DESIGNING RISK ASSESSMENT APPLICATIONS FOR INTERNET OF THINGS PROJECTS

Vlad-Valentin Fireteanu[1], Mihai Ciuc[2]

This paper was written to document the useful and usually mandatory steps of creating a risk overview over specific Internet of Things projects. When accepting a project, it would be ideal to know what the resources are and the challenges that can occur when delivering the project. By designing a risk application or even conduct early risk assessment activities we can ease considerably our work. During my eight-year experience within this Information Technology domain, I faced a lot of challenges and discovered multiple blocking issues as delivering different projects - this was an ongoing routine during the development phase itself. Early identification of this assessment process/application results would have helped me to better estimate and implement an optimal project roadmap. The conclusions of this paper can help project managers to think ahead and better allocate resources, so it also delivers practical applicability.

Keywords: Internet of Things, Risk Assessment, Project Management, Software delivery, SDLC, Software Design

1. Introduction

We already know that, during latest years, there was a huge increase regarding the Internet of Things[3] projects. Specifically, the number of businesses that use the IoT technologies has increased from 13 percent in 2014 to about 25 percent today [1]. This is considered a significant increase that leads to significant research efforts being pointed in this new direction. Nowadays, one of the challenges is to determine if it is both feasible and productive to implement IoT projects and that implies knowing the risk factors that can occur.

First, we need to identify the need that our project should cover. After identifying it, it is important to draw a plan that contains the required steps. For a better understanding of these steps, I will map them on a specific IoT project. This is represented by an avalanche prediction module that acquires relevant environment data and computes it to generate useful reports. The software engineer should understand the environment where the project will be deployed.

[1] Faculty of Electronics, Telecommunications and Information Technology University Polyethnic of Bucharest, Romania vlad.fireteanu@gmail.com

[2] Faculty of Electronics, Telecommunications and Information Technology University Polyethnic of Bucharest, Romania mihai.ciuc@upb.ro

[3] During this article we will use the IoT acronym for Internet of Things

For this example, the deployment environment is not a risk free. We intend to gather information from a slope covered with snow, so we have multiple factors that interfere with the accuracy of our data. These factors are represented by the weather itself, wildlife interaction and the battery lifetime. Being outdoors in harsh environment conditions, all the acquiring elements can be damaged, so we need to think about protecting the equipment with different shields, boxes, fences etc. As a result of this, the overall cost of the project will increase.

The novelty of this paper is that we intend to get beyond security and measure even more risk types, each one being the deciding argument when accepting or not IoT projects. Beyond this, as presented above, we have the 'Environment' main risk area that shall be detailed and quantified in this article. Even if there are several articles that approach this subject, quantifying areas such as environment, safety, hardware resources are somehow neglected compared to the security research efforts. The main challenges include energy efficient, communication and data-related challenges, scalability, and safety [2].

After the designer understands all the environment implications, he should decide about what hardware resources should buy. Mapped on this project - the avalanche prediction module, we have only acquiring elements for data reading purposes, processing units and the physical protection used to cover the equipment from the environment conditions.

The process itself is based on a vector built around specific IoT risk factors. After grading each one of these risks we will have an overview over the whole project. This overview will help us understand the financial implications and the feasibility of the project.

The majority of the nowadays reads about risk assessment are centered around security risks that can harm both the users and devices of Internet of Things projects. Even if both consumer and industrial IoT devices are deployed every day, there is a limited number of frameworks that can assess the security risks of these devices [3]. The reason of this is because one of the most important aspects of IoT projects is being first to market. The main list of threats for applications are currently developed by OWASP which expanded their list of ten threats to include IoT devices [4], NIST[4]Draft 8200 which aims to create a set of security standards and even Microsoft who published a detailed SDLC information that includes an introductory security training, five specific stages of implementation and also a response phase. This is one of the popular ways of achieving security by design [5]. By discussing about this area, it will be easier to respond to the following questions: 'Is it worth to invest in this specific application?' 'Which are the areas that can be improved for my project to be successful?' 'How can I assess and quantify the risks of my projects?'

