Berner Fachhochschule
Bern University of Applied Sciences
Haute école spécialisée bernoise

Bachelor Thesis

BTI7321

# Project Management

## Logos Recognition for Webshop Services

*Author:* Noli Manzoni

*Supervisor:* Prof. Dr. Olivier Biberstein

*Expert:* Jean-Marie Leclerc

V 0.1

14.06.2018

# Contents

# List of Figures

# List of Tables

# 1. Introduction

## 1.1 Purpose of the document

This document describes all the project management aspects (objectives, requirements, etc.) of the Bachelor thesis *Logos Recognition for Webshop Services*. This stand-alone report is part of a bigger document. In fact, it is an attachment to the main documentation that describes the entire Bachelor thesis. This document is realised with LaTeX[1], MiKTeX [2] and TeXstudio [3]. The structure and style of this document are based on [4].

## 1.2 Project description

This Bachelor thesis aims to figure out if the presence of specific logos helps to understand the underlying activity of a website. To be more precise, this project should prove that the presence of logos such as Visa, PayPal, DHL, etc. shows that a website provides presumably services typical of webshops such as payment and delivery. This kind of information could be really useful for the identification of websites that sell counterfeit products, in business intelligence, etc. The presence of delivery and payment services in a counterfeit product vendor, raises considerably the danger of such a website and therefore it should be identified faster than the others.

In this Bachelor thesis an application must be developed. It must determine if a website contains certain logos typical of webshops. Logos could be stored either as single images or as parts of larger images. For this reason, the software should deal with both cases. To increase the efficiency of such a program in understanding if a website is a webshop, ~~other information could be used. This info could be extracted from keywords, JavaScript code, hyperlinks, etc., contained on a website~~ [**update 03.04.2018**] neural networks can be studied and used at a high level. The choice of the best technology to extract and classify information from logos must be based on comparison between different techniques. The choice must depend only on the results of the algorithms without any optimization. In fact, the objective is to create a simple application with any constraints in performances.

# 2. Objectives

The main objective of this Bachelor thesis, as seen in Section 1.2, is to create a proof of concept. It must show that the presence of specific logos in a website helps to understand its underlying activity. To be more precise, it must be composed of an efficient classifier which determines the presence or absence of logos within a given set of them. This classifier must be efficient in the sense that it must minimize certain types of errors. In fact, the different technologies and combination of them, should be evaluated with the values of the confusion matrix. Logos can be stored either as single images or as parts of larger figures. For simplicity in a first phase they are contained in a directory. In a second phase, they should be extracted from a given website automatically. Once these objectives are reached, ~~the classifier should be extended. It should use other pieces of information contained in a website such as keywords, hyperlinks, JavaScript code, etc. This extra information should be extracted automatically given the website's URL~~ [**update 03.04.2018**] approaches with neural networks can be studied and used to improve the application.

For more information, please refer to the official document *Definition of Bachelor Thesis Project* found in the attachments.

## 2.1 Stakeholders

Stakeholders for this Bachelor thesis are:

- **Advisor**: Prof. Dr. Olivier Biberstein
- **Expert**: Jean-Marie Leclerc

# 3. System boundaries

This logos' recognition system must have few but precise functionalities. It must be able, given a website's URL and an internet connection, to extract all the images contained in the website. In addition to this, the system ~~must analyse other information contained on the website such as keywords, hyperlinks, JavaScript code, etc.~~[**update 03.04.2018**] could use neural network to achieve a better classification. Once the system has all the information, it must compute the probability that the given website sells products. As surplus, the system must be able to say which services are present on the website. The different technologies tested in this project must not be studied in depth. To be more precise, any optimization must be performed on the different algorithms. However, if optimizations are found, they must be reported as possible improvement for a future project.

As this thesis uses an Agile development model called Scrum, this document describes the system boundaries with user stories and acceptance criteria. User stories are not the same as user cases [5]. In fact, the former is a short description of something the costumer wants to do with the application [5]. Contrary, the latter is a description of a set of interactions between a system and one or more actors [5]. Fortunately, user stories are not just single sentences, but they can go along with acceptance criteria. This additional information defines the boundaries and when a user story is completed. How can a team be able to implement something generic as a user story? "Writing use cases to flesh out user stories in Agile projects is certainly not unheard-of . . . . But it becomes clear as we move through the workshop that user stories are just the start of a process of understanding what the team is making. By the end of this process, you've covered off everything a use case would have told you, but in an organic manner" [5]. This citation answer the question above but it rises another. How the information can be received in an organic manner? In Scrum and in its sprint planning meeting, the product owner presents the user stories from the backlog. In this moment, the team asks questions to further clarify the user stories and the acceptance criteria. This step helps to understand if there are misunderstandings in the team. In this thesis, the product owner is partially the project advisor and partially me because I am able to make some choices. In other words, I can

decide how to implement the different components of the application. In addition to this, because a lot of technologies must be assessed, this project is more research-oriented.

