Risk Assessment Platform

Eduardo Benjamim Costeloe de Gouveia e Melo
eduardo.melo@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisbon University, Portugal

October 2017

Abstract

In the era of globalization, organizations now operate in an environment of greater uncertainty and complexity. In this environment, organizations are confronted with the need to implement processes that allow them to avoid, contain or mitigate risks that may affect the objectives of the organization. This "risk management" activity requires specialized tools to deal with the inherent complexity of the processes involved. In the market there are few applications of "risk management" and those that exist have strong limitations. This project aims to expand the functionality, usability and robustness of an application (HoliRisk) that was designed specifically for risk management, within the scope of an INESC-ID project. It is intended that the project also supports better the requirements of the ISO31000 international standard on risk management, producing a stable version for installation. For this, a method of evolutionary development was adopted through a prototype (prototyping) successively debugged by successive iterations with a test group. As a result of the project, functions were added related to the possibility of assigning a time stamp to the processes so that they reflect the ever changing environment. We have also improved usability through a BackOffice layer to verify the data entered, as well as the entire process of calculating the risk itself according to the defined parameters. We have also developed processes that allow a more free and interactive parameterization of factors, or analysis variables, for risk management. A stable version was delivered for installation at the INCM.
Keywords: HoliRisk; risk management; prototyping; functionalities; usability; objectives.

1 Introduction

In the scope of an INCM project, a web application called HoliRisk, was developped for Risk Management that would support the ISO 31000. However, this application does not yet completely support the ISO 31000 and in addition, it still has some bugs. This application can be considered a prototype, as it is still in the development stage towards becoming a stable version.

With the volatility of the external environment in which most organizations operate in the global economy, risk management becomes highly important. In this sense, an application that integrates the objectives, evaluation metrics of such objectives in function of results and the underlining logic to the organization's activities, will be of great use to organizations. With this in mind, the application intends to sustain the risk management's activities considering the above mentioned requirements and the good practices described in ISO 31000.

In order be effective, risk management should be integrated into all levels of the organization, including the top level, because it is at this level that decisions occur and the goals are established. Therefore, a risk management application should take into account these requirements and allow for the modulation and comprehension of the organization's processes and respective relations to their goals. In the present moment there are not many applications that allow this in a complete way, being that the most commonly used tool for risk management is the spreadsheet.

The purpose of this work was to create a stable application that was easy to use and more complete and also to have a stable version installed in the INCM.

2 Risk management fundamentals

According to [6], risk is "the effect of uncertainty on objectives". This factorizes the risks' nature in causes and consequences, i.e. which causes are unpredictable factors in an uncertain environment, and which consequences are the effects on the established goals, that may compromise the achievement of such goals. Still, according to [5], risk management is "coordinated activities to direct and control an organization with regard to risk". Considering the above statements, it may be deduced that risk management consists of anticipating and controlling the factors that may have an effect on the established goals.

Risk management may be viewed as an optimization process that increases the likelihood to achieve an organization's goals [4]. As such, risk management requires some primal tasks, which are: identification of the risk factors, an estimate of their likelihood, their impacts on the organization's goals and how to mitigate them. ISO 31000 [6] describes a process that approaches these tasks.

2.1 The Risk Management Process

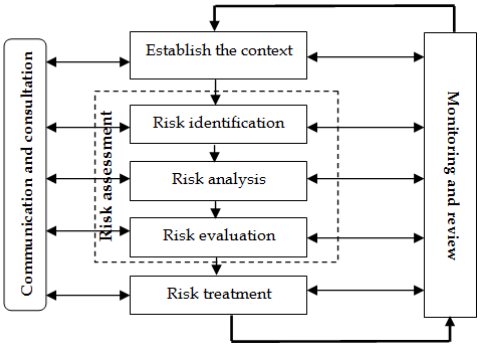

Figure 1: ISO31000: Risk Management Process (Image from [6])

This process, given by the ISO 31000, was inspired by an Australian standard, AS/NZS 4360:2004 [1], on which two elements were added: Communication and consultation, and Monitoring and review, as can be seen in figure 1. This is a generic process, giving the organizations the responsibility to adapt it to their reality.

Of all the tasks of the process there are two that we can consider continually active [6, 4]:

- Communication and consultation with internal and external stakeholders, to understand their perspective while defining the risks criteria.
- Monitoring and review to maintain the process reactive to internal and external environment changes, in order to ensure the risk control and learn from risks that occurred and controls that failed, amongst others.