[4] Developed by National Institute of Standards and Technology USA (NIST)

2. Risk Assessment Parameters

The creation of risk assessment parameters emerged after identifying the required steps needed for building IoT applications. After we have a clear list with them, we need to measure the each one of it based on the current state of the IoT projects. Research must be performed for us to accurately estimate each risk parameter, by assigning a grade between a chosen interval. The general architecture of risk assessment applications can look like this:

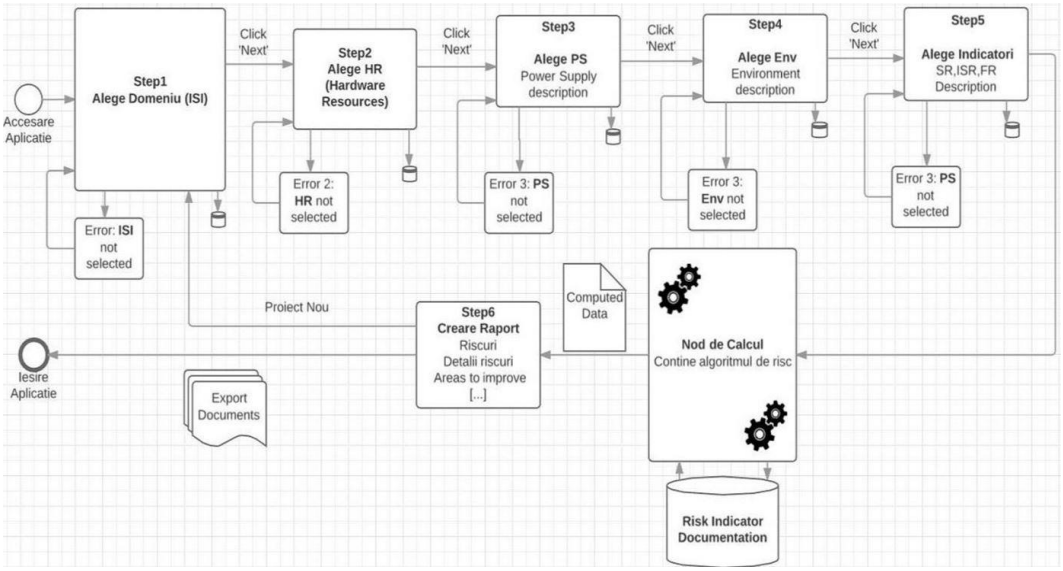

Fig. 1. General risk assessment architecture (created with Lucidchart)

As we can see from Fig. 1, we identified the most common risk parameters that can affect the IoT Projects. The actor who performs the risk assessment is prompted with a user-friendly interface where he should complete the answers to some relevant technical questions. After providing the responses, all the information will be gathered and computed through a calculus node that has different types of deliverables. Both reports will be generated (we can choose to save them as .xml, .txt, .csv etc.) and an overall risk grade will be assigned to the project. Each state of the above diagram is responsible of saving a risk type and a grade based on the project manager's answers. If a risk parameter is not provided, then an error will be thrown (Ex: 'Environment risk can't be calculated due to missing valid inputs from the user'). After this error will be thrown, the user will be prompted with the same question until all the mandatory answers are provided. By understanding the risk types and how each one of it can affect the success of a project, we will be able to model our algorithm to have a better predictability of production scenarios. Most of the parameters can be found within the following

list in Fig 2. After the major areas are defined, the next step is to identify the next level of risks in hierarchy and grade them based on threat level. This threat level is very important since is the direct result of the research activities.

