

# Title of the master thesis

Subtitle (optional)

Dissertation presented by  
**Zélie MULDER**

for obtaining the Master's degree in  
**Computer Science and Engineering**

Supervisor(s)  
**Kim MENS, Philippe BARET , Adrien DOCKX**

Reader(s)  
**Hélène VERHAEGE,**

Academic year 2017-2018

# Contents

## Chapter 1

# Introduction

## Chapter 2

# Problem analysis in market gardening

In this chapter, we will describe the background of this master thesis. It contains all pieces of information needed to understand the thesis, even without any knowledge of the field processed. The background can be split between the two fields concerned by this master thesis. On the one hand, we have the background concerning the development of the application, related to the field of the software engineering. On the other hand, we can describe the background relative to the field of market gardening.

### 2.1 In market gardening

**Market gardening** A market gardener is someone who produces fruit and vegetables on a relatively small area. The difference between a farmer and a market gardener is principally in the type of final product. Where a farmer will produce more cereals, a market gardener is specialized in fruits and vegetables. We will use the term *Truck farmers* for gardens that are cultivated with heavy machinery.

**Bed** A bed in a market garden is a surface of production, usually a line. It is used to divide the field in smaller cultivated areas. A representation of beds is shown on figure ???. Market gardener usually choose the width of their beds according to the width of the tools they are going to use.



Figure 2.1: Beds in a market garden

**Crop rotation** In order to preserve the soil from draining and to eliminate some diseases specific to some plant species, some market gardener rotate their cropping. If they plant one type of vegetable on a bed, the year after they will plant another type of vegetable on this bed. They will not plant two years in a row the same vegetable on the same bed.

**Work hours** Market gardeners don't count their hours as regular workers. They work all day in order to reach their objectives of the day. Most of them have no idea of how long each culture takes. It also means that they have no idea how profitable their cultures are.

**Planification** Market gardeners often plan during winter (when they have more time) what they are going to crop this year. However, day to day planning is not always easy. From our meetings with farmers, we have learn that it often happens that they have to stop what they are doing because they forgot to treat one of their cultures, for example. Of course, the degree of organisation depends on each market gardener.

## 2.2 Combining these two fields, what is the state of the art ?

From our researchs, there are not lots of software to help farmers of all kind in their daily life. Most of them are not open source.

First, we found softwares like *Mes petits légumes*[?] intended for non-professional market gardeners, with a great library of data about lots of vegetables. The software can be bought once for 19 € or one can use the free incomplete version.

Then, we have softwares like *LEA*[?] more focused on the management of the business and intended for big farms. Once again, the software is not open source and a subscription is required to use it.

Finally, we have found a software that seems to have a purpose and a target audience similar to this project. *Tend*[?] is a software developed in the USA by a Startup.

These softwares show that farmers are in need of tools to help them in their planification and organisation. The poorness of softwares really adapted to their needs show that this field has been forgot by technology.

+ say that this is where this thesis comes in Give more informations on each software. Screenshots.

## Chapter 3

# Problem analysis in software engineering

For each thing you explain also say why you explain it, what role will it play later in this thesis

### 3.1 In computer science and in software engineering

**Agile Methodology** The Agile methodology is a set of techniques and principles for conducting a development project. The main principle is to iterate over short periods (called sprints) divided in subphases in order to build the final product in an incremental way. A sprint lasts between one and two weeks and is divided as follows:

- At the start of each iteration, we plan with the client what we are going to do this iteration.
- Next step is to think about how to build a good design to achieve the objective
- Then, we develop the features
- After, we test these features. If we switch these last two steps, we apply what is called Test-Driven Development. With this methodology, a team write tests before developing the corresponding features, ensuring that the tests will cover all the cases
- At the end of each sprint, we meet the client again to validate the changes and new features, collect his feedback about the project's progress and to define together the future work.

A visual representation of this iterative approach is shown on figure ??

A good summary of the Agile methodology is the agile manifesto[?], that states the main principles of this methodology.

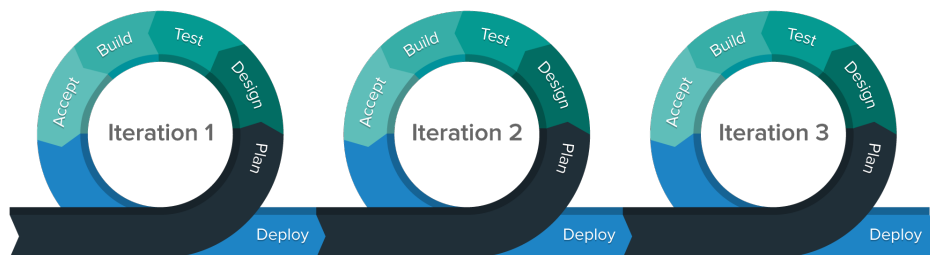


Figure 3.1: Iterations using Agile methodology

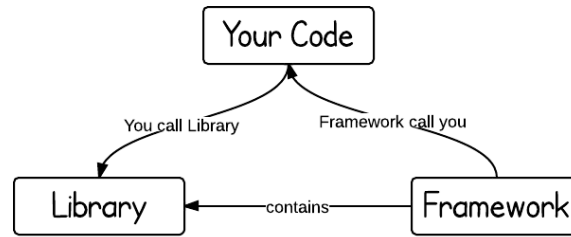


Figure 3.2: Framework versus library



### Agile Manifesto [?]

*Individuals and interactions* over processes and tools.

*Working software* over comprehensive documentation.

*Customer collaboration* over contract negotiation.

*Responding to change* over following a plan.

That is, while there is value in the items on the right, we value the items on the left more.

The main advantages of this methodology are to have flexibility and regular feedback on the product delivered.

**Python** Python is a programming language used in many fields. Its main advantage is great readability.

Talk more about python? Creator, specificity's, interpreted language, speed, community, TIOBE Index...? <https://www.tiobe.com/tiobe-index/> Mainly explain why you chose it and what its main features/specificities you will rely upon

**Software framework** A framework is a set of tools that provides features that can be reused in multiple applications. A framework is different from a library in the sense that when using a library, we call the methods of the library while when using a framework, our code is called by the framework. The difference is shown on figure ???. Frameworks help to build reusable and maintainable applications

give some examples of frameworks + why you talk about it, integrate Django with it?

**Django** Django is a web framework to develop web applications in Python.

Develop, why django, what others, what features of Djangos do you rely upon, what plugins, and so on

**Continuous Integration** Continuous integration (often referred as CI) is a software engineering practice to help development teams working on a set repository. When using CI, we automate a set of checks at each push request on a repository and developers should merge to this repository regularly. With continuous integration, we will detect bugs faster and more easily. We can also add tests to our continuous integration tool, these tests will then be run at each push request and we will receive a notification if they fail. Thanks to that, we can see exactly which commit implied a failure in the tests. If we do small regular commits, it will be really easy to find where the failure comes from.

Why you explain this, what CI you use, what CI features and tools you use; why you use these

## Chapter 4

## Solution



## Chapter 5

# Architecture

## Chapter 6

# Implementation

# Chapter 7

## Validation

## Chapter 8

# Conclusion and future work