## 3.1 User stories

This section describes the user stories and their acceptance criteria. For the description the classic template *As a [Actor] I want [action] so that [achievement]* is used .

### 3.1.1 Webshop probability

As a [User] I want [give a URL] so that [I can know the probability that the given website is a webshop].

#### Acceptance criteria

- Users cannot submit a non-existent or invalid URL

### 3.1.2 Web-shops probability

As a [User] I want [give a list of URLs] so that [I can know the probability that they are web-shops].

#### Acceptance criteria

- Users cannot submit non-existent or invalid URLs

### 3.1.3 Service offer

As a [User] I want [give a URL] so that [I can know which services (i.e. DHL, PayPal, etc.) are offered by the given website].

#### Acceptance criteria

- Users cannot submit a non-existent or invalid URL

### 3.1.4 Services offer

As a [User] I want [give a list of URLs] so that [I can know which services (i.e. DHL, PayPal, etc.) are offered by the given website].

#### Acceptance criteria

- Users cannot submit non-existent or invalid URLs

### 3.1.5 New logo type

As a [User] I want [add a new logo type] so that [I can improve the range of recognized logos].

**Acceptance criteria**

- Users cannot add an existing logo type

### 3.1.6 Logos addition

As a [User] I want [upload images of logos] so that [I can improve the recognition of a certain type of logo].

**Acceptance criteria**

- Users must chose an available logo type
- Users cannot upload logos of different types

### 3.1.7 Logos removal

As an [User] I want [delete logos] so that [I can lighten the load of the system].

**Acceptance criteria**

- Users can select multiple logos
- If there are active recognition processes they are stopped

### 3.1.8 Logo types removal

As an [User] I want [delete logo types] so that [I can lighten the load of the system].

**Acceptance criteria**

- Users can select multiple logo's types
- If there are active recognition processes they are stopped

### 3.1.9 System train

As an [User] I want [train the system] so that [it can use the new added logos/logos' types].

**Acceptance criteria**

- If there are active recognition processes they are stopped

# 4. Requirements

This chapter exposes all the requirements of this Bachelor thesis. Each requirement is identified by an ID, a status (draft, approved, released, outdated), a priority (must, optional) and a description. Before the list of all requirements (see Section 4.2 ff), Section 4.1 lists all the sources of requirements for this project.

## 4.1 Sources and procedures

### 4.1.1 Stakeholders

In this thesis, the majority of the requirements are decided by the advisor. The other stakeholders are here to share their experience and to give useful feedbacks for new features. Therefore, new requirements/features must be discussed first with the thesis advisor.

### 4.1.2 Documents

Requirements are mainly based on the official document *Definition of Bachelor Thesis Project* (see attachments). In addition to this, because of this Bachelor thesis is a continuation of the project *Logos Recognition* [6] a lot of information is derived from it.

### 4.1.3 Systems in operation

As seen in Section 4.1.2, this thesis is a follow-up of the project *Logos Recognition* [6] in which some prototypes of logos recognition were created. For this reason, the requirements are partially restricted from the results achieved in that project.

## 4.2 Functional requirements

Functional requirements describe what the system is supposed to do. This thesis has three main categories of functional requirements, i.e. logos recognition, information extraction and logos classification.

| ID | Status | Priority | Description |
|---|---|---|---|
| F1 Logos recognition | | | |
| F1.1 | Released | Must | Recognition of logos as single images |
| F1.2 | Released | Must | Recognition of logos included in larger images |
| F2 Information extraction | | | |
| F2.1 | Released | Must | Extract images from a website given its URL |
| F2.2 | Outdated | Optional | ~~Extract keywords from a website given its URL~~ [**update 03.04.2018**] |
| F2.3 | Outdated | Optional | ~~Extract hyperlinks from a website given its URL~~ [**update 03.04.2018**] |
| F2.4 | Outdated | Optional | ~~Extract JavaScript code from a website given its URL~~ [**update 03.04.2018**] |
| F3 Logos classification | | | |
| F3.1 | Released | Must | Classification of payment/delivery services logos such as Visa, DHL, etc. with classic techniques |
| F3.2 | Approved | Optional | Classification of payment/delivery services logos such as Visa, DHL, etc. with neural networks techniques [**update 03.04.2018**] |
| F3.3 | Released | Must | Computation of the probability that a website sells online products |
| F3.4 | Outdated | Optional | ~~Integrate other pieces of information to increase the efficiency of the classifier~~ [**update 03.04.2018**] |
| F3.5 | Released | Optional | Extent the classifier so that it can say if a website offer a specific delivery/payment service |