The ISO 31000 process has three elements: establish the context, risk assessment and risk treatment. Risk assessment can be further divided into three sequential elements: Risk identification; Risk analysis and risk evaluation. All these elements communicate with the two above mentioned elements, that are always active. This process is illustrated in figure 1.

Establish the contexts involves analyzing the external and internal factors that may influence the established goals. With this said, it has two sub contexts, an internal and external. ISO 31000 establishes, in the internal context, the capacity to change and adapt the goals to the organization's culture, and the generic and long term goals. In the external context, the importance of the social and cultural environment in which the organization is inserted is highlighted, as well as the legal surroundings. It also includes other external factors that may compromise the organization's goals.

Risk identification is performed, essentially, by the identification of the internal and external factors of the events that may compromise the established goals.

Risk analysis involves the causes and effects of the identified risk, as well as their likelihood. ISO 31000 [6] advises that:

- The way that the consequences and their likelihood are expressed and combined, should reflect the risk type and the output purpose being used, to be consistent with the risks criteria.
- The confidence in assigning a level to a risk, it's preconditions and assumptions should be effectively communicated with people responsible to make decisions and other stakeholders.
- The risk analysis should occur with the adequate depth and be presented in the most adequate way to be processed.

The purpose of risk evaluations is to help with the decision making, dictating which risk should be treated and their priority. This involves the comparison of the risk level to the criteria established in the context.

2.2 Risk Register Concept

A risk register is a document that registers all that is produced in the process of risk management. According to [10], the risk register registers the output produced in the following processes: risk identification; risk analysis and risk treatment.

According to [3], a risk register may be structured in many different ways, depending on the organizations that implement it. Furthermore, accordingly to [10, 3], the risk register should register all events that may cause an impact on any risk, as well as possible responses to those risks.

During the activity of the organizations, the risk management process should keep its risk register up to date at all times, because it will be used in other processes, such as risk analysis and control.

As mentioned before, there are many possible structures for risk registers, most of them consisting of spreadsheets. By conducting a brief research we found a lot of spreadsheet structures and only a few applications. This document will show two examples, one application and one spreadsheet.

In figure 2 [1] the Atlassian Marketplace application is shown. This application is for managing

[1] Atlassian MarketPlace

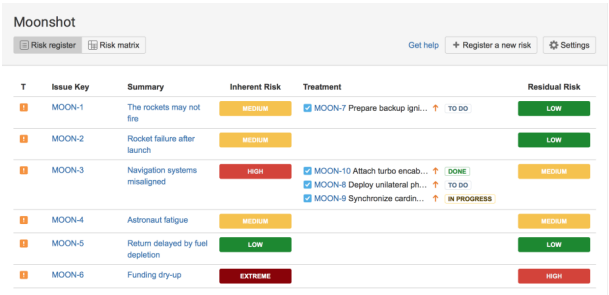

Figure 2: Risk Register: Atlassian Marketplace

projects, where one plug in that incorporates a risk register can be added. As showed in 2, and after exploring the application, this risk register was shown to be quite limited. This is because the users cannot define what is risk management for them, therefore being forced to apply the application's view of risk management.

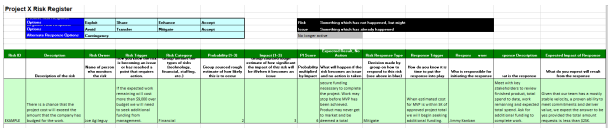

Figure 3: Risk Register: spreadsheet

In spite of having a great number of different spreadsheet templates available, like the template shown in figure 3 (image from[2]), the organizations tend to adapt them to approximate them to their reality. It is therefore unlikely to have a generic template that works with all organizations, but rather a process that establishes what is risk management for their organization. However, this becomes difficult due to the complexity of the organizations' risk management processes and reaches even higher levels of difficulty due to the limits imposed by the spreadsheets.

To conclude, it becomes very clear that there is a need to have an application that gives the user more freedom and is easier to use in the risk management processes: an application that supports the process defined in the ISO 3100.

3 Analysis and solution

HoliRisk is an application created by INESC-ID in 2014, with the purpose of being a multi-context application that may be used for a risk register, i.e. which supports the risk assessment tasks (risk identification, risk analysis and risk evaluation) for a specific context. This project is the continuation of the work performed by João Edmundo and Carlos Martins, where the market study and architecture was defined in the dissertation of Carlos Martins. [2].

