HoliRisk - Risk Management Reporting

Diogo Estevens
diogo.estevens@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2017

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, no actual component that analysed the data and provided useful information existed. This work aimed to implement a data analytics component, named Reporter, that allows the creation of multiple types of reports, using multiple risk assessment techniques over the existing data. This Reporter's objective is to improve the way that HoliRisk helps any enterprise to thrive, providing its users with the knowledge and information necessary for success.
Keywords: Risk Management, Risk Assessment, Data Analytics, Reporting, Information

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. 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. 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 providing a better support for decision making. Therefore, it was needed a reporting component that would allow the user to create and configure reports of different types in order to acquire knowledge from them, taking into account the data inserted on the platform.

This work aimed to provide an augmentation to the HoliRisk platform in order to make available a better way to analyse existing data. The Reporter component feeds on the platform's data in order to produce knowledge or business intelligence, allowing risk management specialists to create multiple reports that use multiple types of risk assessment techniques. This will 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.

2 Related Work

The process of risk management should be a part of management, culture, practices and adapted to the business processes of the organization [4]. The activities of this process are represented in Figure 1 but the main focus of this dissertation is the risk assessment process, highlighted in green, and respective techniques.

Risk assessment improve the understanding of risks, if the objectives are achieved 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 [2].

In order to assess risk, it is necessary to go through three stages, shown in Figure 1: Risk Iden-

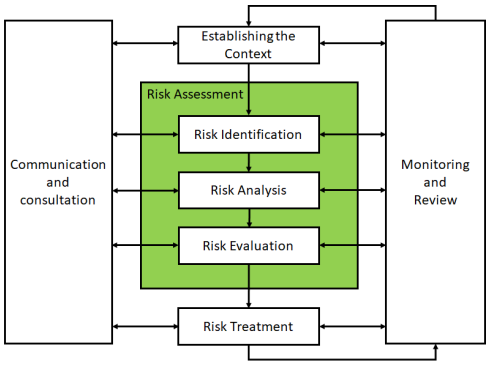


Figure 1: Risk Management Process [4]

tification, Risk Analysis and Risk Evaluation

Risk identification is the first step, where the organization should identify sources of risk, areas of impacts, events such as changes in circumstances and their causes and their potential consequences [4]. The purpose of identifying risk is to generate a list of risks based on the events that could create, prevent, enhance, accelerate, degrade or delay the achievement of the organization's objectives.

After identifying the risks, one must then analyse them - **risk analysis**. The analysis of the identified risks demand a better understanding of the risks and should try to answer the following questions: 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?

To answer such questions it is necessary to consider the causes and sources of the risks, the likelihood of those risks occurring and their positive and negative consequences. An event can have multiple consequences and therefore affect multiple objectives. Depending on the possible consequence, the risk must be contained.

Risk analysis output is to be used as an input to the next step of the risk assessment: **risk evaluation**. Risks are evaluated as to 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. This comparison will establish if there is the need for the risk to be treated.

2.1 Techniques to explore

Risk assessment process can be performed or assisted using tools and techniques. The techniques to be explored are related to the report types that the stakeholders wanted to be implemented into the HoliRisk platform: brainstorming, check-lists, cause-effect analysis and consequence/probability matrix.

Brainstorming is a technique that aims to ensure that people's imagination is triggered by the thoughts and statements of others in a group. A group of knowledgeable people of the organization, system, process or application join to *create storms in each other minds* in order to trigger an idea or conclusion that could be used to improve the topic being assessed.

Check-lists correspond to a technique used to check that everything has been covered. 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 [2].

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. Causes of contributory factors are identified often through brainstorming and displayed in a tree structure or fishbone diagram.

Consequence/probability matrix is a technique that can be applied in all steps of the risk assessment process. This technique corresponds to a matrix that is used to rank risks, sources of risk or risk treatments regarding its the level of risk [2]. These levels of risks depend upon the axes on the matrix and the values of the data. The standard corresponds to the definition of three dimensions: consequence (abscissa), likelihood (ordinate) and severity of the risk as the third dimension. In order to apply this technique and use it in a tool, it needs people, ideally a team, with expertise and as many data as available to help judge the respective consequences and probabilities.