Table 4.1: Functional requirements

## 4.3 Technical requirements

Technical requirements describe how the system is built. This thesis has three main categories i.e. software architecture, Graphic User Interface (GUI) and documentation.

| ID | Status | Priority | Description |
|---|---|---|---|
| T1 Software architecture | | | |

| | | | |
|---|---|---|---|
| T1.1 | Released | Must | Choose the more suitable programming language for the project between Java, Scala and Python |
| T1.2 | Released | Must | Use an image recognition technique or a combination of them to recognize payment/delivery logos |
| T2 Graphic User Interface (GUI) | | | |
| T2.1 | Released | Must | Command-line application |
| T2.2 | Released | Optional | GUI application |
| T3 Documentation | | | |
| T3.1 | Released | Must | Use GitLab and GIT as a version control |
| T3.2 | Released | Must | Use Scrum as methodology framework |
| T3.2 | Released | Optional | Use LaTeX to write the documentation |

Table 4.2: Technical requirements

## 4.4 Quality requirements

Quality requirements describe quality criteria, factors and metrics for the project. This thesis has only one category of quality requirements i.e. confusion matrix errors.

| **ID** | **Status** | **Priority** | **Description** |
|---|---|---|---|
| Q1 Confusion matrix errors | | | |
| Q1.1 | Released | Optional | Minimize false positives |
| Q1.2 | Released | Optional | Maximize true positives |

Table 4.3: Quality requirements

# 5. Planning

## 5.1 Work packages

As seen from Section 4.3,this Bachelor thesis uses the Scrum methodology and GitLab for version control. These two products work perfectly together. First, GitLab makes available an issue Tracker for tracking the evolution of new ideas or the process of solving a problem [7]. Second, this tracker is perfect to use with the Scrum methodology. In fact, it gives the possibility to create boards of issues/working packages like a Scrum backlog. Furthermore, this system gives the possibility to estimate the workload of a task, to keep track of the real time spent and to assign labels. In addition to this, if a milestone is created for each sprint end, the entire Scrum methodology can be reconstructed without any effort. For the full reproduction of the Scrum's system, four different boards are created. In Figure 5.1 starting from the left there are: the Backlog, the current sprint tasks, the working in progress tasks and the completed tasks.

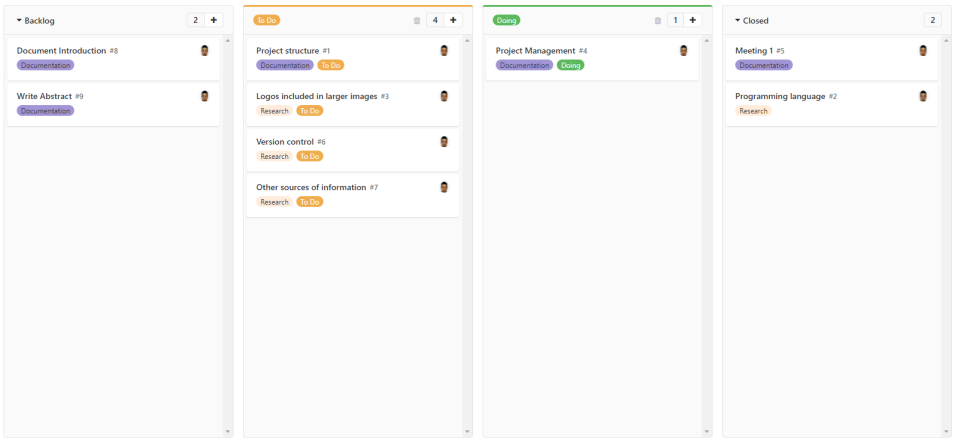


Figure 5.1: GitLab boards

Each of these boards contains multiple tasks with one or more labels (documentation, development, research, critical) and an estimation of the work load. To keep track of the real time spent for a task, each time a small part

is finished, it is committed. Then, in the task description, a comment with the real time spent and a link to the commit is added.

To overcome the problem of not be able to manage the great amount of time for this Bachelor thesis, I decided to use the Pomodoro technique as a time management method. This technique breaks down work into intervals of 25 minutes separated by short breaks. I chose this technique because it helped me when, during my apprenticeship, I struggled to work many hours on the same project. The exact working principles of this technique can be found at [8].

## 5.2 Sprints

For this project, as it is sixteen weeks long, there are four sprints of four weeks each. This choice allows to have enough time in each sprint to deliver, at the end of each of them, a functioning product. The second sprint is a week longer because of spring break (Easter holiday) where theoretically students do not work.