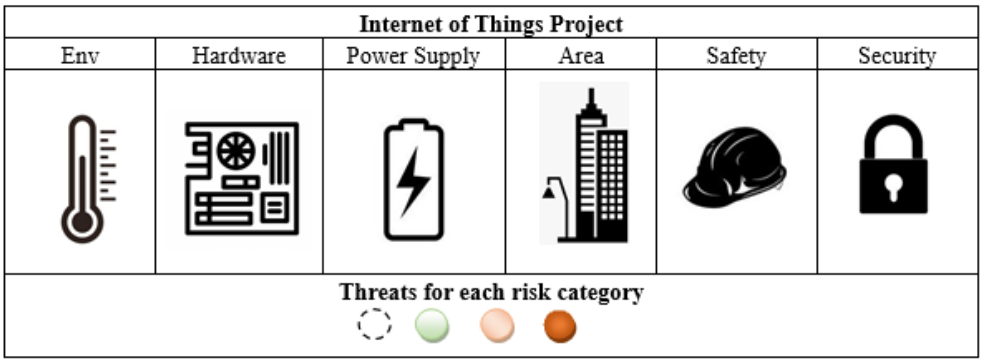

Fig. 2. IoT projects major risk areas

The threat level is graphically represented by the gradient dots that can take values between 0 and 1 because it can be easily correlated in a percentage. The value '0'is ideal and indicates that a risk is either not applicable for a chosen project or the major area is risk free [3]. As you can already assume, there is no such thing as risk free projects or areas, so we will assume different risk intervals for each one of the dots. By doing so, the final risk diagrams will be more representative and easy to follow. Based on this, we have the following values:

Risk parameter not applicable for this project (which means **0%**)

Risk with low incidence (with an interval of [**0 – 0.3**))

Risk with high incidence (with an interval of **[0.3- 1)**)

Certain risk with value **1** (which means **100%**)

The highlighted numeric values from above will be introduced in our configurable algorithm and their proportion set accordingly to the latest domain updates. This is the reason that determined us to build an algorithm that is based on a parametrization approach.

For a better understanding about how we rate a risk parameter, I will choose one from the above list. Usually, one risk parameter is centered around multiple questions that define the factors. Each of these questions is based on research performed in a specific domain and on already existing statistics that define de incidence of the risk factors themselves. The chosen risk parameter is

represented by the 'Environment' risk type and it is centered around the following questions:

1) Is the projected intended to work in harsh weather conditions?
   Risk factors: cold, heat, humidity, wind, and lightning [2, 6]
2) Are there any mechanical hazards that can harm your equipment?
   Risk factors: mechanical interaction (falling objects, landslides, moving objects, dust, snow drifts)
3) Will the project be deployed in a radiation free environment?
   Risk factors: single event effects, total ionizing dose effects [7]
4) Can the wildlife interaction interfere with the quality of the acquired data?
   Risk factor: wildlife interaction
5) Can the human interaction interfere with the quality of the acquired data?
   Risk factor: human interaction (environment data tampering, accidental data corruption)

Each of these questions has different weights depending on the project's nature itself. Below you can find a table with the assigned weight for the 'Environment' risk category.

Table 1

'Environment' weights depending on project*

Domain/ Risk	Weather	Mechanical	Radiation	Wildlife	Human	Unknown
Smart Cities	30%	30%	0%	0%	30%	10%
Smart Environment	30%	30%	0%	15%	15%	10%
Smart Watering	30%	30%	0%	15%	15%	10%
Smart Metering	20%	30%	10%	10%	20%	10%
Security and Emergencies	10%	TBD[5]	TBD	0%	30%	10%
Retail	10%	40%	0%	0%	40%	10%
Logistics	10%	40%	0%	0%	40%	10%
Industrial Control	10%	50%	10%	0%	30%	10%
Smart Agriculture	30%	30%	0%	15%	15%	10%
Smart Animal Farming	20%	20%	0%	30%	20%	10%
Domotic and Home Automation	0%	40%	0%	0%	50%	10%
eHealth	0%	10%	0%	0%	80%	10%

*Note that the sum of all weights for each row is 100%.

[5] Note that we have two TBD in the table. The percentages are shifting between two parameters: 'Radiation' and 'Mechanical' depending on the emergency nature.

$$100\% = \sum_{i=1}^{n} w_i \tag{1}$$

A table like this is needed for all the main categories presented in Fig 2. The rows are the same, but the column differ depending on the specific risk types. For example, the 'Power Supply' risk type is defined by other risks, therefore the column will have different names, such as 'Battery', 'Wall Sockets', 'Hibernation routines'. Even so, some of them will remain the same, such as 'Weather' or 'Unknown', because weather can damage the hardware but can also considerably reduce the batteries lifetime. The 'Unknown' part affects each one of the risk types and we should leave room for it because there are a lot of vulnerabilities that were not discovered yet, very similar to zero-day exploits. Zero-day is a flaw in software, hardware or firmware that is unknown to the party or parties responsible for patching and/or fixing them [8].