2.2 Risk Management Tools

Nowadays, spreadsheets remain the most used Risk Management system but there are other tools that are actually oriented towards Risk Management. The tools to be presented are Resolver ERM[1] and Logicgate ERM[2].

Resolver is a software that offers a Governance Risk Management and Compliance (GRC) enterprise solution. The risk management solution is labelled Resolver ERM and is an out-of-the-box solution that is made available to any enterprise. For each step there is a possible solution and the existent configuration is the one made available on the demo (see Figure 2 [3]). If the client wants to change anything, it would be necessary for the Resolver's developers to change it, which means an extra cost. Therefore, even though the Resolver ERM does offer multiple steps regarding the risk management process, it is not a flexible software.

[1] http://www.resolver.com/
[2] https://www.logicgate.com/
[3] http://www.businesswire.com/

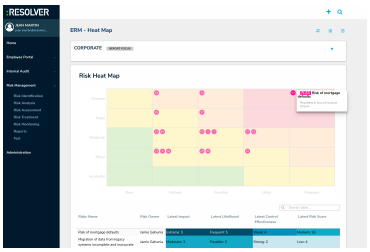


Figure 2: Resolver ERM Interface

Logicgate is another GRC solution where the risk management is centred on risk. From Figure 3 [4] it is possible to see a workflow to handle a risk. For example, 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 to handle it. Tasks are also possible to define and assign to an existent user or role. Even though the workflow can be configured, the risks are the only concept available making it a restricted software when a company wants to define, for example, events, controls or residual risks.

In sum, the existent tools may do the job but are not flexible enough to adapt completely to any organization, department or activity.

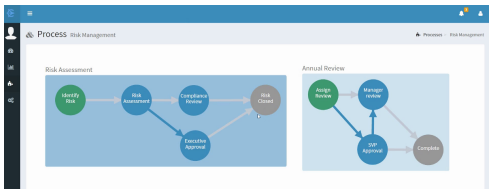


Figure 3: Logicgate Interface

3 Problem Analysis and Solution Design

In order to implement a reporting component, requirements were defined during meetings with the stakeholders of the platform. It was introduced the idea of two main modes, configuration and exploration. On the configuration mode it should be possible to configure reports and on the exploration mode, it should be 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 one that handles the configuration of the reports and tests the end result to verify that they are correct, in the exploration mode. 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 4 present the main functionalities that should be available and respective actors for the new HoliRisk Reporter. The risk owner should only be able to view and refresh the reports. The risk expert should be able to use every use case scenario.

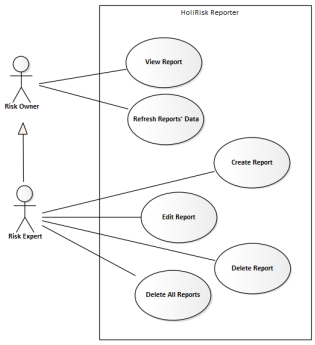


Figure 4: Reporter Use Case Model

It was decided to implement multiple report types in order to apply multiple risk assessment techniques over the data. The report types to be implemented were: **Overview** - 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; **Filtered List** - 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 previously defined reports.

3.1 Architecture

This dissertation was based on the HoliRisk previous version and therefore, its architecture was augmented with the addition of the Reporter component that is presented in Figure 5, highlighted in green.

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. The Reporter component architecture is composed of three main components: Report Management, Report Data Management and Data Analytics Engine.

