# Risk Analysis

## 1.1 Version

| Version | Date | Change | Author |
|---------|------|--------|--------|
| 0.1 | 20.09.2014 | Setup document | JR |
| 0.2 | 28.09.2014 | Add risks, evaluation, consequences | JR |
| 1.0 | 05.10.2014 | Grammar, layout | JR |

## 1.2 Introduction

### 1.2.1 Purpose

This document evaluates and calculates all possible risks and defines actions that can minimize these risks as well as possible.

## 1.3 Risk evaluation

### 1.3.1 Unclear requirements

The start of a project is normally no easy task because its requirements are vaguely known. If they are not well defined as soon as possible, the requirements will stay vague through out the whole project, which can lead to a disaster.

### 1.3.2 New technologies

The new technologies which are present in this project are the following:

- Gradle

- Travis CI

- Python

Each of them brings its own risk.

### 1.3.3 Integration Boilerpipe

The Boilerplate algorithm needs to be integrated into the text extraction framework. Every interface of an external component is a possible risk factor.

### 1.3.4 Integration Justext

The Justext algorithm needs to be integrated into the text extraction framework. Every interface of an external component is a possible risk factor.

### 1.3.5 Implementation RSS algorithm

The development and implementation of a new algorithm is predestined to generate risks.

## 1.4 Assessment of risks

| Risk | Impact | Probability of occurrence | Risk factor |
|---|---|---|---|
| Unclear requirements | 2 | 4 | 8 |
| New technologies | 3 | 3 | 9 |
| Integration Boilerpipe | 5 | 1 | 5 |
| Integration Justext | 5 | 5 | 20 |
| Implementation RSS algorithm | 1 | 5 | 5 |

## 1.5 Consequences

### 1.5.1 Unclear requirements

As I am working with the client each and every day, it is very easy prevent misunderstandings with asking the client at once. Even though misunderstandings can occur between student and expert. To prevent this, it is necessary to have a document to define the requirements as soon and as exact as possible. This well be done in the form of the system requirement specification in the first mile stone. Possible ambiguities can be clarified at the first mile stone meeting.

### 1.5.2 New technologies

It is important to do prototyping with new technologies in the first phase of the project to eliminate these risks as soon as possible.

Gralde and Travic CI are needed in the first mile stone to set up the programming environment. So if there is any problem it will occur in a very early stage of the project and a possible solution can be found.

### 1.5.3 Integration Boilerpipe

This risk is rated much lower than the Justext interface because it's implementation is in Java and it provides a Java API. Never then less a prototype should be done as soon as possible to prevent any nasty surprises with the interface.

### 1.5.4 Integration Justext

This point is classified as the highest risk of all. This is because the implementation is in Python and it is not clarified yet how it will be integrated into the text extraction framework. An analysis of possible solution with prototypes needs to be done as soon as possible.

Possible solution are:

- jython (http://www.jython.org)

- Implementation in Java

- Java Processor Interface

### 1.5.5 Implementation RSS algorithm

This risk has a very high probability of occurrence because it is very likely that a development and an implementation of a new algorithm is going to cause problems. There is no real solution to that risk. But because of this requirement is nice to have, the impact on the outcome of the project is very low. Further more Patrik Lengacher, the tutor of this project, is very experienced in this subject area and will be able to help out if any problems occur.