# Requirement Specification

[INCOSE] describes a system life cycle for processes and activities. In the Emergency call button system, we have chosen to describe some of the activities in the life cycle model management process, which is a part of the organizational project enabling processes. The activities and scope for this project is illustrated below. In the life cycle model management, we have focused on one task in the project process, namely risk management. In the technical process our first priority has been the architectural design process, and seconds the implementation face, to support simulation of our architecture. The stakeholder requirement definition, and requirement analysis process have been melted together in one title (requirement specification). This decision is because our customer is an experienced system engineer.

|  |
| --- |
|  |

System Life cycle Processes Overview[[1]](#footnote-1)

### Concept of Operation

This section will contain a storytelling from where many use cases stakeholdes can be derived.

### Storytelling

An old lady has fallen and knocked her-self in the bedroom. The old lady presses emergency call button and gets redirected to a service assistant at the help care center. At the help care center, they quickly realize that the old lady needs help, and immediately sends a car off to get her.  
  
An old man tried to reach the coffee on top shelf, but cannot reach it. He then presses the emergency call and gets redirected to a service assistant at the help care center. The service assistant quickly realizes that it is not seriously and urgent help is not needed.

Storytelling helps people understand what the system is all about. It focuses on real life scenarios and how the system interacts in people’s daily life. In our case the system is designed to improve the helper’s workflow, by providing two way communications, which is new in this branch. The last story is an example of the strength of our system.

### Use case context diagram

In Figure 1 all communication goes through the Emergency call base, yet where the communication is merely relayed to another actor the Emergency call base is not shown as part of the communication. Figure 1 below shows stakeholders and use cases within emergency call button.



Figure 1 - Use case diagram

In Figure 4 may be seen a simplified activity diagram of what happens when a user pushes the emergency call button on a charged and configured emergency. What is left out is e.g. the technical mechanism for cancelling the emergency (e.g. in-stream messaging).

|  |
| --- |
|  |

Figur 2, activate state

## Stakeholders

Techinical

Commune

Help care center

The older

System administrators

## Requirement verification tracability matrix

To keep track of oure requirement we have developed a requirement verification tracability matrix (RVTM)[[2]](#footnote-2), though we have shipped the “v” verification because it is beyond the scope for this project.

The table below illustrates all system requirement together with is corresponding use case, and an indication whether it is a functional requirement or non functional.