[4] https://www.getapp.com/it-management-software/a/logicgate/

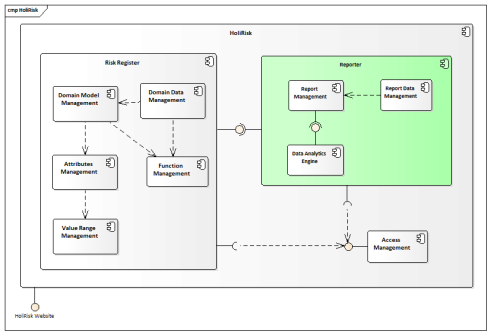


Figure 5: HoliRisk Architecture

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. There are five types of reports: Overview, Filtered List, Risk Matrix, Fishbone and Combined. Furthermore, the risk expert has to be able to do everything and the risk owner is only allowed to view/explore and to refresh the existing data.

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.

When entering the exploration mode, the **Data Analytics Engine** is called to create the information to be exposed in each report. 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 that is being processed.

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.

"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" [5].

3.2 Technology

The main technology to be used was the same as the HoliRisk platform i.e. it was used the MEAN[5] (MongoDB, ExpressJS, AngularJS, NodeJS) stack. There was a need add more dependencies on the front-end to create the necessary reports: pdfmake and html2canvas libraries were added in order to extract a HTML div as a PDF file; ui-grid library was used to expose data in a grid with multiple functionalities such as the exportation of the data on the grid to PDF and Excel files; highcharts library was used to expose risk matrixes; and d3 library was used to expose fishbone diagrams.

4 Solution Implementation

For the demonstration and evaluation part of this dissertation, a scenario was created. This scenario corresponds to a pizzeria that needs to manage its risk, hence the name Pizzeria Under Risk (PUR). The PUR domain was defined inside the HoliRisk platform, i.e. its data model, attributes, ranges and logic data were defined. 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.

4.1 Reporter Interface

The Reporter main page is divided into two main sections (see Figure 6): 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 - used to show the selected report.

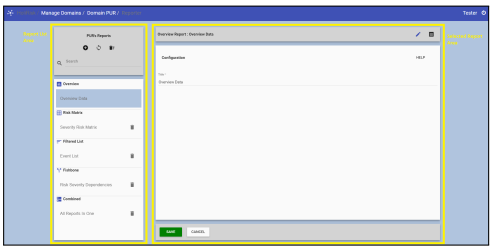


Figure 6: Reporter Interface

4.1.1 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, refresh, delete, delete all, filter.

[5]http://mean.io/

4.1.2 Selected Report Area

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. Each report can be seen in two possible modes already introduced: the configuration mode and the exploration mode. The interface allows the user to change between configuration and exploration modes as pleased and consequently, to change between the risk expert and risk owner roles.

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 configure 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 7. 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.

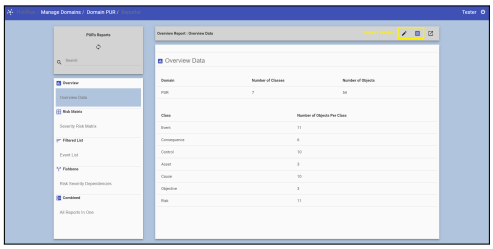


Figure 7: Report Modes

4.1.3 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.

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 the configuration of the report.

The exploration mode of the overview report 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 8). This type of report is used to assess, control and verify the numbers of existent data of the risk management process.

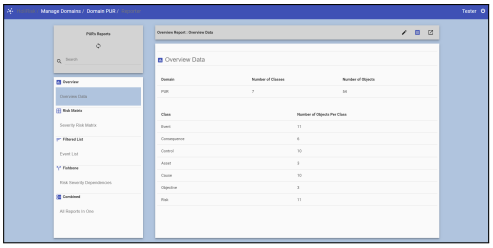


Figure 8: Overview Report Results

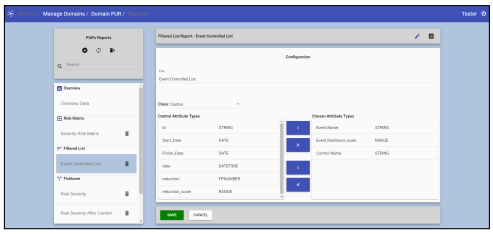
Filtered list is a report which objective is to obtain all the possible relations between all the data and expose it to the user in a grid format. The configuration necessary for this report is to define the attributes (and respective classes) to be shown in the exploration mode (see Figure 9(a)). Even 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 9(b). This was possible after developing an algorithm, similar to a depth search on a tree, that will now be described.

