtered List report is a type of report meant for the user to select multiple attributes from the same or different classes and the Reporter joins the data and presented all the possible connections between said attributes.

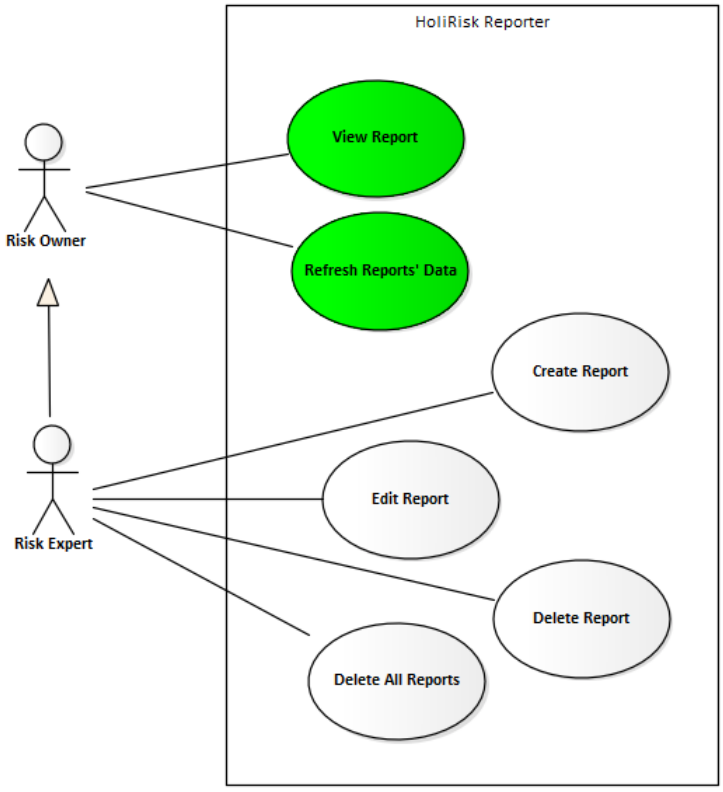


Figure 4.4: Risk Owner Use Cases

The Risk Matrix report is a type of report that uses the risk assessment technique Consequence/Probability Matrix. This technique was already detailed but it is necessary to define an attribute that is represented by a function with two attributes, e.g. an attribute severity that is calculated by the multiplication of the attributes consequence and likelihood. Some steps are needed to configure this type of report in order to define the axis of the risk matrix and dimensions needed for the ending result to be a risk matrix and a grid. On the risk matrix, the risks must be presented at the point of the matrix correspondent to the abscissas and ordinates of the risk, e.g. using the same example as before, if the consequence is Low and the likelihood is Medium, then the risks with the severities calculated by that combo of consequence and likelihood should appear at the respective point of the matrix. On the grid, it is shown the list of risks that appear on the matrix with the respective values of consequence and likelihood. Between the matrix and the grid is an interconnection so that if a point of the matrix is selected, all the correspondent risks on the grid are highlighted. And the inverse also happens, i.e. if a risk on the grid is selected, the point where the risk is in the matrix is also highlighted.

The Fishbone report is a type of report that serves to know the dependencies of a specific attribute. For example, if the Risk Owner wants to know what dependencies a risk severity has without having to go to the domain data, it is possible to use the Fishbone report to understand what dependencies an attribute has. In the case presented before in the Risk Matrix report, the severity was calculated by the consequence and the likelihood and therefore, in this type of report, severity depends on consequence and likelihood.

The Combined report aims to provide a manner of joining multiple reports already created. If the Risk Owner wants to see the Overview report and a Risk Matrix report, it is possible using the combined report by selecting the ones to be presented and the end result is the combination of the selected reports.

Before continuing, it is necessary to sum up the actors and use cases. The actor Risk Owner is able to Refresh Reports' Data and to View Report. The actor Risk Expert is able to everything that the Risk Owner is allowed to do and to Create Report, Edit Report, Delete Report and Delete All Reports. There are five types of reports (Overview, Filtered List, Risk Matrix, Fishbone and Combined) but only four can be created by the Risk Expert since the Overview report is always available. The difference between the actors is implemented in the Reporter as modes, the configuration and the exploration modes where the Risk Owner corresponds to the configuration mode and the Risk Expert corresponds to the Exploration mode.

4.2.2 Report Data Management

The report data management component serves the purpose of managing the reports' data and making it available for presentation and for extraction. Extraction of the resulting reports and the respective valuable information is done in different manners depending upon the type of report and the information available on them:

Overview Report - It is possible to extract the data on this type of report in an image format

Filtered List and Matrix Risk Reports - Tables or grids on these reports make it possible for the extraction of this data in Excel and PDF extensions. On the Filtered List, the grid available for extraction corresponds to the data available taking into account the user's chosen attributes. On the Risk Matrix, the grid available is the one regarding the attribute that is represented by a function with two attributes that was chosen by the user.

Fishbone Report - It is possible to extract an image regarding the fishbone in question that is representing an attribute and its dependencies.

4.2.3 Data Analytics Engine

As previously mentioned before, the Risk Owner and Risk Expert are implemented in such a way that they correspond to the exploration and the configuration modes, respectively. On the configuration mode, it is possible to configure the all report in order to be with the configuration that the Risk Expert desires. Once the configuration is done, it is possible to switch to the exploration mode. This switch entails that a process is run using the configured data to build the report it is supposed to expose on the exploration mode. The process that happens when entering the exploration mode corresponds to the call upon a service with the name of Data Analytics Engine that receives the configuration data, uses it to create the data necessary for the resulting report and returns it for the Report Manager to use in the exploration mode.

The service called upon is named Data Analytics Engine can be explained in two parts. Firstly, Data Analytics corresponds to the process of analysing sets of data to draw conclusions about the information they contain. Secondly, it is an Engine since it is a service that supports all of the existent types of reports. When it is requested to change to exploration mode, the Data Analytics Engine is called and receives the data to be processed. The process corresponds to different types of algorithms depending on the report type it is being processed.

The Overview report data is used to count the number of classes, objects and objects per class. The Filtered List report data is used to expose the objects of different classes joined together in one grid in order to take advantage of it to understand for example if all the events have possible controls associated with them. The Fishbone report data is used to check the dependencies of an attribute, for example, the severity of a function can be calculated by a consequence and a likelihood, establishing that the severity is dependent upon those two attributes. The Risk Matrix report is the report that needs more data to be used in order to expose a risk matrix and the respective risks in a grid.

Once the input data is processed, the resulting data can be used to feed the report that is being exposed to the user. The Risk Expert uses the exploration mode to guarantee that the configuration is correct and that the result is what the Risk Owner is expecting. The Risk Owner uses the information available while exploring the report to guide business decisions.

According to [11] , "for analytics-driven insights to be consumed — that is, to trigger new actions across the organization — they must be closely linked to business strategy, easy for end-users to understand and embedded into organizational processes so that action can be taken at the right time".

4.3 Technology

The main technology to be used was the same as the HoliRisk platform detailed in section 3.2.3, i.e. it is used the MEAN stack.

Even though the main technology is used, there was a need to have more javascript dependencies on the front-end to create the necessary reports. These dependencies were added depending on the report types and they were:

pdfmake and html2canvas libraries were added in order to extract an HTML div as a PDF file;

ui-grid library was used to expose data in a grid with multiple functionalities such the exportation of the data on the grid to PDF and Excel files;

highcharts library was used to expose a risk matrix;

d3 library was used to expose a fishbone diagram.

The library *ui-grid* is used on both Filtered List and Risk Matrix reports. On the Filtered List report, it is shown a list of objects with the attributes that the user chose on the configuration of that report. On the Risk Matrix report, it is shown a list of objects regarding the attribute chosen on the configuration as the attribute that is represented by a function with two attributes upon which it is dependent.

Since the other reports had a possible way to export its data, mainly the Filtered List and the Risk Matrix reports, then it was implemented a way to export the Overview report. It was used the *html2canvas* in order to transform the HTML into an image, specifically the division in which the report is at. Once in a canvas, it is used the *pdfmake* to export the image into a PDF file.

The *Highcharts* library has a set of charts available to be used and it was used a Heat-Map which is a representation of data in the form of a map in which data values are represented as colours. In this specific case, it was needed to adapt the Heat-Map to a Risk Matrix since the Heat-Map changes colours depending on the number of data in a specific quadrant. Risk Matrix has colours already configured by the Risk Owner, e.g. if the matrix were to be represented by an attribute that could have the values Low, Medium and High, the Risk Owner had to associate colours to them as so: Low as green, Medium as yellow and High as red, for example. The adaptation also included that the name of the objects of the attribute chosen on the report configuration would appear in the space regarding the values of the abscissa and ordinate.

The *D3* library was added to the implementation in order to expose a Fishbone. The DAE is the one that created the necessary data and feeds the D3 with the attribute and its dependencies if there are any, in which case only the attribute will appear.

In sum, the MEAN stack was used to develop but there were also some libraries needed to implement the reports themselves.

4.4 Summary

The proposed solution is to add the Reporter component in the HoliRisk architecture to be able to create and view multiple reports. The technology used is the same as the one of the HoliRisk but with some dependencies to develop the required needs of each report.

Chapter 5

Solution Implementation

The Reporter component added to HoliRisk was developed on top of the existent platform and coordinated with an ongoing dissertation [7] which objective was to correct bugs and add a few functionalities such as associating functions to attributes, making it more challenging.

For the demonstration part of this dissertation, a scenario was specifically created. This scenario corresponds to a pizzeria that needs to manage its risk, hence the name PUR. The PUR domain was defined inside the HoliRisk platform, i.e. its data model, attributes, ranges and logic data. Besides, this scenario was also used on the dissertation developed in parallel with this one [7].