The main part of the research activity is represented by the accurate assessment of these percentages. As we can see, we have one highlighted row, represented by 'Smart Environment'. We highlighted this row because, as we said in the introduction part of this paper, our case scenario is represented by an avalanche prediction module. This project falls under 'Smart Environment' category and the following weight percentages apply:

Weather risk type with a weight of 30%, which is a notable percentage. Note that the equipment is deployed outdoors, in extreme cold temperatures. Usually, the weather can interact with the component through four parameters, represented by cold, heat, humidity, wind, and lightning. For our project, heat and lightning are considered negligible. The heat part is obvious since the environment itself is described by a rather negative temperature (maximum 10 ºC) since it is an avalanche prediction module). The lightning damaging the equipment has a very low probability, especially in our country, so it will not be taken in consideration when calculating the risk. To get a thundersnow in the wintertime is challenging because it is very uncommon for pockets of air to rise vertically, unless it is summer. One of the main risk factors that can damage our project is represented by the low temperatures and their effects on electronics. The studies show that the damage starts at - 20 ºC which is common for the chosen environment where the avalanche prediction module will be deployed. Manufactures usually recommend optimal temperatures for operating and storing equipment. Apart of protective cases, this technical documentation should be consulted when choosing the components. This cold damage can be represented

by hardware lubricants that are freezing[6], battery lifespan issues and condensation as a direct result of fast changing temperatures [6]

Based on this analysis, we will grade the 'Humidity' with **0.7**, 'Cold temperatures' with **0.8** and 'Lightning' with **0.1.**

Condensation is another important factor for cold electronics in areas with high humidity. The condensation and the collected droplets can cause consequences as bad as electrical shorts on circuitry. Usually, the circuits should be protected from humidity and we should also think about drastic changes in temperature. The wind part is debatable because the wind itself does not damage the equipment, but the debris do. Even more, the snow carried by the wind can affect the acquired data, because of snowdrifts. Wind pilling up snow on a reading module can affect the integrity of data.

The mechanical damage caused by the wind is covered in the 'Mechanical' risk type. To begin with, in conditions of heavy snowing, wind can carry the fresh snow and pile it against solid surfaces. This phenomenon is called snowdrift and can have a huge impact related to the accuracy of the sensed data. For our project, this risk has a high probability of happening because the sensors can be fully or partially covered by snow, as we can see in Fig 3.

Fig. 3. Example of IoT Acquisition module [7] and snowdrifting

As we can see above, if one of the rod surfaces is covered by snow (the sensors are graphically represented using bold white lines), then the acquired data of that surface is compromised. Having a high probability of happening, we will grade this risk type with **0.8**.

We have also some rather negligible parameters that are represented by 'Dust' and 'Falling objects. There is no need in considering dust as a factor when calculating the risk for our project since the hardware is deployed in winter

[6] This is most common for hardware equipment that has moving parts, such as hard disks

[7] This acquisition module was built Arduino and has five pressure sensors and one distance sensor. The acquired data was collected for us to calculate the difference in snow level density, differences that are leading to avalanche

conditions and dust cannot be found in the debris carried by the wind. Related to the 'Falling objects' we will grade this risk type with **0.1** being an event with low incidence (**< 0.3**)

The following factor is represented by landslides or even by the avalanche itself. This mechanical factor also has a high incidence of appearing, being the main event that is monitored by our project. There is usually a trifecta that lead to an avalanche, a mix of terrain, snowpack, and weather conditions. Based on this, we will rate this risk type with **0.7**. The risk is slightly lower than the snowdrift one, because the main purpose of the project is to predict avalanches, so we can take specific decisions to even prevent them (such as controlled avalanches, detailed reports that keep people for venturing in high risk areas). Apart of this, the human factor is also very important. Humans trigger 90% of the avalanche disasters, with as many as 40 deaths in North America each year.

The 'Human interaction' risk type, still part of the 'Environment'[8] is split in two major factors. One is represented both by the accidental data corruption and hardware damage, which are caused by clueless venturing in data acquisition areas. Secondly, we also have the physical data tampering part[9] which is a malicious act of corrupting the acquired information, by interacting solely with the environment [9]. Physical security is often overlooked when it comes to information security, because most companies focus on technology-oriented security countermeasures. Each of these factors should be graded based on their probability of incidence. Since this project is deployed outdoors in a public place, despite the perimeter security, the risk of sabotage, vandalism, theft is high, and we will grade it with **0.6** [10]. The accidental data corruption will be graded with a lesser score because modules will be deployed within a secure perimeter that is delimited by fences. Therefore, the grade is somewhere around the **0.2** value, because it has a low probability of incidence [11]. The same secure perimeter determines a low grade of **0.2** also for the data corruption caused by animals. This risk parameter is found under "Wildlife Interaction" risk type from *Table 1*.

