Risk Assessment Optimisation with MONARC

Fabien Mathey, Cédric Bonhomme, Juan Rocha, Jérôme Lombardi, Benjamin Joly
SMILE/CASES g.i.e.
16 Boulevard d'Avranches, 1160 Luxembourg
info@cases.lu

***Abstract*—There are many solutions to cover risk assessment: from big software solutions to standalone applications – even based on spreadsheets. A risk assessment takes a significant amount of time and becomes expensive rather quickly resulting in the impossibility for small companies to perform such an analysis. MONARC aims to make performing more accessible through optimisations and sharing. Each of these reduce the time needed to perform such an assessment and subsequently, money. Finally, after the development of the tool, MONARC has been released as open source software (GNU Affero General Public License version 3).**

In this paper, we will introduce MONARC, the optimisation and sharing features. The presentation will include a live demo of the introduced features.

***Index Terms*—risk-analysis, governance, security, cybersecurity, threat, vulnerabilities, risk-assessment, risk-evaluation, risk-treatment**

I. Introduction

There are many solutions to cover risk assessment: from big software solutions to standalone applications – the initial proof of concept for MONARC was based, like others too, on spreadsheets. However, a certain number of these solutions are either limited in functionality or tend to be difficult to use. Additionally, performing a risk assessment takes a significant amount of time and becomes expensive rather quickly. It does not help financially that the risk assessment should be done periodically and that each time recommendations usually result in the implementation of controls, which add to the cost.

To reduce the cost of a risk analysis, both in terms of money and time, experts from CASES[1] have developed the MONARC platform[2] which provides pragmatic, inexpensive optimisations. These optimisations are the result of years of experience in the field. The saved time and finances allow for more available resources to implement a recommendations.

The other bigger advantage of the MONARC solution - and that adds on top of the optimisation - is the sharing of everything; the risk models and even the source code[3]. No license fees, no special commercial/non-commercial use models etc. It is under the GNU Affero General Public License version 3.

II. What is MONARC

Depending on its size and its security needs, organisations must react in the most appropriate manner. Adopting good practices, taking the necessary measures and adjusting them proportionally: all this is part of the process to ensure information security. Most of all, it depends on performing a risk analysis on a regular basis. Although the profitability of the risk analysis approach is guaranteed, the investment represented in terms of the required cost and expertise is a barrier for many companies, especially small and medium sized enterprises (SMEs).

MONARC aims to remedy this situation by allowing all organisations, no matter the size, to benefit from the advantages that a risk analysis offers. As a method it allows to perform a repeatable and precise risk management.

The advantage of MONARC lies in the capitalisation of risk analyses already performed in similar business contexts: the same vulnerabilities regularly appear in many businesses, as they face the same threats and generate comparable risks. Most companies have servers, printers, a fleet of smartphones, Wi-Fi antennas, etc. It is therefor sufficient to generalise risk scenarios for many assets (also called objects) by context and/or business.

III. Phases of MONARC

The phases of MONARC start with the context establishment (or plan) phase, followed by the context modelling (or do) phase, the evaluations (or check) phase and finally the implementation and monitoring (or act) phase. Thus it is similar to the PDCA[4] method. The relation between each phase can be seen in Figure 1.

A. Context Establishment

The first step is to take stock of the context, challenges and priorities of the company or organization that wishes to analyse its risks. This particularly serves to identify key activities and critical processes of the business in order to guide the risk analysis towards the most important elements. To do this, a kick-off meeting is organized with the members of the management and key individuals. The goal is to know what makes the company "live" and what could destroy it, to identify the key processes, the internal and external threats as well as organisational, technical and human vulnerabilities.

B. Context Modelling

This phase includes the modelling of objects (secondary assets) and trees. The primary assets were identified in the previous phase. They must now be detailed and formalised

[1] https://www.cases.lu/

[2] MONARC: Optimised Risk Assessment Methodology by CASES: https://www.monarc.lu