Firstly, it is necessary to know that what is shown in the grid are objects, the data instances of each class. 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 existing objects in order to obtain the data regarding the attributes chosen.

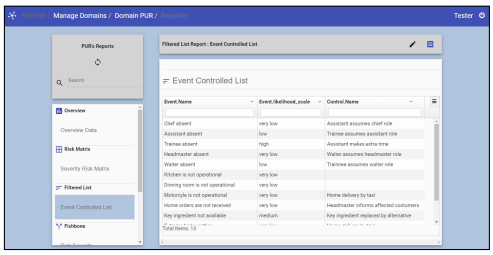
Secondly, each object can have relations to other objects depending on how the data model was defined. If an object has relations to other objects, it is necessary to go through all of them, unless the class of the object being handled wasn't chosen previously in the configuration mode. As in a Depth First Search (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 [8]. Thus, a recursive method is applied to each object to go through all the possible paths.

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

After a list of data is ready in a table format, it is necessary to remove the duplicates, which are



(a) Event Controlled Configuration



(b) Event Controlled Exploration

Figure 9: Filtered List - Event Controlled

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

The Filtered List example, in Figure 9, can work like a check-list of sorts, i.e. one can check if all the events have controls associated with it. It 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 are 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.

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 previously defined in the domain (see Figure 10) and exposes all the other attributes that the chosen attribute is dependent on. For example, if said chosen attribute corresponds to a function that has multiple arguments, those arguments are shown as a dependency of the chosen attribute.

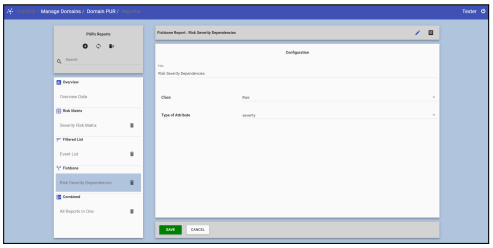


Figure 10: Fishbone Report Configuration

Once again, the configuration requires the user to set the attribute to correspond to 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, after a control, can be seen in Figure 11. 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, and so on.

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.

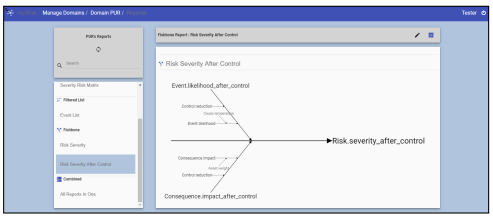


Figure 11: Fishbone Report Exploration - Risk Severity After Control Second Scenario

The **Risk Matrix** report uses the Consequence/Probability Matrix technique in order to rank risks depending on its level. Its configuration is done in 4 steps. 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 two other 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 function 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 12). 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 13). 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 14). 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 functions, and the axes to correspond to the event.likelihood and consequence.impact.

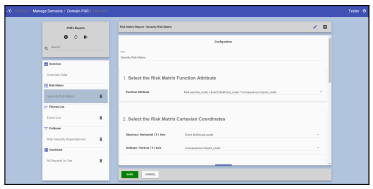


Figure 12: Risk Matrix Configuration Step 1 and 2

Figure 13: Risk Matrix Configuration Step 3

Figure 14: Risk Matrix Configuration Step 4

The exploration mode will expose the risk matrix with the colours chosen for the respective positions, according to the function results. Also, each risk is positioned according to their level of likelihood and impact (coordinates). In a real company, there will be more risks other than the 11 that were created regarding the PUR scenario and therefore, it was needed a little upscale in order to test scalability. Thus, there were inserted multiple risks and the ending result can be seen in Figure 15. In order to insert the data, it was used a spreadsheet to augment the risks registered and imported into the platform. 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 area of the risk matrix, the correspondent lines will be highlighted in the table below.

