Rijksuniversiteit Groningen

Software Architecture

Group 3

Smart Flood Monitoring

Authors:

Eedema, Gerrit
Putra, Guntur
Fakambi, Aurélie
Schaefers, Joris
Klinkenberg, Guus
Brandsma, Jeroen
Menninga, Wouter

Version: 0.1
06 Sept, 2015

Authors

Name	E-Mail
Eedema, Gerrit	G.Eedema@student.rug.nl
Putra, Guntur	G.D.Putra@student.rug.nl
Fakambi, Aurlie	A.Fakambi@student.rug.nl
Schaefers, Joris	J.Schaefers@student.rug.nl
Klinkenberg, Guus	G.Klinkenberg@student.rug.nl
Brandsma, Jeroen	J.T.Brandsma@student.rug.nl
Menninga, Wouter	W.G.Menninga@student.rug.nl

Revision History

Version	Author	Date	Description

Contents

1 System Context

This section shows the context of the system.

1.1 Context

We are the architecture team in a company that specializes in making smart systems for environmental monitoring.

In many parts of Europe, natural disasters cause allot of trouble. These disasters can result in catastrophic events that can take human lives and where it will take a huge amount of money to repair the damage caused.

Floods, for example, of rivers and lakes destroy huge amounts of farmlands and sometimes even causes deaths. Climate change and extreme weather phenomena cause these floods to get worse every year. This makes it hard to timely anticipate on these upcoming floods in order to control the damage it will do. There's a huge need for a system that will help the people during natural disasters like these, which is what this system aims to do.

The goal of our first product is to reduce the damage and problems caused by floods by trying to predict upcoming floods as correctly as possible. By using various kinds of sensors, vehicles and control units, this system monitors the areas and gives warnings about upcoming floods. It then also provides guidance to the people in the (about to be) flooded area to reduce the nuisance the flood causes.

When this flood monitoring system works as planned and shows that it can indeed reduce the damage of floods, new products will be made by our company that will focus on reducing the damage of other kinds of natural disasters.

Over time, the system will try to reduce the amounts of human lives that are lost by natural disasters, starting with floods, and will try to reduce the social and financial consequences. This will revolutionize the way we think about natural disasters. Though the market around environmental monitoring is very competitive, this system aims at being a core part of future monitoring systems by being as dynamic and flexible as possible. Thereby allowing new features to be added easily, allowing the system to grow over time, including new upcoming user needs.

2 Architectural business information

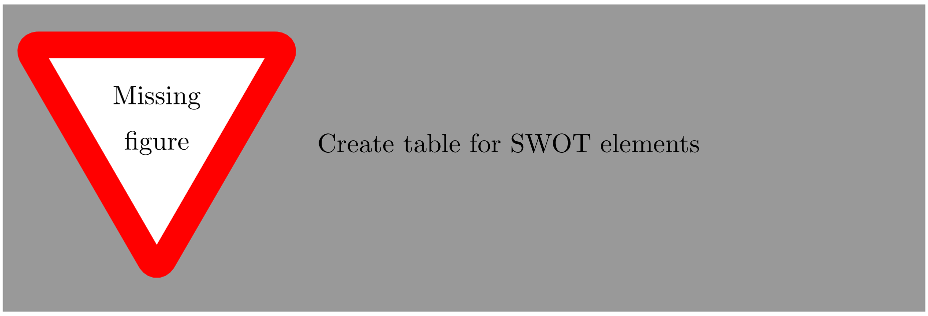
The following section describes the different aspects of the business environment of the Smart Flood Monitor. First we will explain our vision and why there is place for us at the market. After this the product and its customers will be explained. This chapter is completed with a more detailed look at the business model and some models about the market and the financial prospect.

2.1 Business vision

The Smart Flood Monitor will cause a revolutionary innovation on the environmental monitoring market. RugSAG3 will offer a system which can detect floods early and correctly. This system helps us to enforce our vision, to limit the social and financial consequences of floods and avoid the loss of human lives.

RugSAG3 will not be the only competitor on the environmental monitoring market. This means RugSAG3 will have to take their own strength and weaknesses into account. If we combine these qualities with an analysis of the opportunities and weaknesses on the market, we should be able to become a core part of the future environmental monitoring systems.

Such an analysis is called a SWOT-analysis.



Strengths: Quality of the product.
Weaknesses: [No experience?]
Opportunities: Due to climate change, the market will grow.
Threats: New competitors will enter the market.

Our unique selling point is to provide the product with best quality, combined with the best service that is available.

2.2 Business rationale

2.3 Product and service description

2.4 Target audience

2.5 Business model

2.6 Roadmaps

2.7 Financial model

2.8 Competitors

3 Requirements

3.1 Architectural vision

3.2 Stakeholder and their concerns

3.3 Stories and use-cases

3.4 Functional requirements

3.5 Commercial non-functional requirements

3.6 Technical non-functional requirements

In this section, the technical non-functional requirements important to this system are discussed.

3.6.1 Resilience

The system will have many connected sensors, which can have failures. The system should be able to recognize such failures timely and recover from them without the QoS or the functionality of the system being affected.

The system should be able to continue functioning with the same QoS in a situation where up to 5% of the sensors suffer from failures.

3.6.2 Interoperability

The system has dependencies on third-party systems. For example, to make predictions about the development of waterlevel, the system will need to retrieve information from water forecasting services.

Not only for input, but also for output, the system will need to interoperate with third-party systems. If the system has registered a risk of a flood, it should interact with systems of emergency services and other authorities to alert them and sent relevant information.

3.7 Evolution requirements

3.8 Risk assessment

4 Analysis

Assumptions:

- People requesting guidance have access to a phone that is either connected to a mobile network or to the internet
- The system is able to monitor various kinds of sensors
- The system has access to the internet
- The sensors are (in some way) connected to the system
- The system is able to use various kinds of sensors in order to predict floods.
- When the system has enough proof of a flood, the users will be informed in a timely manner (within 30 seconds maximal).
- People within a range of (30km?) will be warned about a detected flood
- The system uses multiple weather forecasting services to make its own predictions about possible upcoming floods.

In what ways will the system detect a flood? Sensors, UAVs, forecasting?
How to detect a flood based on the forecasting? Warn beforehand? Pay extra attention to the new upcoming flood by sending extra robots/uav's?

Decision: Security? (What if someone tries to manipulate a sensor?)
One or multiple warning systems (failover)
Networking? Redundancy of cables (trunked?)
When to alarm? What to do if only 1 sensor warns about a flood?
Security decisions?

Operating system decision
Storage decision
Push or pull updates? (Pushing Im hoping)

Name	**Linux**
Decision	**1**
Status	**New**
Decision	
Alternatives	
Arguments	

5 System Architecture

6 Hardware Architecture

7 Software Architecture

8 Architecture evaluation

9 System evolution

A Time Tracking

A.1 Week 1

Person	**Task**	Hours
Gerrit		
Putra		
Fakambi		
Schaefers		
Klinkenberg		
Brandsma		
Menninga		