[2]Spreadsheet, download link

3.1 Use cases

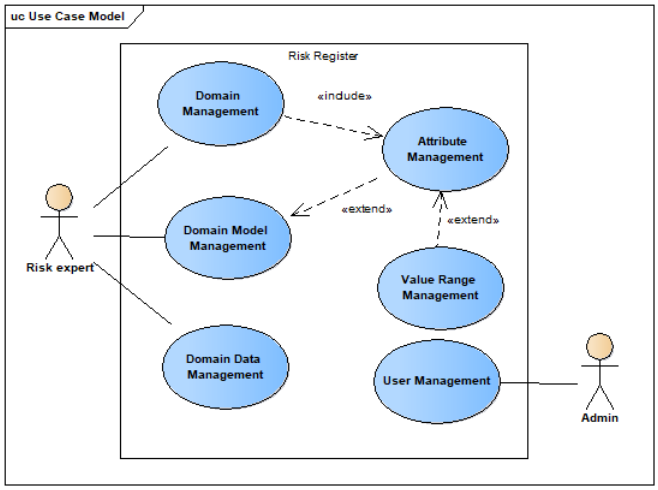

Figure 4: HoliRisk: Use cases

In figure 4 the main use of the cases is shown. The diagram represents the interactions that the users (Risk Experts) and the Admin has with the system. All the information on each use case can be found in [2].

3.1.1 User Management

According to ISO31004 [7], the risk register information is sensitive. It is therefore necessary to guarantee privacy, security and confidentiality of the information stored in the application. This is done by only allowing access to registered users.

3.1.2 Domain Management

Authenticated users are allowed to create new domains, modify and delete domains previously created by the user. A domain is a risk register created for one context and this is defined by a name, description and access privileges. In addition, it is required in order to establish the domain model.

3.1.3 Attribute Management

To have the possibility of usage of attributes by the domain model, these have to first be created. The attributes are defined by the name and type (string, long text, float, number, boolean, date and range), which are both mandatory.

3.1.4 Value Range Management

To have the possibility of usage of ranges by the attributes, they have to create them first. The ranges are defined by the name and type, which are also both mandatory. The type may be one of the following: Qualitative - Set of qualitative values are used to define categories, such as {High, Medium, Low} or {Male, Female} for example; Quantita-

tive - Set of numeric values are used as metrics in quantitative scales, such as {1,2,3,4,5} or the interval between 1 and 5 for example; Tables - Set of previously defined values that are able to define lists, although there is a necessity to create a set of more complex values that are similar to sets such as [{1,High},{2,Medium},{3,Low}] for example.

3.1.5 Domain Model Management

To define a domain model it is necessary to define the concepts and properties that define the entities and their relationships. For this, a graphic interface that allows modeling in the notations of UML [3] with three elements is used: concepts, defined by the name, which can have attributes assigned to it; Relations, defined by two concepts (source and target), multiplicity and legend (optional) and finally, attributes, which are assigned to the concepts to define properties in the assigned concept.

3.1.6 Domain Data Management

This registers all the information of the identified risk, according to the defined domain model. Subsequently, this information will be used in the risk analysis. Domain Data Management contains two features:

- Integration with spreadsheets - Taking into account that the majority of the risk data is stored in spreadsheets, the application has to allow for the import and export of data to and from a spreadsheet. In order for the import feature to be successful the spreadsheet needs to comply with a structure that the application can support. This can be done by exporting a blank spreadsheet and then copying all the information to that spreadsheet.

- Validation - To provide the user with some freedom, the application does not impose any restrictions on the import feature, or when the user is using the application to create new data. However, when requested by the users, the application must inform them, if the data respects the domain model.

3.1.7 Use case proprieties

This application has a human interface based in the web browser, the layout of which takes into account the Nielsen heuristics [8]. With this we have to consider the following properties: Efficiency and flexibility which will allow both experienced and inexperienced users to speed up their tasks; Visibility, which shows the current state and relevant information needed to complete a task; Stability, which reduces the propensity of occurrence of an error, helping the user to avoid, diagnose and recover from errors.