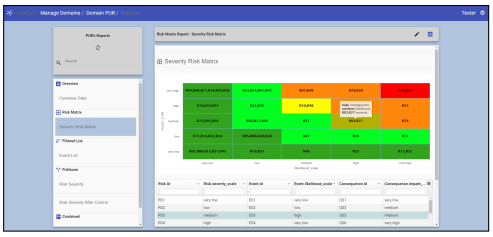


Figure 15: Risk Matrix Exploration

4.2 Combined

The Combined report is a way to view multiple reports in a single report and allow discussions and Brainstormings, one of the risk assessment techniques available. These reports can be created by different users that want to compare the reports and understand the differences, similarities or correlations.

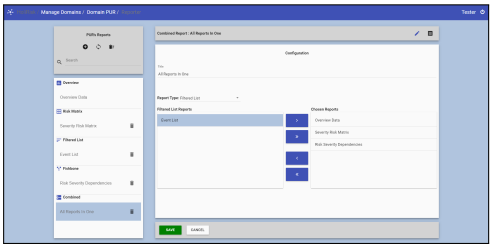


Figure 16: Combined Report Configuration

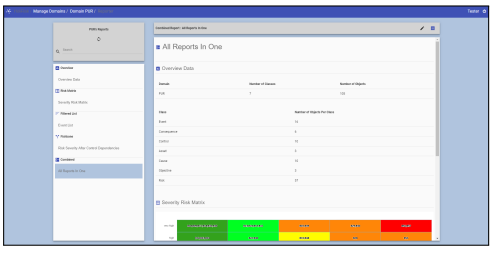


Figure 17: Combined Report Exploration

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 16). The

exploration mode of this report is the conjunction of all the reports previously selected in the configuration mode (see Figure 17).

5 Evaluation

To evaluate this work it was used a scenario of a Pizzeria to be presented to real users at the time of interaction with the platform, in order to have a domain data that would serve as the 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.

5.1 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 - introduce HoliRisk and provide the necessary documents to the user; Setup - computer with internet and the link to the HoliRisk platform; Tasks and Survey.

5.2 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 increasing level of difficulty.

5.3 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.

5.4 Survey

After all the tasks were completed, each user answered the respective survey questions which were divided into three parts: Usability - "effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments" [1]; 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" [3]; and Personal Opinion.

The HoliRisk's **Usability** was evaluated by following the five Es dimensions [6]: Effective; Efficient; Engaging; Error Tolerant; and Easy to learn.

The **UX** was evaluated by following the six points classification that measures the UX according to the UEQ [7]: Attractiveness; Perspicuity; Efficiency; Dependability; Stimulation; and Novelty.

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.

5.5 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.

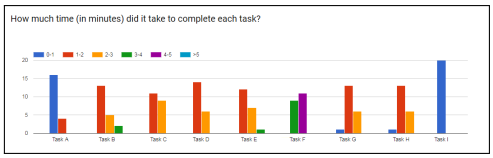
After developing a Reporter module to be as friendly and intuitive as possible, we wanted to infer its actual **usability**, i.e., we want to understand if the developed application is easy to interact with, if the functionalities available are useful for the problem at hand and if the information shown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

During the evaluations, it was registered the time that 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. 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. very unsatisfied and 5. completely satisfied).

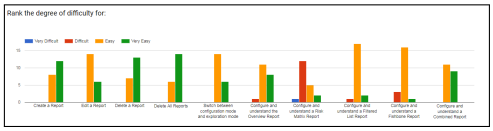
Considering the Figure 18(a), the overall users took approximately 15 minutes to complete all the evaluation tasks being that per task users took in average:

- A, I - between 0 and 1 minute;
- B, C, D, E, G, H - between 1 and 2 minutes;
- F - between 4 and 5 minutes;

As expected, users took the longest on task F where it was asked to create and configure a Risk



(a) Time spent in each task



(b) Level of difficulty on each task

Figure 18: Tasks' completion time and difficulty level

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 18(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 19).