This section lists the different sprints and explains what are their objectives. This information can also be found under the issues' tracker on GitLab where a milestone period replace a sprint.

### 5.2.1 Sprint 1

**Interval:** 19.02.2018-18.03.2018

The objective of this sprint is to try to build prototypes of the mandatory requirements like images recognition, object recognition (logos in larger images) and images extraction. If possible, to choose the most efficient technology for the final product, different methods must be tried and evaluated. Along these steps, a documentation must be written to explain the choices, the development processes and the results of all prototypes. In addition to this, in this iteration the project management aspects and the introduction and abstract of this Bachelor thesis must be realised. The different tested technologies must not be explored in depth but only superficially. They must be used as they are implemented by the different libraries. To be more precise, any optimization must be performed. Moreover, the different prototypes must be developed as fast as possible and their explanation must stay at a high-level.

### 5.2.2 Sprint 2

**Interval:** 19.03.2018-22.04.2018

Since the analysis of the possibilities to classify logos required more time than estimated, almost the entire sprint 1 is shifted in sprint 2. In the sprint planning it has been estimated to use only one or two techniques for Logos' classification. At the end, five different technique were used and therefore the time needed to develop and to write the documentation is quadrupled. In this sprint, all the needed prototypes most be developed and the documentation must be written. These steps should allow to start the application development in the next sprint. To be more precise, in this iteration the documentation of the image recognition technologies, the prototypes and explanation of objects recognition and images extraction must be completed. In addition to this, the project management document must be finalized and the abstract and introduction of this thesis must be written. As the previous sprint, the different technologies must be explored superficially and the prototypes must be developed as fast as possible to stay within the time limit.

### 5.2.3 Sprint 3

**Interval:** 23.04.2018-20.05.2018

The different implementations have been completed but unfortunately, the results were not so good. For this reason, different enhancements have been applied and this additional step took a lot of time. Therefore, in this sprint the final enhancements results must be commented and described in the documentation. Once this is finished, the final application must be developed. At the same time, its design must be documented along with all the necessary diagrams. The final application must be developed as modular as possible so that in the future new algorithms or technologies can be added without problems. In addition to these things, the poster (A1 project description) and the book (A4 project description) must be created and completed because at the latest at the end of this sprint they must be consigned. As the previous sprint, the different technologies must be explored superficially and the prototypes must be developed as fast as possible to stay within the time limit.

### 5.2.4 Sprint 4

**Interval:** 21.05.2018-14.06.2018
In this sprint the book and poster must be finalized and delivered. Then, the GUI application must be developed and tested. Unfortunately, in the previous iteration the development of the application core and terminal interface have taken a lot of time and therefore they have not been documented. For this reason, in this sprint the different application must be described in the

documentation. Moreover, a small application manual must be realised so that the user knows exactly how to use the different parts and how to extend them. To finalize this thesis, the project management document must be updated with all the information about the tasks planning, commits frequency, etc. Once all the documents are completed, they must be reviewed so that the highest quality can be achieved.

## 5.3 Burn down chart

A burn down chart shows graphically how quickly the work is completed. This representation shows how much work is left versus the time. Burn down charts are used to track the completion of a milestone and therefore in this case the conclusion of a sprint. In fact, the vertical axis is the representation of the backlog (outstanding work) meanwhile the horizontal axis represents the time. This type of chart is very useful to predict when all the work will be completed. Furthermore, it is often used in Agile software development methodologies like Scrum. According to [9] burn down charts are great because they make the reality of the project clear, they show the impact of decisions, they warn early if things are not going according to plan, and they allow getting rid of all the wishful thinking around dates. As it have been already said, a burn down chart is available for every project milestone (a.k.a. sprint) and therefore they can be found in the GitLab repository under Project > Issues > Milestones.

## 5.4 Backlog

This section contains all the four sprints' work packages which are available in the GitLab repository. Each work package has an estimation, a workload and comments. A day (1d) is equals to eight hours and one week (1w) is equals to five days.