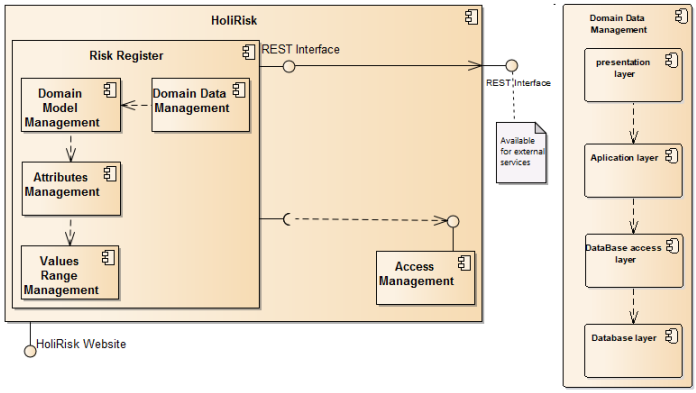

Figure 5: HoliRisk: Use cases

3.2 Architecture

In figure 5 the initial architecture is shown, which supports the application use cases. The left image of the figure shows that HoliRisk is composed of two components: risk register (supports all use cases except for User management) and Access Management (supports User Management use case), whereby risk register is further decomposed into four components, Domain Data Management, Domain Model Management, Attribute Management and Value Range Management.

The Access Management deals with the authentication and access privileges, whilst Domain Data Management, Domain Model Management, Attribute Management and Value Range Management deal with managing the domain data, all that is defined in the domain model, and all attributes and ranges, respectively. The Risk Register component provides an API, REST Interface, that will allow the external application to consult HoliRisk domain data.

In the right image of figure 5 the internal structure of all the Risk Register components is shown, such as the Domain Data Management which can be seen in the image. This structure corresponds to a layered architecture, whereby the presentation layer corresponds to the user interface treatment, the application layer corresponds to the application logic treatment, the access management layer corresponds to the bridge between the database and the application, by providing an access API and finally the database layer corresponds to the CRUD[4] operations treatment. This facilitates changing one layer without suffering huge changes throughout the application.

[3] Unified Modeling Languages
[4] Create, read, update and delete

3.3 Analysis of the initial version

This section will cover a brief critical analysis of the initial version of the application and as a result the new requisites of the application will be defined.

3.3.1 Critical analysis of the initial version

The initial version of this application supported the ISO 31000 process illustrated in figure 1 with the exception of Risk evaluation and Risk treatment. As described by the use cases that the initial version supports, the application allows the user to configure a domain model and by doing this, he is establishing a risk management context. The application also allows, after configuring the domain model, to introduce data, according to the domain model, by using the application directly. This means that the users use the create and edit features provided by the user interface, or import a valid spreadsheet and by doing this, the user is identifying the risk. However, as described in the use cases, a concept can have attributes assigned to it, and when the user is filling those attributes it is analyzing the risks. Observing the ISO 31000 process and the use cases, shows that the Domain Model Management, the Attribute Management and the Value Range Management use cases to support the 'Establish the context' task and the Domain Data Management supports the Risk identification and Risk analysis of the ISO 31000 process.

While exploring the application, several errors were found, but this article will only mention the most important errors. As mentioned in the use cases' properties, the construction of the interface took into account the Nilsen's heuristics, yet the interface allowed the occurrence of logic errors, such as introducing characters in attributes of type number, that provokes further errors in the future use of the application. This directly violates one of the heuristics: error prevention. In fact, it is because of this violation that most of the application errors occurred.

In conclusion, one major problem of the application is the importing of data from a spreadsheet. This violates one of the most important use cases features of the Domain Data Management. As previously mentioned, nowadays the most common tool used in risk management is the spreadsheet. Therefore, it is crucial that an error in the import feature of this application, be resolved with the utmost priority.

3.3.2 New requisites

As a result of the analysis, three new requisites for the application were defined, which are as follows:

- Multiple edition - It should be possible to edit the attribute value of multiple objects at the same time, if that attribute is assigned to a said object. Namely, a user should be able to choose one or more objects, and subsequently select one of their attributes and change their value as wished.

- Notion of time - Time is a crucial characteristic in risk management. Therefore a good risk management application should support the notion of time by offering a set of tools that allows the user to perform risk management contemplating time. Namely, it should allow the user to know which data is valid and insert data that will be active in the future. This application needs to support operations that involve time, making time just another variable.

- Primitive functions - In the risk management activity, there will be attribute values which are the result of an operation between one or more attributes. For example, the level of a risk is generally calculated by multiplying the probability of an event and the impact of a consequence. As the application can have a lot of data, calculating this value by hand could take a long time and therefore, in order to simplify this and save the user's time, the application should provide a set of functions to carry out these calculations automatically. The user would only be required to configure these functions.