Regarding the behaviour of the platform, once a user logs into the HoliRisk platform, it will present a list of available domains to which the user has access to. Firstly, it is necessary for the user to select a domain and after that, it will be possible to edit the domain's definition and also use the Reporter module to better analyse the domain's data.

Once the domain is selected, two sections are presented to the user: the Domain and the Reporter. The Domain section is where the domain is defined and populated, while the Reporter section is where the user can visualize the data in multiple defined reports. The latter refers to the one introduced by this dissertation.

This chapter describes in detail the implementation of the proposed solution and is divided into three parts: report modes, report interface and report types.

5.1 Reporter Modes

The Reporter section was developed so that two possibilities are offered to the user: the configuration mode and the exploration mode. Both take the user to the same interface but the difference is the mode in which the report is on.

If a user wants to configure the reports he should enter through the configuration mode. On the other hand, a user that only wants to explore the data has to enter the exploration mode. Configuration and exploration modes will be further detailed once the reports are introduced.

5.2 Reporter Interface

The Reporter main page is divided into two main sections: the list of reports on the left side of the interface - used to list all the existent reports - and the selected report on the right side (see Figure 5.1) - used to show the selected report.

Figure 5.1: Reporter Interface

5.2.1 Report List Area

As mentioned before, the Report List Area aims to list all the existent reports. The reports are separated by type of report in order for the user to quickly find the wanted one. Another way to find a report is by using the search functionality available, that corresponds to a filter where the user needs only to write the name of the report and the reports without that name will not be displayed in the report list. Furthermore, this left side area also provides a few functionalities to handle those reports:

Create Report - It is possible to create four types of reports that are detailed further in the section Report Types: Risk Matrix, Filtered List, Fishbone and Combined. The Overview report is created automatically[1].

Refresh Reports - Since it is possible for multiple users to use the web-based platform, a refresh button was created to guarantee that the reports' list is always up to date;

Delete Report - It is possible to delete each report individually except the Overview report;

Delete All Reports - It is possible to delete all report in a single action, not being necessary to go through all of them. The Overview report is also deleted but recreated with the default configuration.

Filter Reports - There is an area where the user can insert the title of a report that is desired and the reports will filter taking the input into account. This is useful since there is no limitation on how many reports can be created and the filter functionality makes it easier and faster to find a report.

Report Type Division - All of the reports are divided according to its type. This, alongside the filter, is a performance enhancer for the user since it allows a to differentiate all the reports by type and making it easier to focus on the reports that matter.

In order to create a report, one must first select the type of report to create (see Figure 5.2), only then may the user configure it. Once created, the report can be saved and stored in the database or can be discarded by the user. This also happens to any change made to the report, if the changes are not saved they are lost.

[1] There is only one report of type Overview and it is always present since it exposes the overall numbers of the domain and needs no configuration other than the one of the title.

.

Figure 5.2: Create Report - Report Type Choice

The reports listed are all selectable and when one of them is selected, the right side of the Reporter interface will provide options so that the user may configure and explore it.

5.2.2 Selected Report Area

The objective of the right side of the Reporter interface is to configure and explore the selected report. Each report can be seen in two possible modes already introduced: the configuration mode and the exploration mode. The Overview report will be used as an example to demonstrate how the selected report area works.

When entering the Reporter, the Overview report is the one selected by default. If the intent of the user is to configure the report, it will be in the configuration mode. On this mode is where the selected report can be edited, in the case of the Overview report, it is only possible to change its title. The other reports have more steps to be configured which will be detailed in the next section.

Once the configuration is complete, the report can be changed to the exploration mode, which is highlighted in Figure 5.3. This switch is visible on the top right corner of the selected report area and once changed, the report will be executed according to the configuration defined previously in the configuration mode. Therefore, the execution is made every time the mode is switched (to the exploration mode) and in real time, in order to guarantee that the data used is up to date. Once that process is complete, the user can analyse all the available data.

As was explained before, when the user is exploring, the functionalities are not all available because in this mode the user is only supposed to explore and not to edit or create anything.

.

Figure 5.3: Report Modes

The interface allows the user to change between configuration and exploration mode as pleased. In this specific and only case of the Overview report, it is possible to extract all the report into Portable Document Format (PDF), using oriented libraries (*html2canvas* and *pdfmake*). On the other reports, it is only possible to extract the data itself using other libraries available, such as *ui-grid* where it is possible to extract to both Excel and PDF files.

The configuration and exploration are specific to each type of report and for that reason, it is detailed separately in the next section.

5.3 Report Types

There are five types of report: Overview, Filtered List, Fishbone, Risk Matrix and Combined. Each is different from one another in terms of configuration and in terms of what is exposed in the exploration mode.

5.3.1 Overview

The overview report is the first one to be created in an automatic way since it reports the overall numbers that define the domain selected before and therefore, it needs no input from the user. However, the title may be changed as a configuration of the report.

The exploration mode of the overview reports exposes the domain name, the number of existent concepts/classes, the number of existent objects/instances of the classes and the number of objects per class (see Figure 5.4). Only on the exploration mode of the overview report is possible to export its data in a PDF format.

The Overview report is used to assess, control and verify the numbers of existent data of the risk management process.

Figure 5.4: Overview Report Results

5.3.2 Filtered List

Filtered list is a report which objective is to obtain all the possible relations between all the data and expose it to the user in a table.

The configuration necessary for this report is to define the class attributes to be shown in the exploration mode (see Figure 5.5). Even though the configuration is simple, the process of the filtered list to obtain the exploration mode is not, since all the possible relations have been accounted for in the execution.

The end result of this process can be seen in the Figure 5.7. This was possible after developing an algorithm, similar to a depth search on a tree, that will now be described.

The objective of this report, using and adapting to the risk assessment technique check-list, is to allow the risk owner to check the resulting list and identify something that should not happen or that is prejudice to the company. For example, guaranteeing that every possible event has a control set upon it or that for every consequence there is an impact well defined.

Figure 5.5: Filtered List Report Configuration

Firstly, it is necessary to know that what is shown in the table is the data instances of each class, the objects. Another thing that is necessary to be taken into account is that the attributes chosen can be from one or multiple classes. Therefore, the algorithm runs through the objects that are instances of the classes of the chosen attributes in order to obtain the data of said attributes' values.

Secondly, each object can have relations to other objects depending on how the data model was defined. If an object has relations, which also correspond to objects, it is necessary to go through all of them, unless the class of the object being handled wasn't chosen previously in the configuration mode.

As in a DFS algorithm, one starts at the root (in this case would be each object) and explores as far as possible (through all the existent object's relations) along each branch before backtracking [12].

Thus, a recursive method is applied to each object to go through all the possible paths. Figure 5.6[2] represents well this algorithm logic. The numbers shown on each vertex are correspondent to the order in which they were visited.

.

Figure 5.6: DFS example

Since this algorithm was implemented recursively, which means that the same code is run for every object, a stopping point is needed to start the backtracking. Therefore, there are three possible stop points:

There are no more relations;

The object being handled was already handled before;

The object being handled has the same class as one of the previously handled elements of the same branch.

.

Figure 5.7: Filtered List Report Exploration

After a list of data is ready in a table format, it is necessary to remove the duplicates, which are created due to the recursive algorithm, and besides that, it is also necessary to calculate the most complete path. The latter is achieved by merging all the existent lines, through a comparison of relations. For example, if A has a relationship with B and also has a relationship with C, and two different lines are showing A-B and A-C, this will be merged as A-B-C if, and only if, A B and C are objects of different classes.

Once algorithm is complete, the data is presented to the user (see Figure 5.7). The library used to show the tables is the *ui-grid* which already has default features such as moving columns pinning columns to the left or right.

Another example to show the Filtered List interest is that it can work like a check-list of sorts, i.e. one can check if all the events have controls associated with it. In Figure 5.8 can be seen the configuration and exploration of the join between the Event and the Control concepts. Most of the shown events have a control associated with it but there a few that do not, such as when the kitchen is not operational or when the dining room is not operational. The Risk Owner, with this report, could check this type of check-lists and guarantee that the best risk management is in place for its business.

5.3.3 Fishbone

The fishbone report uses the same basis as the Cause-and-effect analysis technique, i.e. it uses a fishbone diagram to find the cause for an effect. In this case, the report asks the user to choose an attribute of a class/concept previously defined in the domain (see Figure 5.9 where the user chose the severity attribute of the class Risk), and exposes all the other attributes that the chosen attribute

[2] `http://www.techiedelight.com/dfs-interview-questions/` - accessed 10-September-2017

.
(a)
Event
Con-
trolled
Con-
fig-
u-
ra-
tion
.
(b)
Event
Con-
trolled
Ex-
plo-
ration

Figure 5.8: Filtered List - Event Controlled

is dependent on. For example, if said chosen attribute corresponds to a function that has multiple arguments, those arguments (which are also attributes) are shown as a dependency of the chosen attribute. The arguments are the cause of the chosen attribute, where the attribute itself represents the effect.

.

Figure 5.9: Fishbone Report Configuration

Once again, the configuration requires only that the user sets the attribute to be the effect and the exploration mode will present a fishbone diagram with the attribute in the centre of the diagram, as the central spine, and all the respective dependencies exposed as the secondary spines representing the causes of the attribute. The example of the severity attribute of the class Risk can be seen in Figure 5.10. This example allows the user to know, without going through the data searching for the attribute, that the severity of a Risk is calculated and dependent upon the likelihood of an Event and the impact of a Consequence.