The last risk type is represented by 'Radiation', but it is not applicable for our project since it will be deployed in a radiation free environment. As we can see from the same *Table 1*, this risk factor is present only in some niche IoT areas such as smart metering, industrial control, and emergencies (depending on the nature of the emergency itself). Because of this, the risk will be mitigated under the 'Unknown' buffer of 10%, as we can see in Fig 4.

[8] Note that we have a 'Human interaction' risk type also in the 'Security' category, not just in 'Environment'

[9] The physical data tampering is represented by unauthorized physical access to the acquisition areas with the purpose to sabotage the modules. This is different than software data tampering, which is performed through unauthorized channels, using hacking tools.

After performing research activities for grading each risk, we must compute all the information, using a risk formula. Before that, it will be easy for us to centralize all the grades in a suggestive diagram, using the colors code described in Fig 2.

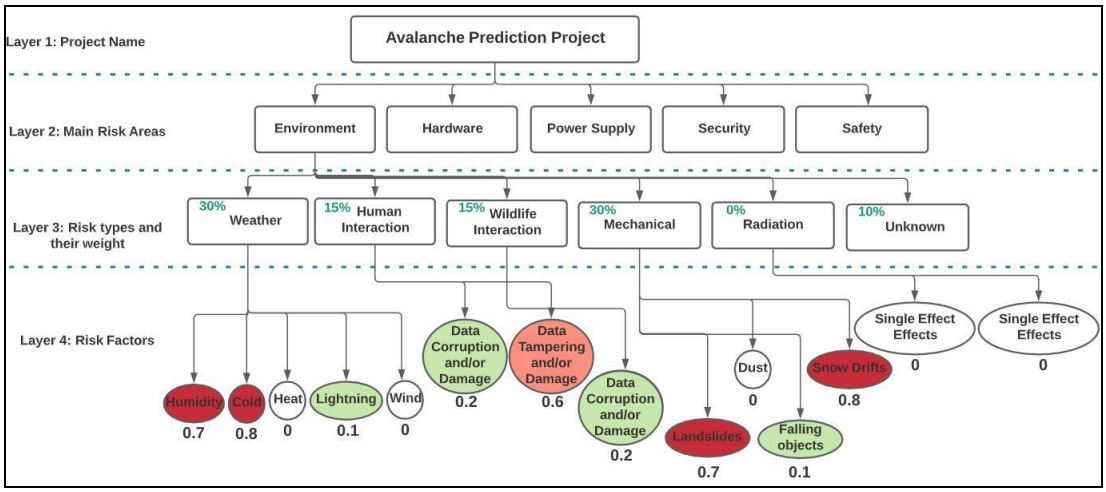

Fig. 4. Risk Diagram for Avalanche Prediction Project– Environment

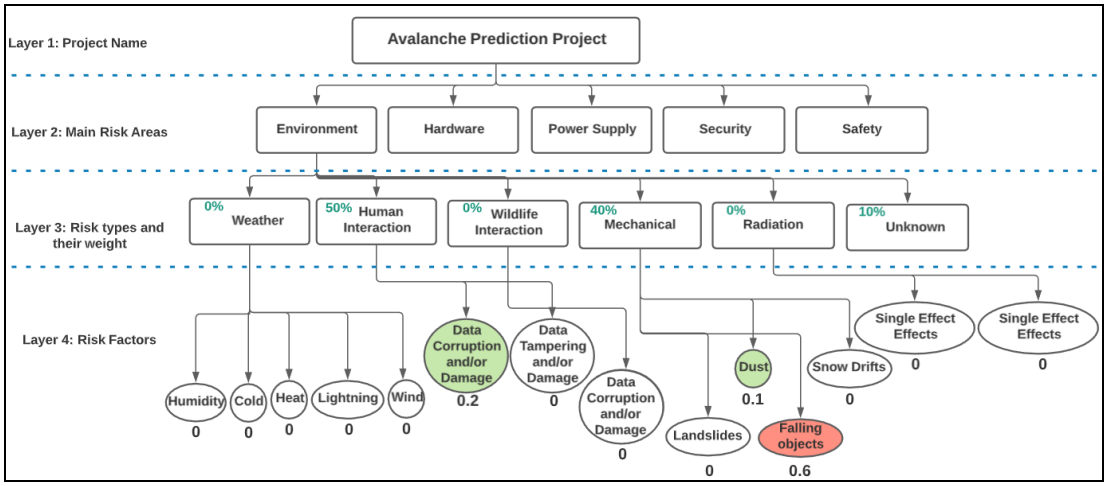

Fig. 5. Risk Diagram for Home Automation Project – Environment

The structure is the same regardless of the project, which is a notable improvement since all the existing frameworks are custom built around specific areas [12]. The only difference is represented by the grades. I chose a home automation project for Fig 5. because we have a reduced number of risk factors for this project. There is a very notable difference between a slope covered in a snow and the almost risk-free conditions provided by a domestic house/ apartment etc. After performing the research needed for grading each risk factor, the next

step is calculating the risk for our project. Let us see how these grades affect the 'Environment' main risk area.

First, we must think how the risk is defined. The strength of an application lays in the strength of the weakest link - it places an emphasis on the concept of a single point of failure [13]. Based on this, we have the following formula:

$$R = max\{r_{ij}: 1 \le i \le n, j \in c_i\} \quad (2)$$

where:

- r_{ij} is risk score of the *j*th component from the *i*th category
  ex: 'Cold' is a component from the 'Weather' category
- c_i is a set of components in each category
  ex: 'Cold', 'Heat', 'Lightning', 'Wind', 'Humidity'