3.4 Proposed solution

In the beginning of this thesis, Professor José Borbinha accepted the request of Diogo Stevens, a Masters student, to join this project. As a consequence, the work proposed in this project had to be divided in order to allow the accomplishment of two dissertations. Accordingly, the work for this dissertation was reduced to all aspects related to data editing, leaving the data presentation aspect to Diogo Stevens.

3.4.1 New architecture

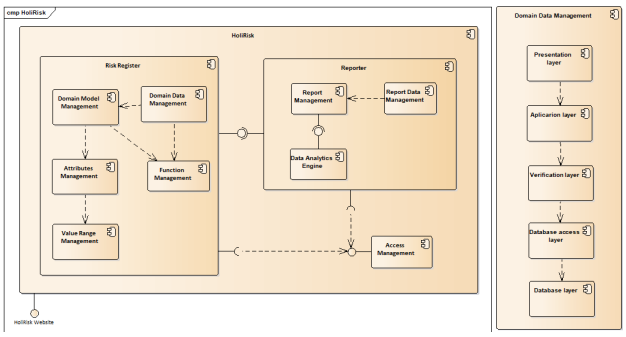

Figure 6: HoliRisk: new architecture

In figure 6 the new architecture of the applica-

tion is shown. Comparing this to the old architecture shows that the changes are minimal, but relevant. As can be seen in the left image of the figure, HoliRisk has two new components, Function Management and Reporter. The latter is part of Diogo Stevens' dissertation and as such, is not relevant to this dissertation. The function Management component was added to deal with the functions described below.

The second change, illustrated in the right image of figure 6, relates to the internal structure of the Risk Register's components. Comparing this architecture to the old one shows that there is only one change, the introduction of a new Verification layer, between the Application layer and the Database Access layer. This layer was added with the purpose of complying with one of Nielsen's heuristics: error prevention. With the introduction of this layer, all communication attempts between the application layer and the Database Access layer go through the Verification layer, where they will be verified and validated, thereby preventing user mistakes.

3.4.2 New Functionalities

With the definition of the new requirements and analysis performed during this dissertation, there are six new functionalities:

- List of changes - Maintains a record that contains all changes made to the application data with the change time stamp, enabling the possibility of backtracking a temporal line of changes of a given attribute.
- Time filters - With the introduction of the notion of time, the application data started to have a time validity consisting of three validity states (active, inactive, active in the future). With this addition, the application added filters that allowed the user to see data according to their validity state.
- Value management according to time - Allows the user the possibility of adding values to an attribute that will only be active in future dates, or was active in passed dates. The application only shows the user the value that is active at present.
- Multiple editions - As mentioned in the requisite, the application offers a mechanism to change a value of an attribute of several objects simultaneously.
- Functions - The application allows the user to configure a function and assign it to an attribute while he is configuring the domain model. Subsequently, when he is adding data, the application will automatically calculate the value of those attributes.
- References - When a user is creating a range of type table, it gives him the possibility to add a previously created range to that table, and when filling in the table, it lets him choose the value from the selected range values.
- User Preferences - On the risk register, the interface looks like a spreadsheet where the user can change the columns visibility to his preference. These changes are registered and the next time the user opens the application's risk register, it only shows the columns the user had visible when exiting the application.

4 Implementation

This section will describe how the risk register was implemented and how the main application errors were resolved.

4.1 Risk Register

As shown in figure 6, the risk register is divided into four main components: Value Range Management; Attribute Management; Domain Model Management and Domain Data Management. The application provides user interfaces for these four components, which were created to fulfill the use cases and the ISO 31000 process. The first three are related to the 'establish the context' task and the last one to the 'risk assessment' tasks.

This application was built to look like a structured data base: it allows the registration of data according to a defined structure (domain model). The application used a mixed approach when dealing with the structure compliance: it is rigid in some points and flexible in others. The selection between rigid and flexible is associated with the complexity of the use case. For example, for attributes it is rigid, but for relationships it is flexible. This is due to the relations complexity being very high and the attributes being low. Such flexibility could bring inconsistencies between the application data and the defined structure. In order to resolve this, the application has a tool that validates the data and informs the user of any inconsistencies found in the data. This tool was developed to give the user some flexibility whilst inserting data into the application.

4.2 Verification layer

This layer was added, as mentioned earlier, to prevent the user of making any mistakes. This was carried out by forcing the application to verify the data before storing it in the database. If the validation is successful the data is stored in the database, otherwise the application informs the user that the data is not valid, requiring correction before being stored into the database.