Another example for the Fishbone report type can be the severity of the Risk but after a control. A control in terms of Risk Management can be used to try and control a Risk which means that the severity of the Risk is going to be dependent upon said control. In the case of Figure 5.11, the dependencies show that the controls are applied directly on the likelihood of the Event and on the impact of the Consequence, creating a second level on the fishbone. The control affects the likelihood and impact which affect the severity of the Risk, which brings the conclusion that the control also affects the severity, even if not directly.

.

Figure 5.10: Fishbone Report Exploration - Risk Severity

On the data model of the PUR scenario, the Event has a relation with Cause and the Consequence has a relation with Asset. Now let us imagine that the company decides to change the data model and updates the Event.likelihood to depend upon the temperature of the Cause and updates also the Consequence.impact to depend upon the weight of an Asset. The resulting Fishbone of the same

severity after control of the Risk which is updated as well with two more bones on the Fishbone diagram (see Figure 5.12).

This type of report, Fishbone, can be very useful to verify the dependencies of any attribute and make sure that the changes that were done on the domain data model were indeed successful. Besides that, a Risk Management analyst can analyse this data and possibly the data on Filtered Lists and realize that it is necessary to change strategies.

.

Figure 5.11: Fishbone Report Exploration - Risk Severity After Control

.

Figure 5.12: Fishbone Report Exploration - Risk Severity After Control Second Scenario

5.3.4 Risk Matrix

The Risk Matrix report uses the Consequence/prob matrix technique in order to rank risks depending on its level.

The configuration of the risk matrix report is done in 4 steps:

1. Select the attribute that is represented by a function of two arguments
2. Select the arguments to be used as abscissa and ordinate of the risk matrix
3. Select the colors that will correspond to the values calculated from executing the function already introduced
4. Choose the order of the range values of the two axes by importance level

On step 1 it is necessary to choose a function attribute, i.e. an attribute that is defined by a function and is dependent upon the other two attributes. The configuration mode shows all the attributes that are represented by a function of two arguments X and Y.

Step 2 is to define the risk matrix Cartesian coordinates, i.e. which argument corresponds to the abscissa and which argument corresponds to the coordinate. It is necessary to, every time this is set or updated, build the data from the configuration defined in step 1 and 2 (see Figure 5.13).

Step 3 is to define the colours related to the ranges of values that correspond to the result of the attribute function defined in step 1 (see Figure 5.14). After defining the colours is also necessary to build the data to prepare the next step by pressing the *Build* button available the end of this step.

Step 4 is to select the risk matrix axes order regarding the X and Y set in step 2 (see Figure 5.15). Once selected the order, it is necessary to build the data before switching to the exploration mode.

The standard is for the result to correspond to the risk concept, and the axes to correspond to the event.likelihood and consequence.impact.

.

Figure 5.13: Risk Matrix Configuration Step 1 and 2

.

Figure 5.14: Risk Matrix Configuration Step 3

The exploration mode will expose the risk matrix with the colours chosen in the respective positions, according to the function results. Also, each risk (R1 to R10) are positioned according to their level of likelihood and impact (coordinates). Adding to that, there is a grid with all the risks, represented on the matrix, exposing in detail the values they assume for each the coordinates attributes.

This report also reacts to the user's selections. If a user selects a specific an area of the risk matrix, the correspondent lines will be highlighted in the table below.

In a real company, there will be more risks other than the 11 exposed regarding the PUR scenario was of 11 and therefore, it was needed a little upscale in order to test scalability. Thus, it was inserted multiple risks and the ending result can be seen in Figure 5.16. In order to insert the data, it was used a spreadsheet to augment the risks registered and imported into the platform.

.

Figure 5.15: Risk Matrix Configuration Step 4

.

Figure 5.16: Risk Matrix Exploration

.

Figure 5.17: Risk Matrix Exploration

5.3.5 Combined

The Combined report is a way to view multiple reports in a single report and allow discussions and Brainstormings. These reports can be created by different users that want to compare the reports and understand the differences, similarities or co-relations.

.

Figure 5.18: Combined Report Configuration

.

Figure 5.19: Combined Report Exploration

The configuration of this type of report corresponds to the selection of the reports that the user wants to see in the final report/visualization. The reports suited for selection are all the ones that are not of type Combined, for example, it is possible to have an overview report and a risk matrix in the same report or even all the existent reports except combined ones (see the example on Figure 5.18).

The exploration mode of the report is the conjunction of all the reports previously selected in the configuration mode (see Figure 5.19). The need for this type of report brought a few challenges since it had to include all the reports chosen and all the respective functionalities as well. These challenges and others are detailed in the next sections.

5.4 Technological Challenges and Decisions

In the beginning of the dissertation and while adapting to the languages being used, there was the need to correct the Risk Register component and to add and contributed with new functionalities. Once the Reporter component implementation started, there were a few other challenges and decisions to be made to maintain best practices while developing.

5.4.1 Risk Register Challenges

Risk Register is the core of the platform that allows the definition of a data model to represent the manner in which the risk is going to be managed. That data model is then populated with actual data. This data is available to be analysed by anyone that needs to see it but there was no data analytics component to help the users to take knowledge from HoliRisk in a better way than just observing the data. The Reporter was the solution but before its implementation, the Risk Register had to be tested in order to guarantee that all functionalities were working. The dissertation of Eduardo Melo, with whom the work had to be coordinated with, had the job of correcting said bugs in order for the Reporter to analyse the data [7]. In order to get used to the HoliRisk development, it was also made a few corrections to help Eduardo and consequently this dissertation.

The Risk Register has a functionality that allows the import of data into the platform which allowed to remove the already existent data. This functionality was corrected and since it was a useful one, a Remove All feature was added so that the user can clear all the data without having to do it only when importing some data.

Other errors were also discovered and corrected mainly regarding uncaught exceptions that had to be handled with. For each one, a different message was defined in order for the user be able to understand what went wrong and correct it in the platform.

5.4.2 Reporter Interface Decision

After the corrections, it was time to decide how the Reporter interface would look like. There were some paper mockups done but in the end, the main idea was to have all the existent reports on the left and the selected report on the right. The interface had to be functional but also modern and that is why the interface has a look and feel of the web version of Whatsapp[3] or Messenger[4] where all conversations are listed on the left and the selected conversation can be seen in detail on the right with all the inputs from the ones interacting.

5.4.3 Best Practices Decisions

The Reporter was developed following the MVC architectural pattern where the View corresponds to HTML, the Controller and Model to Javascript. So, for the Reporter, there is a single main View that is

[3] https://www.whatsapp.com/
[4] https://www.messenger.com

presented to the user, a single controller that manages every action coming from the View and a single Model representing a report on the database. It seems simple in the beginning because there are no variants but when it is necessary to manage multiple reports and multiple commands regarding each report, it becomes complex and best practices must be followed for development to be manageable.

The controller started out to be the main point where the all the logic would be at. The actions come from the view, the controller where all the business logic would be at. At one point, a single file is not the best practice to develop since it becomes too comprehensive. Therefore, the best practice decided was that the controller handles the routing of the view requests to services, i.e. most of the business logic was re-factored into services. These services correspond to Javascript as well, that work similar to an Application Programming Interface (API) where it is possible to use its methods. Other services can also work as classes and in this case, there was defined a Report Manager Singleton that is instantiated on the controller to manage the reports.

Taking into account that the controller is where every command is received, it is also where every error is caught and a corresponding message is given to the user in order for it to be clear what needs to be changed for it to go as planned.

5.4.4 Risk Matrix Report Challenge and Decision

It is hard to find a library that handles the implementation of a Consequence/Probability Matrix and therefore it was needed another option. Highcharts offers a library of charts and there is one that corresponds to a heat map where individual values are represented as colours, presented in a matrix and in this specific case (see Figure 5.20) where the colours are more intense depending on the number of the respective point.

Figure 5.20: Highcharts Heat Map

The Risk Matrix had to be adapted from this heat map to show the colours that are dynamically defined during the configuration of the correspondent report. The axes themselves are also defined by the user on the report's configuration and the matrix is displayed dynamically depending on that configuration.

Another challenge was the need to coordinate with another dissertation from which this one had dependencies regarding the development of Functions. The manner in which the Function Management was developed had to be decided in order to fulfil the requirements for the Risk Matrix report.

5.4.5 Combined Report Challenge and Decision

After implementing the Overview, Filtered List, Fishbone and Risk Matrix types of report, there was the need for a Combined Report to show all of the others together. The problem was on how the other reports were implemented in terms of the View: since there was only one report selected at each time, the way that the Reporter was implemented was by showing or hiding the division corresponding to the

type of view selected. The problem with that implementation with the addition of the Combined Report was that, if the user decides to have two reports with the same type, only one would appear. Therefore, it was necessary to switch from static HTML for the reports to dynamic one that could be added and removed as pleased.

Before detailing how the code was refactored, it is necessary to understand that the HTML is processed once the web page is loaded creating a Document Object Model (DOM) of the page, i.e. the HTML turns into a HTML DOM that is constructed as a tree of objects, making it possible for Javascript to create dynamic HTML. Since the directives are implemented using AngularJS, being the JS for Javascript prominent, it becomes possible to compile and insert new objects into the already loaded page.

Furthermore, it was created a directive for the report and 5 other directives for each type of report. The report directive is always present on the view, and inside of it can be added and compiled the report types. Once a report is selected, the respective report type directive is added to the main report directive making it available on the view. Even though the addition is important, so is the removal of the report type selected previously. If it is not removed it can lead to an error of memory leak once the selected report is changed multiple times and creates multiple objects without clearing the cache of the previous objects.