### 5.4.1 Sprint 1

This sprint estimation and workload is really low because a lot of tasks were not completed and have been taken to the second sprint.

| **Work package** | **Estimation** | **Workload** | **Comments** |
|---|---|---|---|
| Meeting 1 | 30m | 1h30m | Problem with latex bibliography |
| Meeting 2 | 1h | 50m | - |
| Meeting 3 | 1h | 50m | - |
| Meeting 4 | 1h | 1h15m | - |

| | | | |
|---|---|---|---|
| Other sources of information | 2h | 1h45 | - |
| Project Management | 1d2h | 1d5h15m | Problems with risks mitigation |
| Logos included in larger images | 4h | 2h | - |
| Programming language | 4h | 3h40m | - |
| Project structure | 4h | 3h30m | - |
| **Tot** | **3d3h30m** | **3d4h40m** | **-** |

Table 5.1: Work packages sprint 1

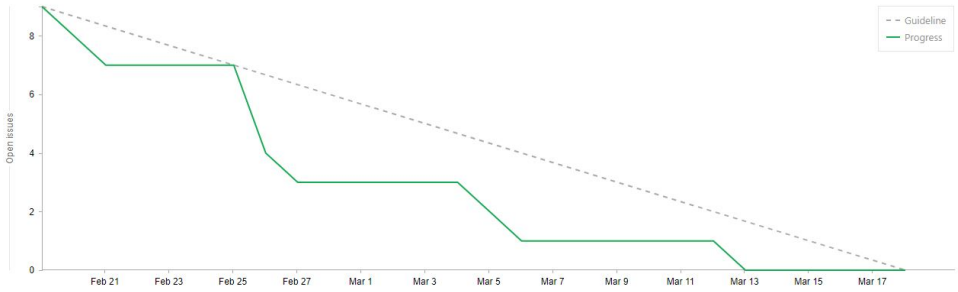


Figure 5.2: Sprint 1 burn down chart

### 5.4.2 Sprint 2

| **Work package** | **Estimation** | **Workload** | **Comments** |
|---|---|---|---|
| Meeting 6 | 2h | 2h40m | - |
| Meeting 7 | 2h | 2h | - |
| Extract images automatically | 6h | 7h | More complicated than previewed |
| Project Management Document Review | 3h | 4h20m | - |
| Logos in larger images | 3d | 4d2h20m | Problem with Selective Search (see documentation) |
| Logos recognition | 3d | 4d7h5m | Bag of Word realisation was complicated |

| | | | |
|---|---|---|---|
| Classifier | 1w | 1w1d2h30m | Problem with classifiers' implementation and documentation was really long |
| **Tot** | **2w3d3h** | **3w2d3h55m** | **-** |

Table 5.2: Work packages sprint 2

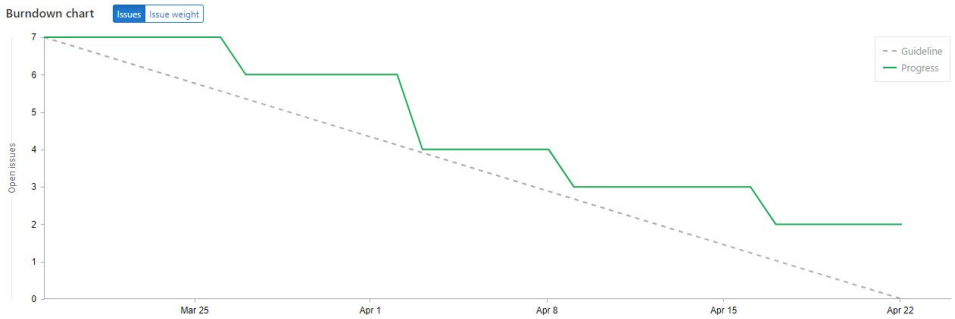


Figure 5.3: Sprint 2 burn down chart

### 5.4.3 Sprint 3

| Work package | Estimation | Workload | Comments |
|---|---|---|---|
| Default classifier implementations | 2d2h | 2d2h | - |
| Command Line application | 2d2h | 2d30m | - |
| Design application | 6h | 6h | - |
| Sprint 3 description | 2h | 1h | - |
| Weka introduction | 1d4h | 2h | It has been decided to use the time for other work packages |
| Document structure refactorisation | 4d | 4d1h | - |

| | | | |
|---|---|---|---|
| Objects recognition improvements | 2d2h | 2d4h30m | Training process was really long |
| Abstract | 2h | 1h30m | - |
| Introduction | 3h | 3h | - |
| **Tot** | **2w3d7h** | **2w2d5h30m** | **-** |

Table 5.3: Work packages sprint 3


Figure 5.4: Sprint 3 burn down chart

### 5.4.4 Sprint 4

| **Work package** | **Estimation** | **Workload** | **Comments** |
|---|---|---|---|
| Techday Presentation | 3h | 3h30m | - |
| Meeting 9 | 1h | 30m | - |
| Document application | 2d2h | 2d1h | - |
| GUI application | 2d2h | 2d4h | - |
| Application diagrams | 1d4h | 1d3h30m | - |
| Poster | 6h | 6h | - |
| Book | 6h | 5h30m | Training process was really long |
| Conclusion | 2h | 1h30m | - |
| Tests | 6h | 5h | - |
| Document review | 6h | 5h | - |
| Project Management update | 1d | 6h | - |

| | | | |
|---|---|---|---|
| Neural Networks introduction | 2d | 1d4h | - |
| Conclusion | 4h | 2h | - |
| **Tot** | **2w3d** | **2w1d6h** | **-** |