[3] Github Project Page: https://github.com/monarc-project

[4] PDCA: plan-do-check-act

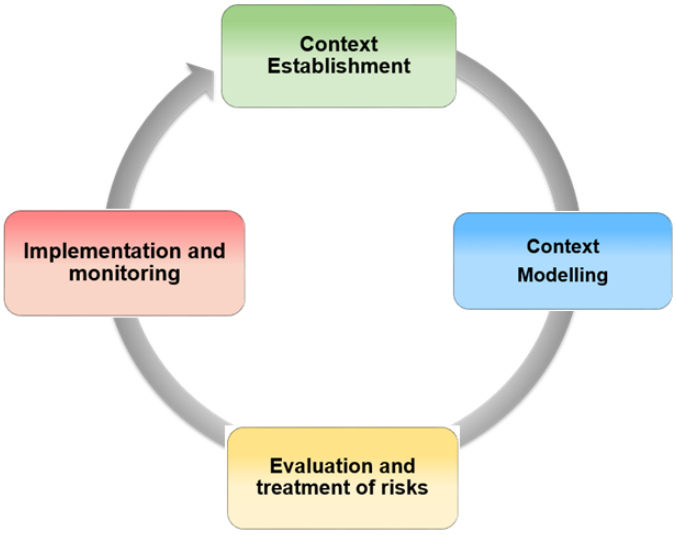

Fig. 1. MONARC, a method in four phases

in a diagram that displays their interdependence. Impacts are defined at the level of the primary assets (processes or information), following the information gathered in the context establishment phase. The secondary assets inherit the impact of the primary asset to which they are attached (object tree). The impact level of the secondary assets can be modified manually.

C. Evaluation and treatment of risks

The assessment consists of quantifying the threats, vulnerabilities and impacts in order to calculate the risks. To do this, it is necessary to have quality information about the exact likelihood of the threats, the ease of exploitation of vulnerabilities and potential impacts; hence the need to rely on metrics that have been validated by experts. When the risk assessment identifies a risk that is higher than the acceptable level (risk acceptance grid also known as risk appetite), risk treatment measures should be implemented in order to reduce the risk to an acceptable level.

D. Implementation and monitoring

When the first treatment of risks has been carried out, an ongoing management phase with security monitoring and recurring control of security measures must be entered, in order to improve it in a sustainable manner.

This fourth phase also allows to continuously optimise security by increasing the detail of objects used and by expanding the scope of the risk analysis.

IV. Optmisations

MONARC includes 5 levels of optimisations and simplifications:

1) Assets with attached risks;
2) Heritage of impacts;
3) Global assets;
4) Import and export of assets including (or excluding) the evaluation;
5) Generation of reports/deliverables.

But before we can get into the details, we should highlight how the assets and risks are defined, as they are essential to understand the optimisations possible.

There are two risk types existing:

1) Operational risks which are directly connected to primary assets. These do not have direct threats nor vulnerability and thus are evaluated differently ($impact * probability$);
2) Informational risks which are based on the relation of vulnerability, threat and impact.

This also explains that there are two types of assets:

1) Primary assets, like core business (departments, information, etc...);
2) Supporting assets, like servers, employees, buildings, etc.

In general, an asset includes a special set of risks. Looking at the example in Figure 2, we have a printing service which has a building and a printer. The building has risks attached and so does the printer. Now that the basic structure of modelling assets is known, we can touch on the different parts of optimisation.

A. Assets and their risks

The modelling of assets takes some time, especially if the risks have to be identified for each asset first. Identifying these risks takes time and a risk can be forgotten easily. Thus coming back, adding the missing risk and only then continuing the risk assessment reduces productivity. To limit the time lost, we allow to define assets with risks already attached. In that way, when the asset is added to the analysis, the risks attached to that asset are automatically added as well.

B. Heritage of impacts and risks

As previously discussed, each risk has an impact and its context is given by the asset the risk is attached to. The context of that asset is the parent asset and so forth. So the optimisation here is, since the primary asset gives the context to each of the child assets, only the primary asset needs to be impacted: the child assets will automatically inherit the impacts (using a top-down approach). It is still possible to manually evaluate each impact but for most situations, this is really accelerating the evaluation process.

In contrast to the heritage of impacts, the risks of the supporting assets are automatically inherited to the primary asset. Because if a supporting asset fails, the primary asset is impacted or fails as well. As such, the heritage is not only top-down but also bottom-up.

C. Global assets

Sometimes it is the case to have supporting assets that support multiple primary assets: a building for example. The optimisation in this case is to introduce global assets. This means that less risks have to be evaluated. As an example: the building, as it is the same for many services, only needs to be evaluated once – the one with the highest impact.