Since there was a need for this update and refactor because of the Combined Report, all of the reports were changed as well in order for the Reporter to be completely dynamic.

5.4.6 Reporter Development Structure

All the challenges and decisions have defined the structure of the client side, the front end side of the Reporter. The back end side of the Reporter follows the same structure as the rest of the HoliRisk platform and therefore will not be detailed further.

Figure 5.21: Reporter Development Structure

The final Reporter structure can be seen in Figure 5.21, where it is possible to distinguish, as said previously, the main view as HTML, the controller as Javascript, and the corresponding divisions where it is encapsulated the changes detailed before. It is divided into three folders:

Templates[5]: Reusable HTML components such as the report types or the dialogs to interact with the platform's user;

Directives[6]: Attributes or elements that augment an existing DOM element or represents a reusable DOM component, specifically the report, the report types, the combined report and the *highcharts* included into the risk matrix report.

[5] https://docs.angularjs.org/guide/templates
[6] https://docs.angularjs.org/guide/directive

Services[7]: Used to organize and share code across the application, such as the Data Analytics Engine.

5.5 Summary

A Reporter module was introduced into the HoliRisk platform and its interface has two main areas, the reports list area and the selected report area.

For this new module, there are five types of reports: overview, risk matrix, filtered list, fishbone and combined. Furthermore, the available functionalities correspond to the complete CRUD except for the overview report that cannot be deleted. There are two modes for the selected report: configuration and exploration. On the configuration mode, the report's details can be configured, while when on the exploration mode, one can explore and assess the result of the report.

The development code is available on the INESC-ID servers on the HoliRisk repository but credentials are required to access it.

Finally, it was summed up the main challenges and decisions through out the implementation of the solution that supports the state in which the Reporter is today.

[7] https://docs.angularjs.org/guide/services

Chapter 6

Evaluation

The involvement of real users is a crucial step in the development of a user-centred solution, a series of evaluations were conducted with a group of volunteer users in order to evaluate the Reporter solution. The main objective was to gather information about the users' expectations, opinions and difficulties while interacting with the platform and see if the main objectives were accomplished (regarding the usability and user experience).

It is important to note that the platform solution was evaluated by giving the volunteers a list of simple tasks to accomplish (while following their reactions and taking notes) and then responding to a survey regarding their experience and the platform itself.

To evaluate this work it was used a scenario called PUR, on Appendix A, created beforehand by José Borbinha and Ricardo Vieira, and chosen to be presented to the real users at the time of interacting with the platform, in order to have a domain data that would serve as a base for the evaluation of the Reporter component. Given that, we were able to focus only on validating the user experience according to the amount of time spent to conclude each task with success.

6.1 Process

Each user performed the same tasks while using the platform and the estimated time for the evaluation was between 10-20 minutes and was divided as follows:

Preparation - it was made an introduction to the domain side of the platform and to the data already available in order for the user to understand over which data the Reporter would be applying its techniques upon. Users were given a Guideline document and a User Manual (see Appendices C and B, respectively) to be read before starting the tasks (however, it was not mandatory to read these documents);

Setup - after receiving the necessary documents, the users did the required setup displayed in the Guideline document which included having a computer with internet access and entering the HoliRisk platform;

Tasks - with the setup done, the users completed a set of nine tasks, with different levels of difficulty, detailed in the Guideline document;

Survey - after finishing all the tasks, users were not only asked to fill a survey to measure the level of usability and utility of the Reporter, but also to measure the user experience and obtain their feedback regarding the module added to the platform (negative and positive aspects, new ideas for the Reporter and final comments). This survey can be found in Appendix D.

Regarding the documents available to the users, the User Manual explains all the user functionalities that the HoliRisk Reporter offers and shows the users how to perform them. This manual was available for consulting before and during the evaluation tasks. On the other hand, the Guideline document contains a brief introduction regarding the platform, specifically the Reporter, and the evaluation - the setup required and all the tasks to be performed.

6.1.1 Tasks

The proposed tasks consisted in exploring the platform and using all of the Reporter's available functionalities in order to cover the whole Reporter potential and receive the respective feedback in the end. The users followed a specific order to execute the tasks - from the easiest type of report to the more complex - with an the increasing level of difficulty.

The Overview report was the first one to be configured and explored because of its simplicity which allows users to get used to the main interface and explore the main functionalities. The users then created a Fishbone diagram, explored the Filtered List report type and finally created a Risk Matrix report which uses the risk matrix risk assessment technique.

As final tasks users were asked to delete all reports and answer a survey. All these tasks are presented in the Guideline document which can be found in Appendix C.

6.1.2 Participants and Setup

There was a total of 20 users that participated in the evaluation and all of them had at least a bachelor's degree and a couple of them had some knowledge about risk management. The majority of the evaluations were made online using the Skype application with the screen sharing option activated (i.e. see the user's screen), while other evaluations were made in person at a quiet place with no interruptions. These choices allowed a closer observation of the user's movements, note the eventual mistakes, register the time taken to complete each task and obtain better feedback.

Regarding the setup for the evaluation, the users needed a computer with internet access, the Chrome browser (recommended but another browser could be used) and the Skype application in case the evaluation was made remotely. As previously mentioned, each user received a briefing regarding the domain or risk register side of the platform and on the data already inserted.

Afterwards, the users opened the platform with the respective browser in order to start the tasks. As mentioned, the setup is described in more detail on the Guideline document.

6.1.3 Survey

After all the tasks were completed, each user answered the respective survey questions which were divided into three parts as follows:

Usability - “effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments” [13];

User Experience - “a consequence of brand image, presentation, functionality, system performance, interactive behaviour and assertive capabilities of the interactive system, the user’s internal and physical state resulting from prior experiences, attitudes, skills and personality, and the context of use” [14];

Personal Opinion.

The HoliRisk’s **Usability** was evaluated by following the five Es dimensions [15]:

Effective - How completely and accurately is the work/experience complete or the goals reached;

Efficient - How quickly the work can be completed;

Engaging - How well the interface draws the user into the interaction and how pleasant and satisfying it is to use;

Error Tolerant - How well the product prevents errors and can help the user recover from mistakes that do occur;

Easy to learn - How well the product supports both the initial orientation and continued learning throughout the complete lifetime of use.

These dimensions are evaluated by observing the users interact with HoliRisk: the **effectiveness** dimension were measured by observing if the users successfully completed the tasks; the **efficiency** dimension is determined by the average time the users took to complete the tasks; the **engaging** dimensions were evaluated using the answers to the survey that inquires about the level of satisfaction regarding the whole experience using the platform; the **error tolerant** dimension corresponded to the count of how many mistakes the users made during the tasks and if they found the way back to the guideline; the **easy to learn** dimension was measured by the questions on the survey regarding the level of difficulty of the tasks they users had to complete.

The **User Experience (UX)** was evaluated by following the six points classification that measures the UX according to the User Experience Questionnaire (UEQ) [16]:

Attractiveness - Overall impression of the product. Do users like or dislike it?

Perspicuity - It is easy to get familiar with the product?

Efficiency - Can users solve their tasks with the product without unnecessary effort?

Dependability - Does the user feel in control of the interaction?

Stimulation - Is it exciting and motivating to use the product?

Novelty - Is the product innovative and creative?

The attractiveness and novelty questions used were directly the ones provided by the UEQ but for the rest, it was used similar ones based on the System Usability Scale (SUS), which presents a five-point scale numbered from Strongly Disagree to Strongly Agree [17].

The final questions of the survey focused on the user's personal opinion where they were inquired for positive and negative aspects of the Reporter, ideas to improve the platform and any last observation.

The survey had a total of 13 questions: 4 questions for usability, 5 for user experience and 4 for personal opinion respectively.

6.2 Results and Discussion

This section will present the results obtained regarding the evaluation of the first version of HoliRisk's Reporter module, after doing the 20 tests with real users. All users completed the tasks on the guideline and answered a survey.

6.2.1 Usability

After developing a Reporter module to be as friendly and intuitive as possible, we wanted to infer its actual usability, i.e., we want to understand if the developed application is easy to interact with, if the functionalities available are useful for the problem at hand and if the information shown on each part of the platform were helpful for the completion of the tasks.

During the evaluations, it was registered the time that the users took to accomplish each of the tasks separately, the errors or mistakes made during specific tasks and if they managed to recover from those mistakes. While in the survey, the users classify the perceived difficulty of the asked tasks using a 4 point scale (1.Very Difficult, 2.Difficult, 3.Easy, 4.Very Easy). Besides, it was also measured the usefulness of specific functionalities and aspects available in the application with a 4 point scale as well (1.Should not be available, 2.Useless, 3.Useful, 4.Very useful). In addition, using a 5 point scale (1 being strongly disagree and 5 strongly agree) the users also classify their understanding (i.e. if they knew what was happening on the framework while following the tasks) and satisfaction regarding the whole experience (1 being very unsatisfied and 5 completely satisfied).