Table 5.4: Work packages sprint 4

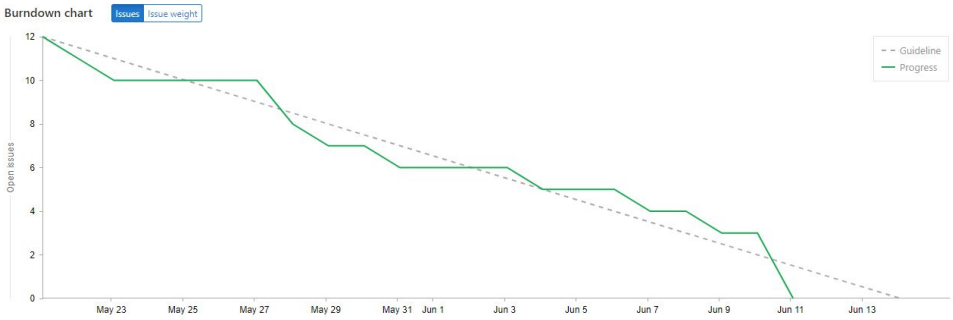


Figure 5.5: Sprint 4 burn down chart

### 5.4.5 Conclusion

According to the Bern University of Applied Science [10] an ETCS credit represents 30 hours work. This thesis is worth 12 ETCS credits which represent 360 hours of works. If the workloads of this section are summed up, they reach a total of 9 weeks 4 hours and 5 minutes which translate in 364 hours. This amount it the right quantity of work needed to acquire the 12 ETCS credits of this Bachelor thesis. Moreover, the different burn down charts show how the work has been equally spread across the entire project and how there have not been delays in the project. All these factors show that all the tasks have been well planned and organized throughout the entire time.

## 5.5 Other Statistics

This section contains other statistics useful to understand the workload.

### 5.5.1 Line of code

The two following images contains the number of lines of code of the prototypes or application. If the two total are summed up, the project has in total almost 3500 lines of code.

| Source File | Total Lines | Source Code Lines |
|---|---|---|
| Classifier.py | 121 | 83 |
| Classifier.py | 176 | 135 |
| confusion_matrix_plot.py | 62 | 47 |
| Features.py | 197 | 118 |
| Features.py | 194 | 95 |
| images_extraction.py | 183 | 110 |
| selective_classifier.py | 361 | 226 |
| selective_search_test.py | 80 | 47 |
| **Total:** | **1374** | **861** |

Figure 5.6: Prototype's lines of code

| Source File | Total Lines | Source Code Lines |
|---|---|---|
| __init__.py | 0 | 0 |
| __init__.py | 0 | 0 |
| __init__.py | 0 | 0 |
| __init__.py | 0 | 0 |
| __init__.py | 0 | 0 |
| __init__.py | 0 | 0 |
| app.py | 431 | 287 |
| bow.py | 595 | 380 |
| classifier.py | 92 | 56 |
| conf.py | 166 | 42 |
| directoryManager.py | 115 | 72 |
| feature.py | 122 | 74 |
| gui.py | 1618 | 1004 |
| imageExtractor.py | 29 | 17 |
| imageManager.py | 88 | 52 |
| randomForest.py | 245 | 171 |
| setup.py | 10 | 9 |
| terminal.py | 357 | 230 |
| testExtractor.py | 30 | 18 |
| websiteExtractor.py | 221 | 144 |
| **Total:** | **4119** | **2556** |

Figure 5.7: Application's lines of code

### 5.5.2 Commits

The following image contains statistics about the project's commits. The Figure 5.8 shows how the workload (12 ETCS = 360 hours) has been spread across the week. This project has been realised for 6 hours Monday and Tuesday, 5 hours Wednesday and Thursday and one extra hour in the other days. This subdivision leads to a 24 hours workload for week that represent, if multiplied with the duration of the project (16 weeks), the right amount of work (16*24 = 368 hours ≈ 360 hours).