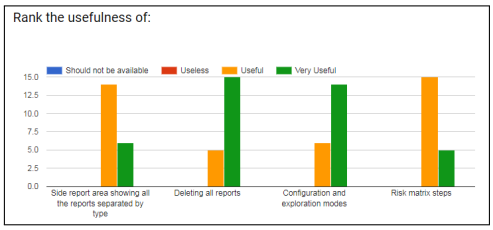
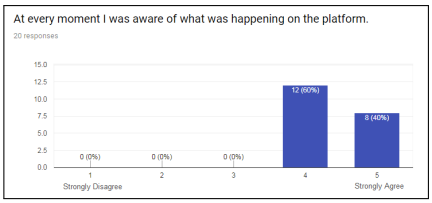


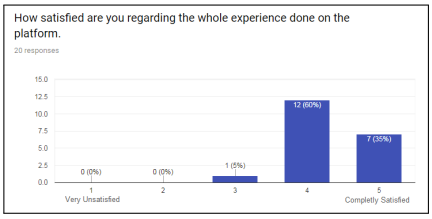
Figure 19: Features usefulness

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 20).

It can be concluded that the Reporter module's



(a) Users' awareness

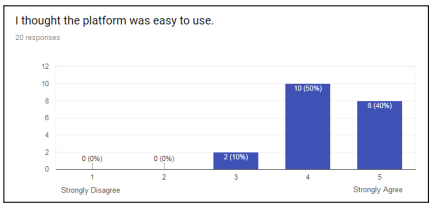


(b) Users' satisfaction

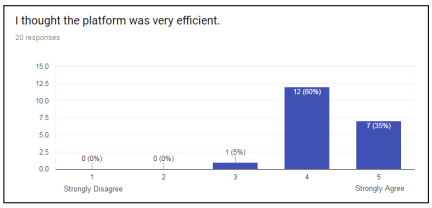
Figure 20: Level of awareness and satisfaction

usability is quite good and all the 5Es dimensions were achieved.

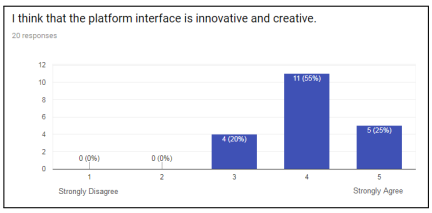
In the survey, the users answered five questions classifying each question using a 5 point scale (from 1 to 5, 1 being strongly disagree and 5 strongly agree) to classify the **User Experience**. The answers to three of the five questions can be seen on Figure 21.



(a) Platform's easiness



(b) Platform's efficiency



(c) Platform's innovation

Figure 21: User Experience Results

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.

5.6 Opinions and Observations

The users provided their feedback and opinions in person and by answering the questions of a survey. Users mentioned the simplicity of the platform's design, as well as its simplicity of usage, with very intuitive controls and a modern design. 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.

Some suggestions for new improvements were received and can be considered for future work. The positive feedback received regarding the whole module shows its potential and future growth.

6 Conclusions

The addition of a data analytics component to the HoliRisk is an important asset to have and to offer 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 that allow users to obtain important knowledge. The Reporter provides 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 very intuitive and modern. Even for users that were not aware of risk management processes, it was straight forward for them to complete all tasks and take conclusions from the reports.

7 Future Work

After the work done in this dissertation, the possible future steps can be to add 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 of report. Another possibility is to create 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.

The Reporter component was created and has now a strong foundation on which can be built an even stronger data analytics component.

References

[1] Ergonomic Requirements for Office Work with Visual Display Terminals. *Usability (Principles)*, 2010.

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

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

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

[5] 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*, 2011.

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

[7] 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)*, 2014.

[8] R. Tarjan. Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing*, 1972.