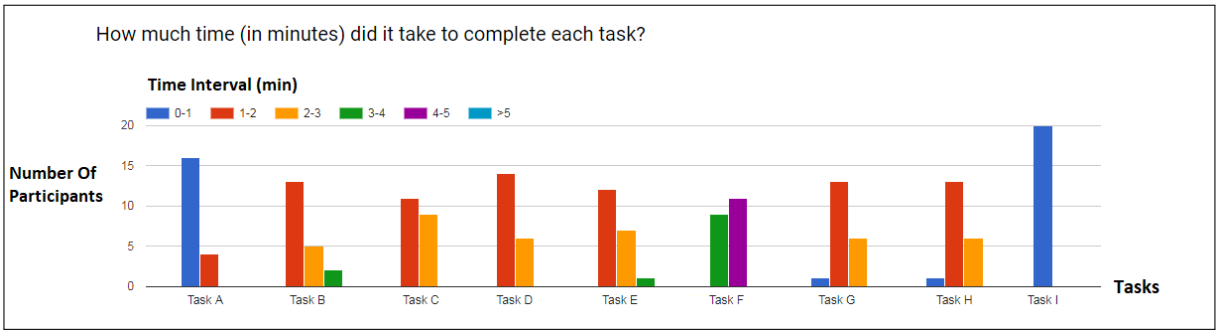
Considering the Figure 6.1(a), the overall users took approximately 15 minutes to complete all the evaluation tasks being that per task users took in average:

A - between 0 and 1 minute;

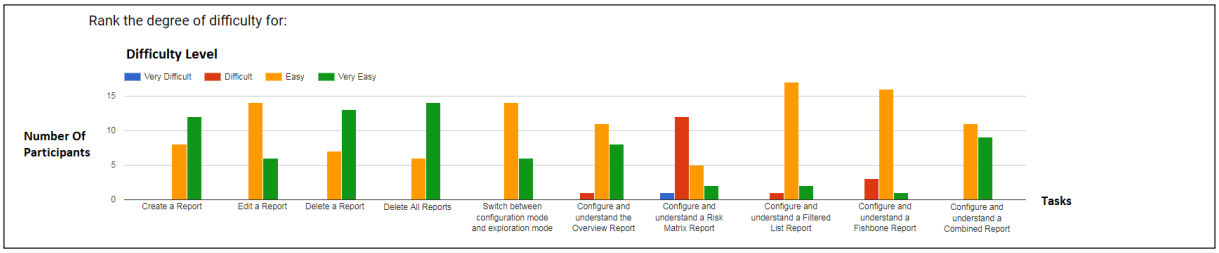
B - between 1 and 2 minutes;

C - between 1 and 2 minutes;

D - between 1 and 2 minutes;



(a) Time spent in each task



(b) Level of difficulty on each task

Figure 6.1: Tasks' completion time and difficulty level

E - between 1 and 2 minutes;

F - between 4 and 5 minutes;

G - between 1 and 2 minutes;

H - between 1 and 2 minutes;

I - between 0 and 1 minute;

As expected, users took the longest on task F where it was asked to create and configure a Risk Matrix report, as well as understand its outcome. On the other hand, the tasks the users completed more quickly were A and I which corresponded to create an Overview and a Combined report respectively.

Regarding the level of difficulty, in average the users found the tasks easy to do except for task F (Configure and understand a Risk Matrix report) which was ranked as "Difficult" due to its complexity and number of steps needed to get the final report. This can be seen on Figure 6.1(b).

When it comes to useful features, users found the side panel informations and the risk matrix steps useful for the understanding and completion of some tasks and very useful the options of deleting all the reports at once as well as the two existing types of interaction - configuration and exploration modes (see Figure 6.2).

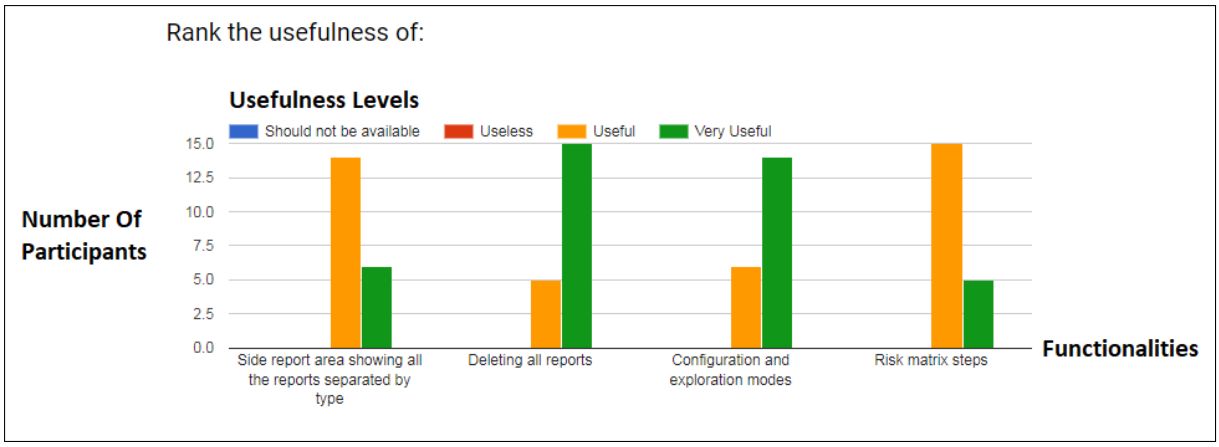


Figure 6.2: Features usefulness

In terms of being aware of what was happening in the application during all the process of evaluation and the level of satisfaction that the interaction with the system brought to the users, the answers were unanimous, 60% of the inquired agreed that they were aware of the state of the application at every moment and felt very satisfied when interacting with it. 35% of the total users felt completely satisfied after using the Report module (see Figure 6.3).

Overall, all users completed every single task successfully and with a feeling of satisfaction. The majority of the users did not make any mistakes or eventual errors, but the ones who did it, in the end, managed to recover or undo the respective mistakes. Besides that, the time spent on most of the tasks was low (1 to 2 minutes in average) which means that the users were able to get used to the application.

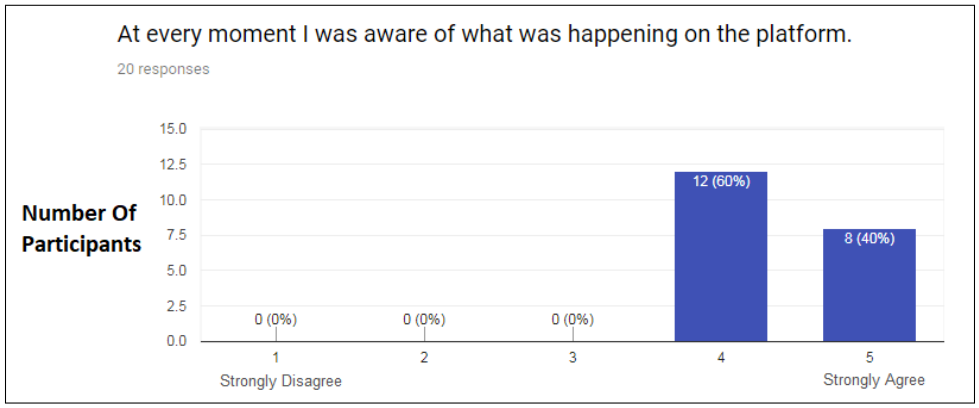
It can be concluded that the Reporter module's usability is quite good and all the 5Es dimensions were achieved.

6.2.2 User Experience

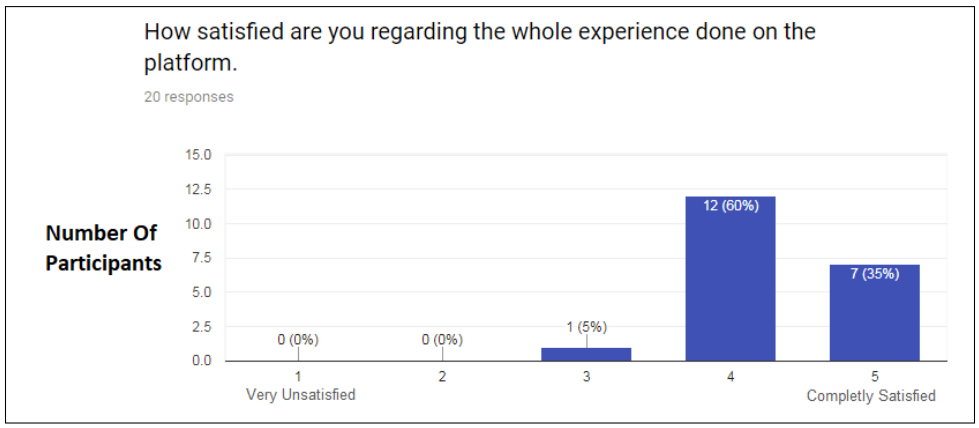
In the survey, the users answered five questions classifying each question using a 5 point scale (from 1 to 5, 1 being strongly disagree and 5 strongly agree). The answers to three of the five questions can be seen on Figure 6.4.

Considering the obtained results, it can be observed that the user experience was very satisfactory based on the majority of the answers - 80% or more of the users selected 4 and 5. Overall, users found the platform easy to use with a pleasing and friendly interface. They considered that the platform was efficient and also felt very confident when using the system.

Regarding innovation and creativeness of the new module, users found it a very modern web interface and similar to some web applications that they usually use, which allowed them to adapt quickly to this new platform. Thus, the 6 scales (i.e. Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation and Novelty used by the UEQ) to be evaluated were achieved successfully.