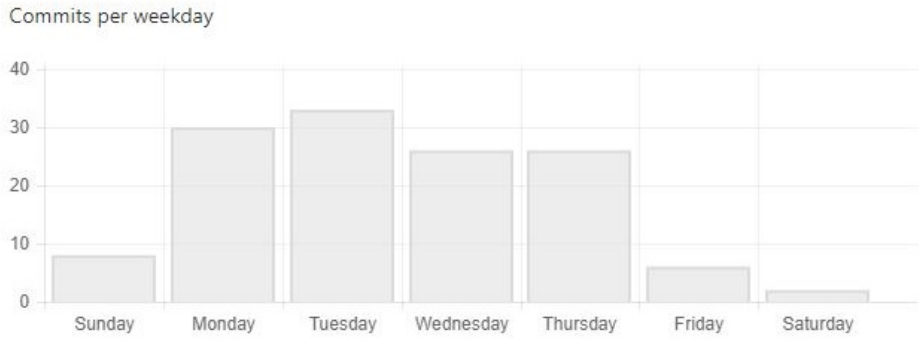


Figure 5.8: Commits statistics

# 6. Risk mitigation

This chapter describes the risks that can occur in this Bachelor thesis. First, the risks are listed and described. Then, the probability of occurrence of each risk and the countermeasures to mitigate their side effects are explained. What is a risk? In Software Engineering a risk is a potential problem, it is an activity or event that may compromise the success of software development project [11].

This definition explains what is a risk but why risk mitigation is so important and what is it exactly? Risk mitigation planning is the process of developing options and actions to enhance opportunities and reduce threats to project objectives [12, pp. 273]. Risk mitigation implementation is the process of executing risk mitigation actions [12, pp. 273]. Risk mitigation progress monitoring includes tracking identified risks, identifying new risks, and evaluating risk process effectiveness throughout the project [12, pp. 273].

Now, with these two definitions, the different risks of this project can be described. The structure used to document the risks is based on [13, pp. 143].

## 6.1 Risks

In this project, most of the risks do not come from quality requirements but from functional and technical requirements. This is because this project aims to compare different algorithms and because it is more research oriented. To be more precise, the technologies discussed in this project could be combined to create a product that must have the highest precision without consider other quality restrictions.

All the risks are listed in Table 6.3.

### 6.1.1 Measures

For the evaluation of the probability of occurrence and side effects a.k.a. impact, different measures are used.

The descriptions shown in Table 6.1 and 6.2 are based on [13, pp. 137].

| Score | Description | Definition |
|---|---|---|
| High | Critical | Will occur frequently, has occurred on past projects, and conditions exist for it to recur |
| Medium | Significant | Will sometimes occur, has happened a minimal number of times on past projects, and conditions are somewhat likely for it to recur |
| Low | Negligible | Will not likely occur, has never occurred on past projects, and conditions don't exist for it to recur |

Table 6.1: Probability measure

| Score | Description | Definition |
|---|---|---|
| High | Critical | A consequence that will cause loss, cause severe interruptions to the customer, or severely delay the completion of a major deliverable |
| Medium | Significant | A consequence that may cause loss, may cause annoying interruptions to the customer, or delay the completion of a major deliverable |
| Low | Negligible | A consequence that may cause minimal loss, cause minimal interruption to the customer, or cause minimal delay to the completion of a major deliverable |

Table 6.2: Impact measure

### 6.1.2 Risks list

| ID | Risk | Description | Probability | Impact |
|---|---|---|---|---|
| 01 | Recognition of logos | Do not find a technology that achieves a sufficient precision | Low | High |
| 02 | Logos in larger images | Do not find a technique to locate logos in larger images with a sufficient precision | Medium | High |

| | | | | |
|---|---|---|---|---|
| 03 | Images / Information acquisition | Websites that contain specific logos could be difficult to find. A shortage of these types of websites reduce the possibility to extract images ~~and useful information (keyword, hyperlinks, etc.)~~[**update 03.04.2018**]. This scarcity can affect the time needed to test and create an application | Medium | High |
| 04 | Scala | ~~The use of Scala as programming language could be a problem~~[**update 17.04.2018**] Scala is no more the chosen language, the application will be developed with Python | Medium | Medium |
| 05 | Classifier | Do not find a classification technique which minimizes enough false positives | Low | High |
| 06 | Technologies | The analysis of the different technologies for features extraction, images classification, objects recognition ~~and information extraction~~ could lead to a too deep investigation and therefore a *waste* of time [**update 03.04.2018**] | Low | High |

Table 6.3: Risks

## 6.2 Countermeasure

This section explains the different countermeasures that will be taken to mitigate the probability of occurence and the side effect of the risks discussed in Section 6.1. The list of countermeasures is available in Table 6.4.