- n is the number of categories (all the columns form Table 1, excepting the unknown part)
  ex: 'Weather', 'Human interaction', 'Radiation' etc.

After this the *U* variable for the unknown part, the total risk (*T*) formula becomes:

$$T = min\{1, R + U\} \quad (3)$$

where T (total risk) is capped at 1, which means 100% compromised [3]

Even if we know the total risk of an application, it is important to know how many vulnerabilities we have in our product, so we can minimize the changes of it being compromised. Based on this, an additional formula is needed to calculate the vulnerability coverage. This formula is based on various scenarios used to calculate the grades for each factor and it is like the unit testing process performed by developers before committing the code [14].

$$r_i = 1/n \sum_{i=1}^{n} c_i \quad (4)$$

where:

- r_i =the risk for each category ($r_{weather}$, $r_{mechanical}$, $r_{radiation}$ etc)
- n = the number of subcategories ('Cold', 'Heat', 'Data Tampering', 'Landslides')
- c_i = category ('Weather', 'Mechanical', 'Wildlife')

$$C = 1 - \sum_{i=1}^{n} r_i * w_i \quad (5)$$

where:

- C = risk coverage of the project
- w_i = the weight of each category from main risk area

The next step is to apply the formulas above for our case studies (both the avalanche prediction module and the home automation project). We chose projects from very different environments, for us to understand the applicability.

After using the scores assigned in Fig 4. and Fig 5., we have the following values:
R – maximum risk of compromise
U – unknown factor that should be present in all risk formulas (10%)
T – total risk of compromise
C – risk coverage
Avalanche Prediction Project 'Environment' Risk Area– Fig 4.
R = max {risk factors} = 0.8
U = 0.1
T = min {1, R+ U} = min {1, 0.8 + 0.1} = 0.9, so **90%**
C = 1 - [(1.6/5) *30 + (0.8/2) *15 + 0.2 *15 + (1.6/4) *30 + 0 + 10]/100 = **59.4%**
Home Automation Project 'Environment' Risk Area – Fig 5.
R = max {risk factors} = 0.6
U = 0.1
T = min {1, R+ U} = min {1, 0.6 + 0.1} = 0.7, so **70%**
C = 1 - [0 + (0.2/2) *50 + 0 + (0.7/4) *40 + 0 + 10]/100 = **78%**

3. Conclusions

First, the total risk value should be closer to 0 for an IoT project. The unknown parameter taken into consideration when using our formula means that we do not have a risk-free application. Secondly, the application risk coverage should be closer to 100, which is an ideal value achieved when we do not have any unknown factor in our application.