### Initial Requirement Traceability Matrix

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Stakeholder Requirement** | **System Requirement** | **Requirement type** | **Component** | **Requirement ref.** | **UC Ref.** | **Test Case Ref.** | **Comment** |
| **1** | **SR1** | It shall be possible to activate the emergency call button | Functional | Handheld device | N/A | **UC1** | N/A | N/A |
| **2** | **SR2** | Is shall be possible to recharge the battery on the emergency call system | Functional | Handheld device | N/A | **UC2** | N/A | N/A |
| **3** | **SR3** | The system shall automatically adjust the RF transmission power according to the environment, to minimize power consumption | Functional | Handheld device | N/A | **UC3** | N/A | N/A |
| **4** | **SR4** | The system shall notify the user if the batty is below 20 % of max capacity | Functional | Handheld device | N/A | **UC4** | N/A | N/A |
| **5** | **SR5** | The system base shall notify a technician if is doesn’t receive signal for a period of 30 minutes | Functional | Base | N/A | **UC5** | N/A | N/A |
| **6** | **SR6** | It shall be possible to install a new emergency call button | Functional | Handheld device | N/A | **UC6** | N/A | N/A |
| **7** | **SR7** | It shall be possible to update the firmware on the emergency call button | Functional | Handheld device | N/A | **UC7** | N/A | N/A |
| **8** | **SR8** | The emergency call button itself shall not weigh more than 125g | Design | Handheld device | N/A | **N/A** | N/A | N/A |
| **9** | **SR9** | The emergency call button shall not be larger than 40x60x15mm | Design | Handheld device | N/A | **N/A** | N/A | N/A |
| **10** | **SR10** | The button on the emergency call button must be at least 20x30mm or have a circumference of at least 75mm | Design | Handheld device | N/A | **N/A** | N/A | N/A |
| **11** | **SR11** | The ISM band used shall be the EU allocated frequency for social alarms (EN 300 220) at 869.2 – 869.25MHz | Design | Handheld device | N/A | **N/A** | N/A | N/A |
| **12** | **SR12** | All requirements set down by the EU and Denmark regarding EMC, transmission strength and frequency hopping must be met, as well as other legal obligations pertinent to the product/project. | Design | Handheld device | N/A | **N/A** | N/A | N/A |
| **13** | **SR13** | The devices battery life shall be sufficient for at least 24 hours of stand-by (with heart beats) and a 5 minutes conversation | Performance |  | N/A | **UC4** | N/A | \* |
| **14** | **SR14** | The emergency call button shall transmit a button-push to the emergency call base within 500ms of the button being pushed | Performance |  | N/A | **UC1** | N/A | \*\* |
| **15** | **SR15** | An emergency button failure must be reported by the emergency button base no later than 2 hours after the actual time of failure. | Performance |  | N/A | **UC5** | N/A | N/A |
| **16** | **SR16** | A firmware update shall take no more than 30 minutes to complete | Performance |  | N/A | **UC7** | N/A | N/A |
| **17** | **SR17** | The emergency call button shall be able to charge from empty to fully charged in no more than 6 hours | Performance |  | N/A | **UC2** | N/A | \*\*\* |

\* A graph showing the dimensions/weight vs. battery life shall be shown starting at the smallest battery supporting the above time constraint and continuing until the size and/or weight constraint is reached.

\*\* The caregiver must be at the home of the user no more than 30 minutes from the activation of the emergency call. From this a requirement of a maximum delay from emergency call button is pressed until the alarm is received by central office of 10 seconds is created, and as the possible delay on the GSM network covers most of the 10 seconds allotted, the ISM delay of 500ms is determined.

\*\*\* If the user is to charge the emergency button themselves they must be able to do so while they sleep.

# Risk Management

This process is about identify, analyze, treat and monitor the risk continuously. Since the emergency call system, has great prospective it’s important to analyze which factor that could risk the product value on the marked.

**THIS IS NOT COMPLETE RISK MANAGEMENT ANAYLYSE**

The risk of failure for the emergency call system can be derived into different types of risk [INCOSE]

1. Development (*Technical risk* )
2. Maintenance (*Technical risk* )
3. Operation of system ( *Technical risk* )
4. Price (*Cost risk*)
5. Time to market ( *Schedule risk* )

Risk is dealing with uncertainty that is present throughout the entire system life cycle. The goal is to archive a proper balance between risk and opportunity.

## Technical risk

This risk is worth taking serious, because at technical failure could be devastating for emergency call system. The technical risk is the possibility that a technical requirement for emergency call system may not be archived in the system life cycle. Normally large technology companies have experienced from previous project, and know where the risk of failure is most likely, but for emergency call system which is a newly establish company, there are many technology barriers, which have to be confronted.

Some requirements are easily tested then others, especially *SR3* which take the environment into account are candidate for several tests, and would be rated HIGH in the risk probability table [[3]](#footnote-3)definition explained later

The risk of choosing an unknown platform is also worth mentioning. A Perato optimal solution is based upon choosing the right components in respect to cost and performance. These parameter isn’t enough when talking about technical risk. If the pareto optimal solution is a combination which include a FPGA and there wasn’t any IP available for a given functionality, we could be risking a lot because no members in the project team is familiar with HDL implementations. The pareto optimal solution presupposes that each configuration is well known for the project team.