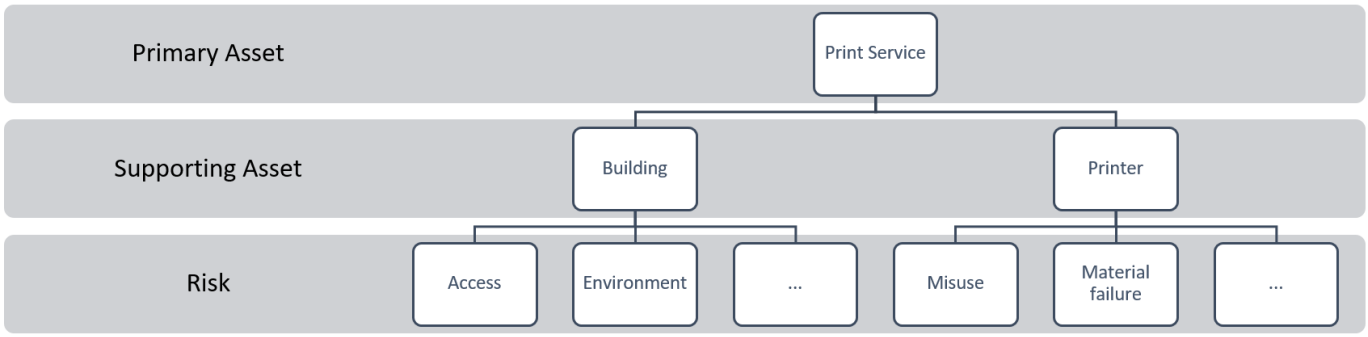

Fig. 2. Artificial Asset Example

D. Deliverable generation

Inputting all this information is great, but if it cannot be exported to report to management, all the work is less productive. Therefore, deliverables can be generated at specific moments: at the end of each phase – MONARC has four phases[5].

E. Import and export

Not only can the tool generate reports to present to management but there is an import and export feature built in. This allows for one department to create a risk model that can then be shared amongst departments. It also allows to be shared publicly of course. This again optimises and reduces the work time: the model already exists and does not need to be created first. Thus, the workload can be distributed over the community and reduces the cost of every individual organisation by some margin avoiding double work.

V. Sharing and community

Sharing is a big part of this software: sharing the code, sharing assets and risk models and finally, sharing a whole analysis. The latter part is very interesting to companies with multiple departments (maybe even international): they can ask their departments to perform a risk analysis with a certain model. Then they can combine the analysis and can define rather quickly how the company risks look in general.

Sharing assets or risk models is an interesting concept as it allows for some organisation to create an asset (or composition thereof) and it linked risks. Once such an asset has been defined, that organisation can then export that asset and make it available to any interested party. The latter can then import it and can immediately start with the evaluation process instead of first having to build the asset themselves. The sharing can of course be selective: each asset can be chosen and exported and then, the exporter can choose with whom to share.

Finally, we share the code so that everyone can enjoy what has been created by everyone. The goal is to have a tool for the community created by the community.

[5]MONARC Method Guide - https://www.monarc.lu/method-guide/

VI. Ongoing and future works

Future work will enhance: 1) the overall MONARC platform with new functions, such as a statement of applicability (SOA), improvements for the new dashboard and hopefully your ideas/contributions and, 2) the sharing of MONARC objects with a new tool and by trying to educate a community not so used to the sharing of information.

Our short to mid-term roadmap is also available[6]. We do not really have a strict long term roadmap - no false promises.

We are working on a solution[7] which will let the users share their MONARC JSON objects much easier. The source code is under GNU Affero General Public License version 3. It will also help the users to build valid security objects, thanks to validations against JSON schemas. In order to gather a maximum of security related objects in this repository the platform will accept contributions that are not only for MONARC. Furthermore security objects are editable via a Web form generated from the schemas. So that every security researcher will be able to contribute with her/his work.

VII. Conclusion

In this paper we briefly presented the MONARC method and platform, highlighting its contribution to the community: optimisations and the ability to easily share the information. Each of the presented optimisations and even the sharing part will reduce the amount of resources needed to perform a risk analysis.

During the presentation a short live demo where each of the aforementioned optimisations and sharing will be demonstrated on a dedicated instance of MONARC.

[6]https://github.com/monarc-project/MonarcAppFO/wiki/Roadmap
[7]MOSP: https://github.com/CASES-LU/MOSP