4.3 Problems with the import feature

This feature was already implemented in the beginning of this dissertation, therefore, in order to resolve this problem, it was reimplemented. After an extensive analysis to realize what the problems were, it was discovered that they were due to an asynchronous solution that cause inconsistencies in the data, due to dependencies that were not resolved.

To resolve this problem, the solution passed from an asynchronous solution to a synchronous one, where the dependencies were resolved.

Doing this had a cost: the time it took could increase, depending on the amount of data in the spreadsheet. However, it was considered an acceptable cost because the correct operation of this feature was more important than it's execution time.

4.4 Problems with managing relations throughout the application

These problems were caused by the resolution of the import feature, because they shared the same functions. The previous implementation was performed by constructing application logic into the database functions. Therefore, when the import problem was resolved, those functions were changed in order to remove the application logic from the database functions, keeping only the CRUD[5] functions, reestablishing the layered architecture. With the database functions change, the relation creation and edition throughout the application user interface, as expected, stopped working completely.

Before resolving these problems, the previous solution was analyzed in order to understand the logic behind it and replicate it in the application layer. However, this proved to be difficult, due to the method the application used to maintain the data in conformity with the defined structure. The application used void objects to maintain it's data in conformity with it's structure, but those void objects were generic (did not work with a user defined structure) and too complex for the user.

The new solution came from a previously discussed topic: rigidness versus flexibility. The decision was made to take a flexible approach and with this in mind, the void objects were removed and objects were allowed to be created without required relations, bringing the application data to a possible state of inconsistency. With this part of the problem corrected, the inconsistency problem remained. In order to resolve this, the validation tool was changed to inform the user of the data inconsistencies that needed to be corrected, i.e, the relations that needed to be created.

[5]Create, read, update and delete

5 Evaluation

Each user followed the same evaluation guide when performing this evaluation. The estimated time was approximately 40 min and was divided as follows:

- Preparation - Before the test, the users received the application's user manual to read, to enable them able to perform the test.
- Setup - At the beginning of the evaluation session, users were asked to perform a few steps, in order to be able to perform the evaluation.
- Tasks - After the setup, the users performed the tasks presented in the evaluation guide, in the order imposed in the guide.
- Survey - After the completion of the tasks, the users were then asked to fill in a survey, in order to measure the application's usability level, user experience and possible improvements.

5.1 Tasks

The proposed tasks consisted of exploring the new version of the application with the new functionalities in order to obtain feedback. As this project is a continuation of another student's dissertation project, the conceived tasks only evaluated the new functionalities: the functions and time functionalities. In addition to these tasks, there was also the need to evaluate the new implementation of the import feature.

Besides the previously mentioned reasons, these tasks were created in order to simulate what a risk expert does, according to the ISO 3100 risk management process. These tasks were based on an evaluation scenario created by Professor José Borbinha and PhD student Ricardo Vieira. With this scenario, the evaluation gained more credibility.

5.2 Participants and setup

Before starting the evaluation, the minimal number of users was stipulated, in order to obtain a good evaluation. According to [9, 11] with a minimum amount of five users, about eighty five per cent of application errors can be found. The decision was made to use ten users, because with this amount, about ninety five per cent of application errors could be found. Of these ten users, nine were masters students and one was a military professional. All the users' tests were carried out in person, which made it possible to observe their first impressions and hear their feedback whilst performing the tasks. As the setup had already been carried out, it was only necessary to provide the users with the evaluation guide.

5.3 Survey

The survey was divided into two parts, usability and user experience. This was done in order

to retrieve two types of information. The usability questions had the purpose of evaluating the application efficiency and level of difficulty completing each task, whereas the user experience questions were intended to evaluate other aspects, such as the attractiveness of the user interface, its level of interactiveness and user satisfaction.

The difference between these two measures was the capability of tasks completion and the users personal experience. Furthermore, the questions on usability were on a scale of five points whereas the user experience questions were open answers, and long text responses to obtain the user's opinion, in addition to the verbal feedback in the evaluation sessions carried out in person. In total, the survey had seven question about usability and three open answer questions about user experience.

5.4 Results and discussion

After completing all the user tests, the respective analysis and discussion regarding this new version of the application were carried out. Note that of all ten users only one of them knew the application and as such the data and observations retrieved from the evaluation were almost all from first experiences.