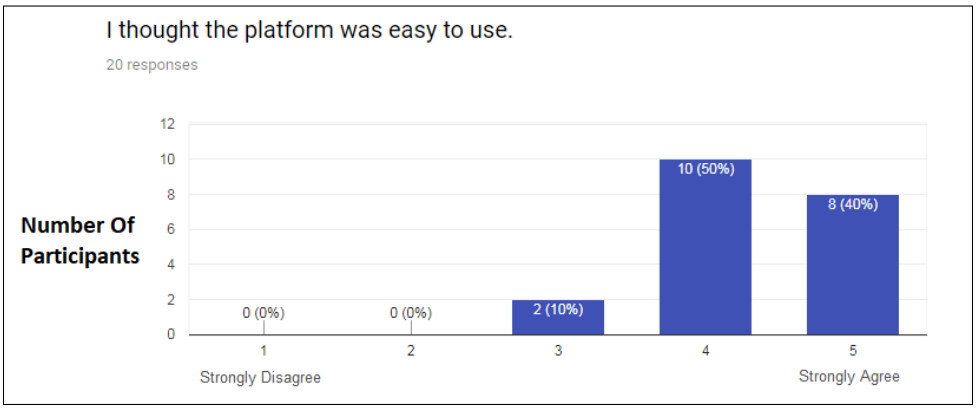


(a) Users' awareness

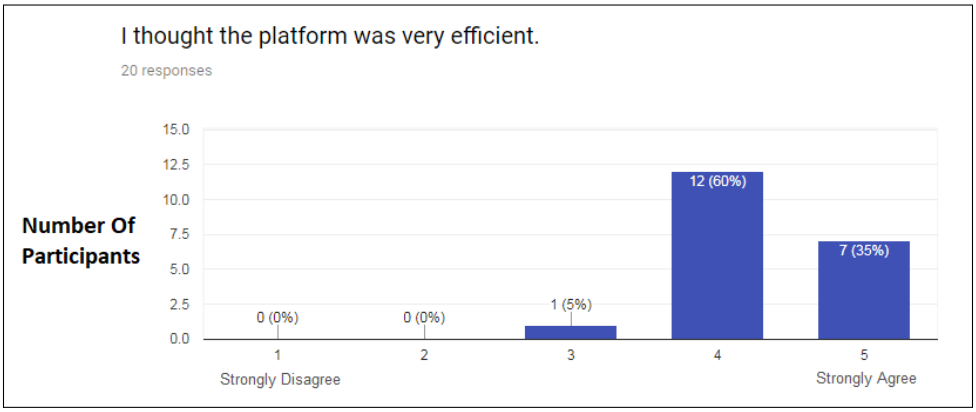


(b) Users' satisfaction

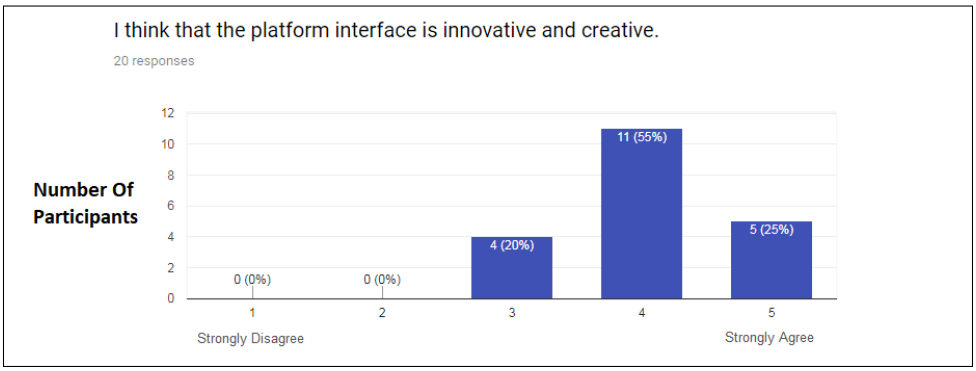
Figure 6.3: Level of awareness and satisfaction



(a) Platform's easiness



(b) Platform's efficiency



(c) Platform's innovation

Figure 6.4: User Experience Results

6.2.3 Opinions and Observations

The users provided their feedback and opinions in person and by answering the questions of a survey.

As mentioned previously, users stated the positivity of the platform such as the simplicity of its design, as well as its easiness to use and understand. Furthermore, users accustomed to dealing with big data analytics found it very quick to respond to users' commands and also when switching to the exploration mode to expose the reports' results.

Overall, users found it simple to use, with very intuitive controls and a modern design.

Some suggestions for new improvements were received such as changing the Risk Matrix steps from using the "Build" button to a wizard that allows the user to go through the steps one by one, in separate windows; changing from small icon buttons to more visible ones regarding the buttons to switch from configuration to exploration mode and vice-versa; finally add a help section containing a brief explanation for each report type.

Some suggestions can be considered for future work as they are good ideas to keep in mind and can improve the application. The positive feedback received regarding the whole module shows its potential and future growth.

6.3 Summary

This chapter presented the evaluation of the developed solution where each of the 20 participants completed the given tasks. Then, they answered a survey regarding their experience during the tasks and the platform itself and offered some personal opinions and suggestions to improve it.

Most of the tasks were classified as easy except for the Risk Matrix report that some found to be a bit complex. Regarding the information and functionalities available in the Reporter module, users found them all useful to conclude the tasks with success. The users understood what was happening in the web application and in the end, users were satisfied with the whole experience.

Overall, the users saw the platform as easy to use with a pleasing, friendly and also efficient interface. In addition, some improvements were suggested such as changing the Risk Matrix steps into a wizard design pattern. Regarding the usability and user experience, it showed to be very satisfactory which indicated that the developed solution achieved its main objectives.

The next chapter will conclude this dissertation by summing up what has been said in this work and state what can be done in the future.

Chapter 7

Conclusions and Future Work

This chapter concludes this dissertation and presents some future steps than can be taken afterwards.

7.1 Conclusions

Nowadays multiple companies have a tool to help manage risks but new ones that are just starting are stuck with the same solutions without being able to have a tool oriented towards the business and the way that the company thinks that risk should be managed. The solution would be to hire a consulting company that creates a custom tool to handle Risk Management. However, there is an alternative with HoliRisk, a holistic platform that can implement Risk Management in any company and adapt to its needs.

The addition of a data analytics component to the HoliRisk is an important asset to have and to offer to any enterprise. Before, it was only possible to manage risks by seeing a list of objects created but with the Reporter one can see this data in multiple other forms. It is possible, for example, to know the number of risks, to verify that each risk has a control associated with it, and a whole lot more knowledge can be attained using the new data analytics component.

There were implemented five types of reports, Overview, Filtered List, Fishbone, Risk Matrix and Combined, that allow the users to obtain important knowledge that they would take much more time to find or would not find at all. The Reporter gives not only knowledge to Risk Owners, but also a manner in which they can improve their decisions and their decision timing. The right decisions made at the right time will make a difference for the company's success or failure.

Regarding the user interface, it seemed to be pleasing for all users and most of them even called it intuitive. Even for users that were not aware of risk management processes, it was straightforward for them to complete all tasks and make conclusions from the reports.

7.2 Future Work

After the work done in this dissertation, the possible future steps are:

Updating the access management, which is a component that was not a focus on this dissertation, in order for users to have roles and depending on those roles, the users should be granted or denied access to specific pages or actions in the platform. Specifically in the Reporter, there could be the Risk Expert role where it would have access to every action and would be in the configuration mode, whereas a Risk Owner role would have no access to any action regarding creation or edition of reports;

Adding a help area, similarly to what exists in the Overview report, where is a brief introduction for the user that is firstly trying to understand the platform. This area could be used to better explain the basis of each type or report;

Creating a second Fishbone report that instead of exposing an attribute's function dependencies, it would expose all the risks that could be the cause of a chosen risk. Finally, some of the suggestions made by the users could also be taken into account for further improvements.

Bibliography

[1] ISO/FDIS. ISO 31000:2009 Risk management - Principles and guidelines. *Risk Management*, 31000:24, 2009.

[2] I. O. f. S. ISO. Guide 73 Risk management — Vocabulary. *ISO Standard*, page 15, 2009.

[3] AS/NZS. Risk Management. *Risk Management*, 2004.

[4] D. L. Olson and D. Wu. *Enterprise Risk Management Models*, volume 12. 2010. ISBN 978-3-642-11473-1. doi: 10.1007/978-3-642-11474-8. URL http://www.springerlink.com/index/10.1007/978-3-642-11474-8{%}5Cnhttp://link.springer.com/10.1007/978-3-642-11474-8.

[5] IEC/FDIS. ISO/IEC 31010:2009 Risk management - Risk assessment techniques. *Risk Management*, 31010:92, 2009.

[6] C. F. C. Martins. *HoliRisk - Plataforma de Apreciação de Riscos*. PhD thesis, 2008.

[7] E. B. C. d. G. e. Melo. *Plataforma de Apreciação de Risco*. PhD thesis, 2017.

[8] ISO/TR. *ISO 31004:2013 Risk management — Guidance for the implementation of ISO 31000*. 2013.

[9] J. Nielsen. 10 Usability Heuristics for User Interface Design, 1995. ISSN 0897916506. URL http://portal.acm.org/citation.cfm?doid=259963.260333{%}5Cnhttp://www.nngroup.com/articles/ten-usability-heuristics/.

[10] J. Crinnion. *Evolutionary Systems Development: A Practical Guide to the Use of Prototyping Within a Structured Systems Methodology*. Perseus Publishing, 1992. ISBN 030644139X.

[11] S. Lavalle, E. Lesser, R. Shockley, M. S. Hopkins, and N. Kruschwitz. Big Data, Analytics and the Path From Insights to Value. *MIT Sloan Management Review*, 52(2):21–32, 2011. ISSN 15329194. doi: 10.0000/PMID57750728.

[12] R. Tarjan. Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing*, 1(2): 146–160, 1972. ISSN 0097-5397. doi: 10.1137/0201010. URL http://epubs.siam.org/doi/10.1137/0201010.

[13] E. N. I. S. O. 9241-11. Ergonomic Requirements for Office Work with Visual Display Terminals. *Usability (Principles)*, pages 1–11, 2010. ISSN 0 580 34318 9. doi: citeulike-article-id:3503574.

[14] International Organization for Standardization. ISO 9241-210: Ergonomics of human–system interaction - Human-centred design for interactive systems. 2010.