| ID | Risk | Countermeasure |
|---|---|---|
| 01 | Recognition of logos | The results achieved in the project *Logos Recognition* [6] allow to say that the probability to not find a technology with a sufficient precision is very low. Anyway to minimize the problem, a prototype at the *beginning* of the project which proves the efficiency of the chosen technology will be made |
| 02 | Logos in larger images | To minimize the problem of finding a technique to locate logos in larger images with a sufficient precision, different technologies immediately at the beginning of the project are studied. Then, a prototype that prove the technology efficiency is developed |
| 03 | Images / Information acquisition | To minimize the possibility to have a shortage of websites which contain logos, an/a application/prototype using pre-downloaded images ~~and information~~ [**update 03.04.2018**] is developed. This solution allows to prove the feasibility of the project. Once a list of websites is available the program is improved to extract the ~~information and~~ images automatically [**update 03.04.2018**] |
| 04 | Scala | To reduce the probability to have problems with Scala this semester I take part at a Scala course. In addition to that, prototypes in Python are developed so that there is a base as support. [**update 17.04.2018**] Scala is no more the chosen language, the application will be developed with Python |

| | | |
|---|---|---|
| 05 | Classifier | In the project *Logos Recognition* [6] different features extraction and classification technologies are analysed. For this reason, the probability to not be able to create a classifier which minimizes enough false positives is very low. Anyway, to further minimize it, prototypes with different technologies are developed |
| 06 | Technologies | To reduce the probability of a too deep investigation and to do not overtake the boundaries, limits of each operation are described at the beginning of each sprint |

Table 6.4: Countermeasures

# Attachments

**Definition of Bachelor Thesis Project**

# Acronyms

**a.k.a.** also known as. 16

**ff** following. 9

**GUI** Graphic User Interface. 11

**i.e.** that is (*id est*). 11

# Glossary

**LaTeX** Is a mark up language specially suited for scientific documents. 1, 11

**Agile** Agile is a time boxed, iterative approach to software delivery that build software incrementally. 15, 23

**Docker** Docker is container platform for developers to build, ship and run distributed application. These containers are a way to package software in a format that can run isolated on a shared operating system. In fact, the containers do not bundle a full operating system but only libraries and settings required to make the software work. 23

**GIT** Git is a version control system for tracking changes in computer files and coordinating work between multiple people. 11

**GitLab** GitLab is a web-based git repository that have additional features like wiki, issue tracking, etc. This product support CI Runners that allow the user to execute builds on machines created on demand which once the build is finished, can wait to run the next builds or can be removed. This scaling can be achieved with Docker and its public machines. 11–13, 15

**MiKTeX** Is an up-to-date implementation of TeX/LaTeX and related programs. 1

**Scrum** Scrum is an Agile project management framework for completing complex projects. 4, 11, 12, 15, 23

**sprint** A Scrum sprint is a time-box of a month or less which a "Done", usable and potentially releasable product is created. 4, 12–14, 20

**TeXstudio** TeXstudio is an integrated writing environment for creating LaTeX documents. 1

# Bibliography


[1] Latex - a document preparation system. https://www.latex-project.org/. (Accessed on 02/20/2018). 1

[2] Miktexorg - typesetting beautiful documents. https://miktex.org/. (Accessed on 02/20/2018). 1

[3] Texstudio - latex made comfortable. https://www.texstudio.org/. (Accessed on 02/20/2018). 1

[4] K. Pohl and C. Rupp. *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - Ireb Compliant*. Rocky Nook computing. Rocky Nook, 2015. 1

[5] User stories: a beginner's guide to acceptance criteria — boost blog. https://www.boost.co.nz/blog/2010/09/acceptance-criteria. (Accessed on 03/14/2018). 3

[6] Noli Manzoni. Logos recognition. Technical report, Bern University of Applied Science, 2018. Project 2 report. 6, 21, 22

[7] Issues - gitlab documentation. https://docs.gitlab.com/ce/user/project/issues/. (Accessed on 02/21/2018). 9

[8] The pomodoro technique® - proudly developed by francesco cirillo — cirillo consulting gmbh. https://francescocirillo.com/pages/pomodoro-technique. (Accessed on 02/22/2018). 10

[9] Burndown charts. http://www.agilenutshell.com/burndown. (Accessed on 03/05/2018). 12

[10] Individueller und flexibler studienaufbau — bfh - berner fachhochschule. https://www.bfh.ch/de/studium/studienorganisation/studienaufbau.html. (Accessed on 06/11/2018). 16

[11] Risk management in software development and software engineering projects. http://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects. (Accessed on 03/05/2018). 18

[12] Project Management Institute. *A guide to the project management body of knowledge : (PMBOK guide) ; an American National Standard ANSI/PMI 99-001-2008*. Global Standard. PMI, Project Management Inst., 2008. 18

[13] Kim Heldman. *Project manager's spotlight on risk management*. John Wiley & Sons, 2010. 18