As mentioned before, this dissertation was a continuation of another student's dissertation, and therefore the work done in this dissertations was mostly back end. As a consequence of this, the results of the new functionalities evaluation was influenced by the previously built user interface.

5.4.1 Usability and it's problems

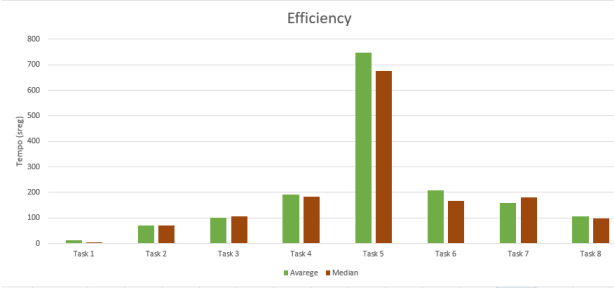

Figure 7: Evaluation: Efficiency

In the bar chart illustrated in figure 7 the average and median of each task the users performed can be seen. This chart shows that both average time and median of all tasks rounds 200 seconds, with the exception of task 5 that rounds 800 and 700 seconds respectively. These results are as expected, because task 5 was the most complex task and therefore, consumed more time than the other tasks did.

Consultation of the bar chart in figure 8 shows that task 5 was the task the users considered the most difficult, where both average and median were

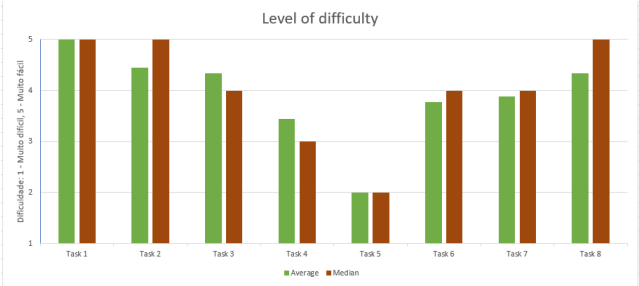

Figure 8: Evaluation: Efficiency

evaluated at 2, corresponding to level 'difficult' in the scale used in this evaluation.

Correlating the results of both charts, shows that the efficiency of the tasks is directly related to their difficulty level. Task 5 was particularly difficult due to the unappealing aspect of the user interface.

Taking into account all the tasks, the users were able to perform well. This means that the users considered the application easy to use, despite some usability problems. However, as previously mentioned, these results were, in most cases, influenced by a previously built user interface.

In conclusion, and taking into account the users' feedback during the evaluation, the application has usability problems. It is not an intuitive application, and much less so for inexperienced users who did not read the user manual. The application therefore requires a new and more intuitive interface to improve it's usability. However, the purpose of this evaluation was to verify if the new functionalities were correctly implemented, as most of them were implemented in back end. During the tasks execution, it was verified that all the new functionalities were working correctly, with the exception of one, that had an error in the data presentation. As such, in terms of the new functionalities, this evaluation presented very good results.

5.4.2 User experience

In general, the users found the interface pleasant to use, but with some problems. Many users reported that in certain places of the application, there is a low intuitive level, making it difficult to complete the tasks. Most of the problems were due to the navigation between the application menus: returning to previous pages was unintuitive. Another problem the users reported was related to the function configuration interface: in certain functions the interface was very hard to use and it triggered some users' mistakes. This can be confirmed by the unanimous response to one of the questions, where nine of the ten users stated that the hardest task was task 5 (functions configuration task). Another significant problem was related to the risk

register: the users considered it very difficult to edit objects, due to the edit button being too far from the id column, and without any highlight feature on the object's line, it became extremely difficult to follow the object line and press the edition button.

The last question in the survey was about possible improvements to the application. Contemplating all the responses, the improvement suggestions were unanimous: every user pointed out improvements to the user interface, i.e. their experience whilst using the application didn't make them think that there was something more to implement, but that the application functionalities could be better exploited if the application had a more intuitive interface. Yet, these results are from the experience of users who did not know anything about risk management This could explain the lack of suggestions about new functionalities, or alternatively, the problems in the interface directed the users' focus only to visual problems. These results will be taken into account in the next section on future work.

6 Conclusions and future work

Upon conclusion of this dissertation,it is possible to conclude that most of the objectives of this dissertation were accomplished. Starting with the objective that gives the user some freedom to modulate his risk management, this application allows the user to create a domain model and by doing this, the user is characterizing the risk management as it is in his vision. Considering the remaining objectives of expanding the analysis and evaluation phases of the ISO 31000 process, this was realized with the addition of new requisites. From these, new functionalities were derived that improved the above mentioned phases with visual characteristics, like the notion of time functionalities and the configured function execution.