[15] W. Quesenbery. Dimensions of Usability: Defining the Conversation, Driving the Process. *Proceedings of the Usability Professional's Association (UPA) conference on Ubiquitous Usability*, 2003.

[16] M. Schrepp, A. Hinderks, and J. Thomaschewski. Applying the user experience questionnaire (UEQ) in different evaluation scenarios. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8517 LNCS, pages 383–392, 2014. ISBN 9783319076676. doi: 10.1007/978-3-319-07668-3_37.

[17] J. R. Lewis and J. Sauro. The factor structure of the system usability scale. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5619 LNCS, pages 94–103, 2009. ISBN 3642028055. doi: 10.1007/978-3-642-02806-9_12.

Appendix A

Scenario - Pizzeria Under Risk

Pizza Under Risk (PUR)

PUR is a pizzeria that serves food in-house, in one dining room, and makes home deliveries.

One headmaster, assisted by one waiter, serves the dining room. The waiter assures the home delivery service, for which a motorcycle is available. A website, hosted as a service at a service provider, receives the home delivery orders. As part of the same service provided by the service provider, the headmaster receives by SMS, on a mobile phone, the information of each order submitted in the website.

PUR has one kitchen, served by one chef, accountable for all orders, which he receives from the headmaster. One assistant assists the chef, and one trainee assists the assistant (the assistant executes tasks under the request of the chef, and the trainee executes tasks under the request of the assistant).

When an order is complete, the chef announces it to the headmaster. If the order is for an in-house service, the headmaster can serve it himself, or can ask the waiter to do that. If the order is for a home delivery service, the waiter delivers it.

The assistant is responsible by the permanent account of the stoked kitchen ingredients, and the chef is responsible by the orders to the providers to refurnish that stock. The trainee is responsible for the cleaning of the kitchen and the waiter for the cleaning of the dining room and for checking the operationally of the motorcycle.

The following are the main goals of PUR:

- **O1. Reputation**: Ensure it maintains a good reputation concerning trust in hygiene and customer intimacy;
- **O2. Financial**: Ensure that expenses and revenues are controlled in order to guarantee financial sustainability;
- **O3. Quality**: Ensure it maintains a consistent quality of service, concerning the characteristics of the food and service time.

Definitions:

- **Customer intimacy**: A marketing strategy where a service supplier or product retailer gets close to their clients. The benefits of greater *customer intimacy* for a business might include improved highly tailored problem solving capabilities and greater adaptation of products to *customer* needs, as well as higher *customer* loyalty levels (www.businessdictionary.com/definition/customer-intimacy.html)

Scenario Core Assessment

The objective of this scenario is to perform a Risk Assessment considering the domain model in Figure 1, and the assessment provided in annex (Excel file) and the request for the following reports for communication:

- Ordered list of objects for each of the classes of concepts, informing on the related objects according to the respective classes
- Causal analysis for each object of the class “Objective”

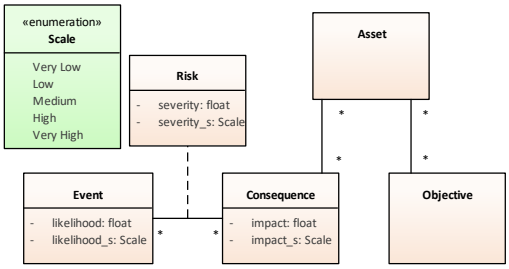


Figure 1: Scenario Core

Scenario Full Treatment

The objective of this scenario is to perform a Risk Assessment considering the domain model in Figure 2, and the assessment provided in annex (Excel file) and the request for the following reports for communication:

- Ordered list of objects for each of the classes of concepts, informing on the related objects according to the respective classes
- Causal analysis for each object of the class “Objective” and for each object of the class “Cause”

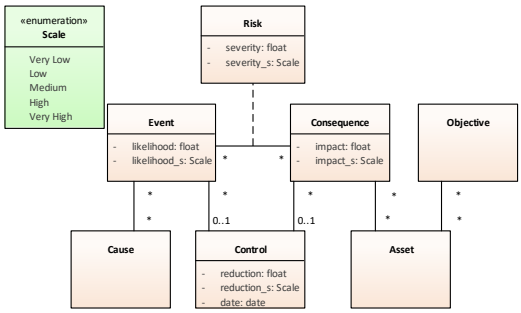


Figure 2: Scenario Full Treatment

NOTE: The rule for the calculation of a control reduction is:

- If the control is applied to a consequence:
  - impact after control = impact - (impact * reduction)
  - impact_s after control must be the qualitative value resulting from the definition of the qualitative scale for the impact
- If the control is applied to an event:
  - likelihhod after control = likelihhod - (likelihhod * reduction)
  - impact_s after control must be the qualitative value resulting from the definition of the qualitative scale for the impact

Appendix B

User Manual

Plataforma HoliRisk

Manual de utilizador

INESC-ID

Appendix C

User Evaluation - Guideline

+ RQ5 LVN 5 HSRUWMU(YDOXDWRQ

+ RQ5 LVN LV D ZHE EDVHG SODWRUP WKDWDLP V WR KDQGOH HQJMSULVH ULVN P DQDJHP HQW 7KLV HYDOXDWRQ IRFXVHV RQ WKH UHSRUWMU VLGH RI WKH SODWRUP LQ RUGHU IRU ULVN P DQDJHP HQW VSHFLDOLVW WR EH DEOH WR EHWWU XQGHUVWDQG WKH VLXDWRQ RI LW FRP SDQ\ DQG KRZ WR KDQGOH LW7KH HYDOXDWRQ P DLQ JRDO LV WR HYDOXDWH WKH UHSRUWMU XVDELOLW DQG LQMUJFH EXW DOVR HYDOXDWH WKH XVHU H[SHULHQFH ,WFRQVLVW RQ VHYHUDO DVMV WR EH FRP SOHWHG DIWMUWKH P DLQ VHVXS LV GRQH DQG ILQDOO D TXHVWRQQDLUH WR EH DQZ HUHG 7KH HYDOXDWRQ VKRXOG WDNH EHWZHHQ P LQXWV

1 RWH WKH + RQ5 LVN Z DV GHYHORSHG WR UHJLWMUULVN UHJDUGLQJ VPHFLLF GRP DLQV 7KH UHSRUWMUZ DV LP SOHP HQWG LQ RUGHUWR DOOZ D EHWWU XQGHUWDQGLQJ RI WKH GDWD LQVHUWG XVLQJ GLIHUHQWWSHV RI UHSRUW WR DVVHVV WKH ULVN RI D FRP SDQ\ 7R WHVWWKH UHSRUWMU GDWD Z DV DOHDG\ LQVHUWG XVLQJ WKH XVH FDVH RI D SI]] HUD RQ WKH GRP DLQ 3I]] HUD 8QGHU5 LVN 385

7KDQN\ RX IRU\ RXUWP H DQG FRODERUDWRQ

6 HVXS

%HIRUH VWDUWQJ WKH VDMNV WKH IROOZ LQJ VHVXS VKRXOG EH GRQH

- + DYH D FRP SXWMUZ LWK ,QMUQHWDFFHVV
- 2 SHQ D EURZ VHU & KURP H
- (QWMUWKH + RQ5 LVN 3 ODWRUP

7 DVNV

$ IWMU GRLQJ DOOR WKH VHVXS UHTXLUHP HQWV \ RX FDQ VWDUWGRLQJ WKH WDVNV EHORZ RQ WKH SODWRUP IROOZ LQJ WKH RUGHU GLVSOD HG 7KH 8 VHU0 DQXDOJLYHQ LV IXVWWRU FRQVXOWQJ LQ FDVH \ RX GRQ¶W NQRZ KRZ WR GR D VWMS IURP WKH WDVN 7KHVH WDVNV VKRXOG EH GRQH LQGLYLGXDOO DQG LQ WKH HQG WKH UHVSHFWYH TXHVWRQQDLUH VKRXOG EH DQZ HUHG

6 FHQDULR / RJ LQ LQWR + RQ5 LVN XVLQJ WKH FUHGHQWDOV SURYLGHG DQG JHWDFTXDLQWHG Z LWK WKH LQWMUFH
7DVN $
/ RJ LQ Z LWK WKH FUHGHQWDOV
D 8 VHUQDP H 7HVWMU
E 3 DVZ RUG UHSRUWMU
6 HOHFW WKH GRP DLQ 385

7KH OHIWVLGH LV ZKHUH WKH GRPDLQ ZDV FRQILJXUHG DQG SRSXODWHG 7KH ULJKWVLGH LV ZKHUH WKH GRPDLQ GDWD LV XVHG WR REWDLQ UHSRUWV XVHIXO IRU GHFLVLRQ PDNLQJ

* R WR WKH FRQILJXUDWLRQ VHFWLRQ RI WKH 5HSRUWHU

6FHQDULR 8QGHUVWDQG WKH PHFKDQLVP RI WKH UHSRUW PDQDJHPHQW XVLQJ D UHSRUW RI W\SH 2YHUYLHZ DV DQ H[DPSOH

7DVN %

7KH RYHUYLHZ UHSRUW LV DOZD\V VHOHFWHG RQFH LQ WKH UHSRUWHU DUHD &KDQJH WKH WLWOH WR 2YHUYLHZ

6DYH WKH FKDQJHV

&KDQJH IURP WKH FRQILJXUDWLRQ PRGH WR H[SORUDWLRQ PRGH ZLWKRXW OHDYLQJ WKH UHSRUWHU DUHD WRS ULJKW FRUQHU RI WKH UHSRUW DUHD