It is nevertheless important to consider a platform risk which can be given a probability as illustrated in the table below

|  |  |  |
| --- | --- | --- |
| **Probability** | **Uncertainty** | **Rank** |
| **> 80%** | Almost certainly, highly likely | 5 |
| **61%-80%** | Probable, likely, probably, we believe | 4 |
| **41%-60%** | We doubt, improbable, better than even | 3 |
| **21%-40%** | Unlikely, probably not | 2 |
| **< 21%** | Highly unlikely, chances are slight | 1 |

Tabel

The above table illustrates how likely a future problem can occur. In the rationale about recommended platform, in respect to pareto point, and probability rank from the above table will be included to give a mere adequate conclusion about platform selection.

We have been discussing a probability factor, which is importer to consider. But what is the consequence if such, if the worst happens. Let’s consider an example.

Let say we choose the FPGA platform. Probability, that the software development exceed time schedule due inexperienced developers would get a rank 4. The consequence would in worst case be lost customers and lawsuits due breach of contract.

## Cost risk

This risk is off cause very important for companies producing large number. While our product is a school project, price of component isn’t very important, though it will be considered when making recommendation for platform selection. If this product is matured for production, the component price is off cause important, and platform selection will be based not only upon optimal solution but also cost price.

## Schedule risk

What if a supplier can deliver before half a year, or what is we can’t implement the system in the given schedule. This normally a hard topic because is involves estimation of task, and milestones. This project schedule has been tackled through small deliveries from each team member to each meeting.

### Avoiding risk

To avoid risk, system engineers have to be proactive in early requirement phase. As stakeholders requirements normally change within a project life cycle, the risk management is an iterative process, and must be maintained all time. If system engineers rank stakeholders requirement in early system phase the concerned components involved could be candidate for early test and verification. The illustration below shows the iterative process of risk management throughout the project life cycle. It is not enough just to identify risks, they also have to be prioritized in order, and for each risk a strategy has to be made.

|  |
| --- |
|  |

Figur 2, Risk management iterative process

The development of Emergency call button involves risk. Some of those risks are off cause more serious than other. Prioritization, and strategy plan of those risks is therefore necessary in the early project cycle.

The table below illustration some off the identified risk in Emergency call system. One could also imagine an updated version of the RVTM, with a reference to a risk id. This way system engineering would have to consider risk as part of each stakeholder requirement. Also system managers would know were extra effort had to be assigned. The project group have made a custom risk schema, below which includes some of the risk in Emergency call button. The table doesn’t includes all possible risk in Emergency call button, but only a few major risks.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Emergency call button risks** | | | | |  |
| **Risk type** | Id | Requirement/  Description | Rank | Strategy | Consequence |
| **Technical** | 1 | SR13 | 3 | **Action:**   * Test power consumption on demo-board   **Resources:**   * 500 hours | Customer choose another supplier  Safety risk |
| 2 | Unknown platform ect. FPGA | 5 | **Action:**   * Training, and workshop   **Resources:**   * 1000 + hours | Schedule |
| 3 | SR14 | 3 | **Action:**   * Test in different environmental condition * Retry counter * Alert user (beep)   **Resources:**   * 300 + hours | Safety risk |
| **Cost** | 4 | SR8 | 2 | **Action:**   * Investigate battery marked | Customer choose another supplier |
| **Schedule** | 6 | Id ref 2 |  |  |  |

It appears that the most risky part in developing the Emergency call button is choosing an unknown platform. The reasoning for choosing another platform is therefore very important, and such platform must provide superior advantages. Chapter JVH pareto point will show if an unknown platform provides those sufficient advantages, and what the tradeoff will be in respect to this risk management analyses.

1. INCOSE Figure 1-1 System life-cycle Process Overview per ISO/IEC 15288:2008 [↑](#footnote-ref-1)
2. INCOSE p. 57 [↑](#footnote-ref-2)
3. [http://www.dtic.mil/ndia/2004cmmi/CMMIT1Mon/Track1IntrotoSystemsEngineering/KISE09RiskManagementv2.pdf] [↑](#footnote-ref-3)