The work developed to accomplish the objectives resulted in the production of a new version of the application that was more robust and more complete. In spite of the problems with the usability (the objective that wasn't accomplished, as shown in the evaluation), there was still a version of the application that was stable, and this version was installed in the INCM. However, the application still has room for improvement by providing a good solution for risk management activity.

6.1 Future Work

HoliRisk is a well developed application, but as shown by the evaluation and analysis throughout this dissertation, there is room for improvement that will improve the user's experience using the application. With this in mind, following are some suggestions to improve this application:

- User interface - According to some user suggestions, the interface should be reimplemented with a few modifications. Firstly they said that the interface for the value range management and attribute management should have a more intuitive design, and use all the space that is viewed, not having underutilized space. Secondly, in the risk register the buttons and tool bars should all be in the same location, instead of spread out through the interface.

- Function configuration interface - As stated in the evaluation, the users found this interface hard to use. This interface should be changed to a wizard-like interface, because it is more intuitive and will therefore increase the usability.

- Extension of the import and export features - The present import and export features are very limited: they only work with a given domain data. Given the fact that to import data to a domain, the spreadsheet needs to obey it's domain model. It is therefore of great value to expand those features to export and import an entire domain, making it easier to transfer data between different databases. The Admin should also be able to export and import all the application data, i.e. all the users, domains and its respective data.

- Usage modes - The application should create usage modes, i.e. have different interfaces based on the user role. Depending on the role, the application adapts the interface to that role. With this, the application can have the edition mode and exploration mode, and the choice of mode is only given to the user depending on his role.

- Language - Presently, the application only has the English language, but this application can be used in any place, therefore it should have the option of choosing the interface language.

- User preferences extension - This was one of the last functionalities that was developed and as a result, the functionality has room for improvement. There exists a lot of visual characteristics that can be added to this functionality, like the interface color scheme, the interface background or the interface resolution. There are a lot of personal configurations that will improve the user's experience.

- Domain Model editor - The application's domain model editor is very limited and slow, i.e. it doesn't provide the user a lot of functionalities and the ones that are provided are very slow. Improving this is very important, because defining the domain model is crucial

in order to be able to introduce risk management data. The new solution for this should offer more functionalities and be more efficient in the user interactions.

- Permissions categories - At present, the user's permissions are very limited: the application only verifies if the user is registered and which domains he is allowed to explore and modify. To expand this, there should be two level permissions: permission to read and permission to modify data (this permission also has the read permission). With this, the application will become more secure and complete, as it can have users that can explore some domains, but without changing anything in those domains and other domains where they can carry out modifications at their will.

- Risk treatment - To have a complete and robust application it is necessary to introduce the risk treatment plans. In order to do this, the application needs a new section to create and explore risk treatment plans, whether they are mitigation plans, avoidance plans or contingency plans.

References

[1] Australian/New Zealand Standard®. AS/NZS 4360:2004 : Risk management, 2004.

[2] Carlos Filipe Coelho Martins. HoliRisk - Plataforma de Apreciação de Riscos, Dissertação de mestrado, 2008.

[3] Celia Desmond. Project Management Tools—Beyond the Basics,IEEE engineering management review, VOL. 45, NO. 3, third quarter, Setembro,2017.

[4] Grant Purdy. ISO 31000:2009—Setting a New Standard for Risk Management, Risk Analisys,vol.30, No. 6,2010.

[5] International Organization for Standardization(ISO). ISO Guide 73:2009: Risk management - Vocabulary , 2009.

[6] International Organization for Standardization(ISO). ISO/FDIS 31000:2009: Risk Management - Principles and guidelines, 2009.

[7] International Organization for Standardization(ISO). ISO/FDIS 31004:2013: Risk management – Guidance for the implementation of ISO 31000, 2013.

[8] J. Nielsen. "Usability inspection methods", in Conference Companion on Human factors in computing System,New York, NY, USA, 1994.

[9] Jakob Nielsen and Thomas K. Landauer. A Mathematical Model of the Finding of Usability Problems, 24-29 April 1993.

[10] Project Management Institute. The Project Management Body of Knowledge ,Fifth edition,ISBN: 978-1-935589-67-9, 1996.

[11] Robert A. Virzi. Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough?, HUMAN FACTORS, 1992.