Practically, the product owners will know about the potential risk factors before starting the actual implementation. Discovering both flaws and risks earlier in the software development life cycle leads to a more cost and resource effective project.This approach can help the estimation activities, especially when using the latest delivery trends, such as Agile Scrum with its numerous planning meetings.

Besides environment, note that we have four more additional areas, represented by 'Hardware', 'Power Supply', 'Safety' and 'Security'. Each one of these have their specific risk factors and weights leading to a more complex overall formula. As future implementation, research activities will be performed to fine tune the risk formulas and also deliver detailed reports within an intuitive and user friendly interface There is a lot of potential in applying the whole risk assessment process, because we take in consideration most of the main risk areas for a project. Note that the existing risk assessment frameworks are built mainly

around the security part, which is indeed an important risk domain, but many improvements can be performed in other notable areas such as power management and physical safety. Applying this assessment process for each of the main risk areas can help us understand what we should implement and/or buy to minimize risk. Based on the values of the risk factors contained in the main area, we can create customized reports with targeted recommendations for each specific project, so one of its advantages are represented by a high level of applicability.

R E F E R E N C E S

[1]. *Mohammad Ammad-Udidin, Imran Baig, El-Hadi M.Aggoune, Muhammad Ayaz,* Wireless Sensor's Civil Applications, Prototypes, and Future Integration Possibilities: A Review, IEEE Sensors Journal, vol.18, pp 4-30, 2017

[2]. *Montbel Thibaud, Huihui Chi, Wei Zhou, Selwyn Piramuthu,* Internet of Things (IoT) in high-risk Environment, Health and Safety (EHS) industries, vol 108, pp 79-95, 2018

[3]. *David Levitsky,* "Assessing Security Risk in IoT Devices", 2018, https://digitalcommons.calpoly.edu/theses/1954, retrieved on 15th November 2020,

[4]. *Colin Tankard,* The security issues of the Internet of Things, Computer Fraud and Security, vol 2015, pp 11-14, 2015

[5]. *Hanny F. Atlam, Ezz El-Din Hemdan, Ahmed Alenezi,* Security Development Lifecycle, Internet of Things Forensics: A review, Internet of Things, vol 11, pp 1-41, 2020

[6]. *Jackie Davis,* "Electronics in the cold", 2011, https://cottagelife.com/design-diy/electronics-in-the-cold/, retrieved on 16th November 2020,

[7]. *Jens Verbeeck,* "Protecting electronics from the effects of radiation", 2017 https://www.neimagazine.com/features/ retrieved on 16th November 2020

[8]. *Ioannis Stellios, Panayiotis Kotzanikolaou, Mihalis Psarakis,* Advanced Persistent Threats and Zero-Day exploits in Industrial Internet of Things, Security and Privacy Trends in the Industrial Internet of Things, pp 47-68, 2019

[9]. *Sachin Babar, Parikshit Mahalle, Antonietta Stango, Neeli Prasad, Ramjee Prasad,* Proposed Security Model and Threat Taxonomy for the Internet of Things, CNSA 2010, Recent Trends in Network Security and Applications, pp 420-429, 2010

[10]. *Lawrence Fennelly,* Effective Physical Security, Editura Elsevier Butterworth, UK, 2004

[11]. *M. P. Ross, H. Lin, T. Mack,* "Designing of a Physical Security Perimeter Fencing System", IEEE International Carnahan Conference on Security Technology, pp. 205-210, 2010

[12]. *Y.P. Tsang, K.L. Chong, C. H. Wu, P. S. Koo, G. T. S. Ho, C.H.Y. Lam,* An Internet of Things (IoT)-based risk monitoring system for managing cold supply chain risks, Industrial Management and Data Systems, vol. 118, no. 7, pp 1432-1462, 2018

[13]. *Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Haslem, Faiz Alotaibi,* Internet of Things security: A survey, Journal of Network and Computer Applications, Volume 88, pp 10-28, 2017

[14]. *Dorothy Graham,* Foundation of Software Testing, Editura Cengage, USA, 2006.