8QGHUVWDQG WKH UHVXOWV DQG ZKDW EHQHILWV WKLV UHSRUW EULQJV

([SRUW WKH UHSRUW WR 3') 2SHQ DQG YHULI\ WKDW WKH FRQWHQW ZDV H[WUDFWHG FRUUHFWO\

6ZLWFK EDFN WR FRQILJXUDWLRQ PRGH

6FHQDULR &UHDWH UHSRUWV RI DOO W\SHV)LVKERQH)LOWHUHG /LVW 5LVN 0DWUL[DQG &RPELQHG DQG H[SORUH DQG XQGHUVWDQG LWV UHVXOWV)LVKERQH LV D UHSRUW WKDW VHUYHV WR H[SRVH WKH GHSHQGHQFLHV RI DQ DWWULEXWH H J DQ DWWULEXWH WKDW FRUUHVSRQGV WR D IXQFWLRQ PD\ KDYH GHSHQGHQFLHV RQ RWKHU DWWULEXWHV WR REWDLQ LWV YDOXH)LOWHUHG /LVW LV VHOHFWLQJ PXOWLSOH DWWULEXWHV RI GLIIHUHQW FODVVHV DQG MRLQ WKHP WRJHWKHU LQ D JULG 5LVN 0DWUL[LV D UHSRUW WKDW DLPV WR H[SRVH D PDWUL[WDNLQJ LQWR DFFRXQW D IXQFWLRQ DQG LWV D[LV &RPELQHG UHSRUW VHUYHV WR SXW WRJHWKHU PXOWLSOH UHSRUWV RI WKH PHQWLRQHG HDUOLHU

7DVN &

&UHDWH D QHZ UHSRUW RI W\SH)LVKERQH ZLWK WKH QDPH µ5LVN 'HSHQGHQFLHV¶ DQG ZLWK WKH FODVV µ5LVN¶ DQG WKH DWWULEXWH µVHYHULW\¶ WLS WKH OLVW PD\ KDYH VFUROOGRZQ

6DYH WKH UHSRUWV

6ZLWFK WR H[SORUDWLRQ PRGH DQG YLHZ XQGHUVWDQG WKH UHVXOWV

6ZLWFK EDFN WR FRQILJXUDWLRQ PRGH DQG FKDQJH WKH DWWULEXWH WR µ1DPH¶

6ZLWFK WR H[SORUDWLRQ PRGH DQG YLHZ WKH UHVXOWV

6ZLWFK EDFN WR FRQILJXUDWLRQ PRGH

7DVN '

&UHDWH D QHZ UHSRUW RI W\SH)LOWHUHG /LVW ZLWK WLWOH µ(YHQWV /LVW¶ DQG FKRRVH WKH DWWULEXWHV µLG¶ µ1DPH¶ DQG µOLNHOLKRRGBVFDOH¶ RI WKH FODVV µ(YHQWV¶ WLS LW LV PXOWLVHOHFW

D (YHQWVLG

E (YHQW1DPH

F (YHQWOLNHOLKRRGBVFDOH

6DYH WKH UHSRUWV

6ZLWFK WR H[SORUDWLRQ PRGH DQG YLHZ XQGHUVWDQG WKH UHVXOWV

6ZLWFK EDFN WR FRQILJXUDWLRQ PRGH

7DVN (

&UHDWH D QHZ UHSRUW RI WVSH)LOWHUHG /LVW ZLWK WLWOH µ(YHQW &RQWUROOHG /LVW¶ DQG FKRRVH WKH DWWULEXWHV µLG¶ µ1DPH¶ RI WKH FODVV µ(YHQW¶ DQG WKH DWWULEXWHV µLG¶ µ1DPH¶ RI WKH FODVV &RQWURO¶

6DYH WKH UHSRUWV

6ZLWFK WR H[SORUDWLRQ PRGH DQG YLHZ XQGHUVWDQG WKH UHVXOWV

6ZLWFK EDFN WR FRQILJXUDWLRQ PRGH DQG GHOHWH IRQW WKH UHSRUW µ(YHQW /LVW¶

7DVN)

1RWLFH WKDW LV QHFHVVDU\ WR SUHVV HYHU\ ³EXLOG´ DV \RX JR WKURXJK WKH VWHSV 7KHUH DUH EXLOGV IRU DOO WKH GDWD WR EH H[SORUHG

&UHDWH D QHZ UHSRUW RI WVSH 5LVN 0DWUL[

D 7LWOH µ5LVN 6HYHULWLHV 5LVN 0DWUL[¶

E)XQFWLRQ $WWULEXWH µ5LVN VHYHULW\BVFDOH¶

F $EVFLVVD (YHQW OLNHOLKRRGBVFDOH

G 2UGLQDWH &RQVHTXHQFH LPSDFWBVFDOH

H &RORUV IRU µVHYHULW\BVFDOH¶

L µYHU\ ORZ¶ 'DUN *UHHQ

LL µORZ¶ 6RIW *UHHQ

LLL µPHGLXP¶ <HOORZ

LY µKLJK¶ 2UDQJH

Y µYHU\ KLJK¶ 5HG

6DYH WKH UHSRUW

6ZLWFK WR H[SORUDWLRQ PRGH DQG YLHZ XQGHUVWDQG WKH UHVXOWV

D 6HOHFW ULVNV DQG YHULI\ WKH LQWHUFRQQHFWLRQ EHWZHHQ WKH ULVN PDWUL[DQG WKH WDEOH

7DVN *

&UHDWH D QHZ UHSRUW RI WVSH &RPELQHG ZLWK WLWOH µ5HSRUWV WRJHWKHU¶ DQG WKH IROORZLQJ UHSRUWV UHSUHVHQWHG E\ WVSH WLWOH

D 2YHUYLHZ µ2YHUYLHZ¶

E)LOWHUHG /LVW µ(YHQW &RQWUROOHG /LVW¶

F 5LVN 0DWUL[µ5LVN 6HYHULWLHV 5LVN 0DWUL[¶

6DYH UHSRUW

6ZLWFK WR H[SORUDWLRQ PRGH DQG YLHZ XQGHUVWDQG WKH UHVXOWV

6FHQDULR ,Q WKH 3885 GRPDLQ DUHD WKH UHSRUWHU FDQ EH YLVLWHG XVLQJ WKH FRQILJXUDWLRQ DQG WKH H[SORUDWLRQ UROH 7KH SUHYLRXV VFHQDULR XVHG WKH FRQILJXUDWLRQ UROH DQG WKHUHIRUH HQWHU XVLQJ WKH H[SORUDWLRQ UROH DQG XQGHUVWDQG WKH GLIIHUHQFH EHWZHHQ WKHP

7DVN +

*R EDFN WR WKH GRPDLQ DUHD

*R WR WKH H[SORUDWLRQ VHFWLRQ RI WKH 5HSRUWHU

6HOHFW PXOWLSOH UHSRUWV FUHDWHG DQG YHULI\ WKH GLIIHUHQFH EHWZHHQ WKH H[SORUDWLRQ VHFWLRQ DQG WKH FRQILJXUDWLRQ VHFWLRQ

6FHQDULR &OHDQ DOO UHSRUWV IRU WKH QH[W XVHU
7DVN ,
' HOHWH DOO UHSRUWV

6FHQDULR $IWHU ILQLVKLQJ DOO SUHYLRXV WDVNV LW LV WLP H WR DQVZ HU WKH VXUYH\
7DVN -
$QVZ HU WKH VXUYH\

Appendix D

User Evaluation - Survey

Survey - HoliRisk Reporter

HoliRisk is a web-based platform that aims to handle enterprise risk management. This survey focuses specifically on the reporter side of the of the platform where it is possible to manage multiple reports and take advantage of it to better manage the risks of a company.

The survey should only take 5-10 minutes and your answers are completely anonymous.

Thank you for your time and participation.

* Required

Task Timer

1. **How much time (in minutes) did it take to complete each task?** *
   Mark only one oval per row.

	0-1	1-2	2-3	3-4	4-5	>5
Task A						
Task B						
Task C						
Task D						
Task E						
Task F						
Task G						
Task H						
Task I						

Usability

The following questions regard the usability of the HoliRisk platform.

2. **Rank the degree of difficulty for:** *
   Mark only one oval per row.

	Very Difficult	Difficult	Easy	Very Easy
Create a Report				
Edit a Report				
Delete a Report				
Delete All Reports				
Switch between configuration mode and exploration mode				
Configure and understand the Overview Report				
Configure and understand a Risk Matrix Report				
Configure and understand a Filtered List Report				
Configure and understand a Fishbone Report				
Configure and understand a Combined Report				

3. **Rank the usefulness of:** *
Mark only one oval per row.

	Should not be available	Useless	Useful	Very Useful
Side report area showing all the reports separated by type				
Deleting all reports				
Configuration and exploration modes				
Risk matrix steps				

4. **At every moment I was aware of what was happening on the platform.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

5. **How satisfied are you regarding the whole experience done on the platform.** *
Mark only one oval.

	1	2	3	4	5	
Very Unsatisfied						Completely Satisfied

User Experience

The following questions regard user experience.

6. **I thought the platform was easy to use.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

7. **I thought the platform was very efficient.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

8. **I felt very confident using the platform.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree						Strongly Agree

9. **I found that the platform and its interface pleasing and friendly.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

10. **I think that the platform interface is innovative and creative.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

Personal Opinion

The following questions regard your personal opinion about the platform. These questions are not mandatory, but it would be appreciated an opinion regarding the negative and positive aspects of the platform.

11. **Negative aspects of the platform?**

12. **Positive aspects of the platform?**

13. **Do you have any thoughts or ways to improve the platform?**