Software Project Management 2015/2016

Smart shopping list

**Team Members:** Stefan Viktoler (team leader), Filip Michnik, Wang Yongjian, Udeshkumar Ganesan, Amélie Lagadec

**Project Proposal**

**Version 0.4**

History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Remarks |
| 0.1 | 24/11/2015 | Amélie Lagadec | Creation of document |
| 0.2 | 01/12/2015 | Amélie Lagadec | Strategic plan, Test plan, Schedule |
| 0.3 | 05/12/2015 | Wang Yongjian(Adam) | Problem statement |
| 0.4 | 07/12/2015 | Wang Yongjian(Adam) | Business case, Risk Management |
|  |  |  |  |

Table of contents

[1 Executive overview 3](#_Toc436749452)

[1.1 The project 3](#_Toc436749453)

[1.2 Benefits 3](#_Toc436749454)

[1.3 Likelihood of success 3](#_Toc436749455)

[2 Problem statement 3](#_Toc436749456)

[3 Business Case 5](#_Toc436749457)

[3.1 Profits 5](#_Toc436749458)

[3.2 Safety 5](#_Toc436749459)

[3.3 Benefit to the community 5](#_Toc436749460)

[4 Risk Management 6](#_Toc436749461)

[5 Cost Estimate 9](#_Toc436749462)

[6 Communications Plan 9](#_Toc436749463)

[6.1 Meeting schedule 9](#_Toc436749464)

[6.2 Point of contact 9](#_Toc436749465)

[6.3 Communications media 9](#_Toc436749466)

[7 Legal Issues 9](#_Toc436749467)

[8 Staffing Plan 9](#_Toc436749468)

[8.1 Participants in the project 9](#_Toc436749469)

[8.2 Briggs Meyers compatibility matrix 9](#_Toc436749470)

[9 Strategic Plan 10](#_Toc436749471)

[9.1 Balanced Scorecard 10](#_Toc436749472)

[9.2 SWOT 11](#_Toc436749473)

[10 Test Plan 11](#_Toc436749474)

[10.1 Functions to be tested 11](#_Toc436749475)

[10.2 Functions not to be tested 11](#_Toc436749476)

[10.3 Testing process overview 11](#_Toc436749477)

[10.4 Testing Strategy 13](#_Toc436749478)

[10.4.1 Data and Database Integrity Testing 13](#_Toc436749479)

[10.4.2 Function testing 13](#_Toc436749480)

[10.4.3 User interface testing 13](#_Toc436749481)

[10.4.4 Security and Access Control Testing 13](#_Toc436749482)

[10.4.5 Configuration and installation Testing 14](#_Toc436749483)

[11 Schedule 14](#_Toc436749484)

[12 Design structure matrix 16](#_Toc436749485)

# Executive overview

## The project

This report elicits key points of a software projects which purpose is to develop an application for Smart phones. This application allows its user to analyze data that is generated by a smart object (refrigerator) in an efficient and intuitive way.

The Internet of things is basically a network of physical objects (so called “things”) that had been embedded with software and equipped with sensors and network connectivity, thus enabling them to collect and exchange data with each other. These objects can analyze their environment and can be controlled remotely across existing network structures such as the TCP/IP Protocol. Objects have a unique identifier (such as their IP-Address) and in cooperate themselves into the existing Internet infrastructure.

“Things” can refer to a wide range of physical objects, such as manufacturing plants, cars, pallets in a warehouse or refrigerators, amongst others. All these objects are endowed with specific embedded software and sensors (e.g. RFID-Readers) that can generate data based on their input data. Smart objects can feed their observations into the existing Internet, thus allowing objects or applications to exchange information with each other.

The smart home is the environment created by turning physical objects of households into smart objects. Possibilities are versatile from smart stoves and pots, which automatically alert their owners, to smart refrigerators, with groceries that are equipped with RFID-Tags, allowing users to analyze their nutrition habits, inform them of shortages or generate shopping lists. The varieties of possible application are huge.

## Benefits

## Likelihood of success

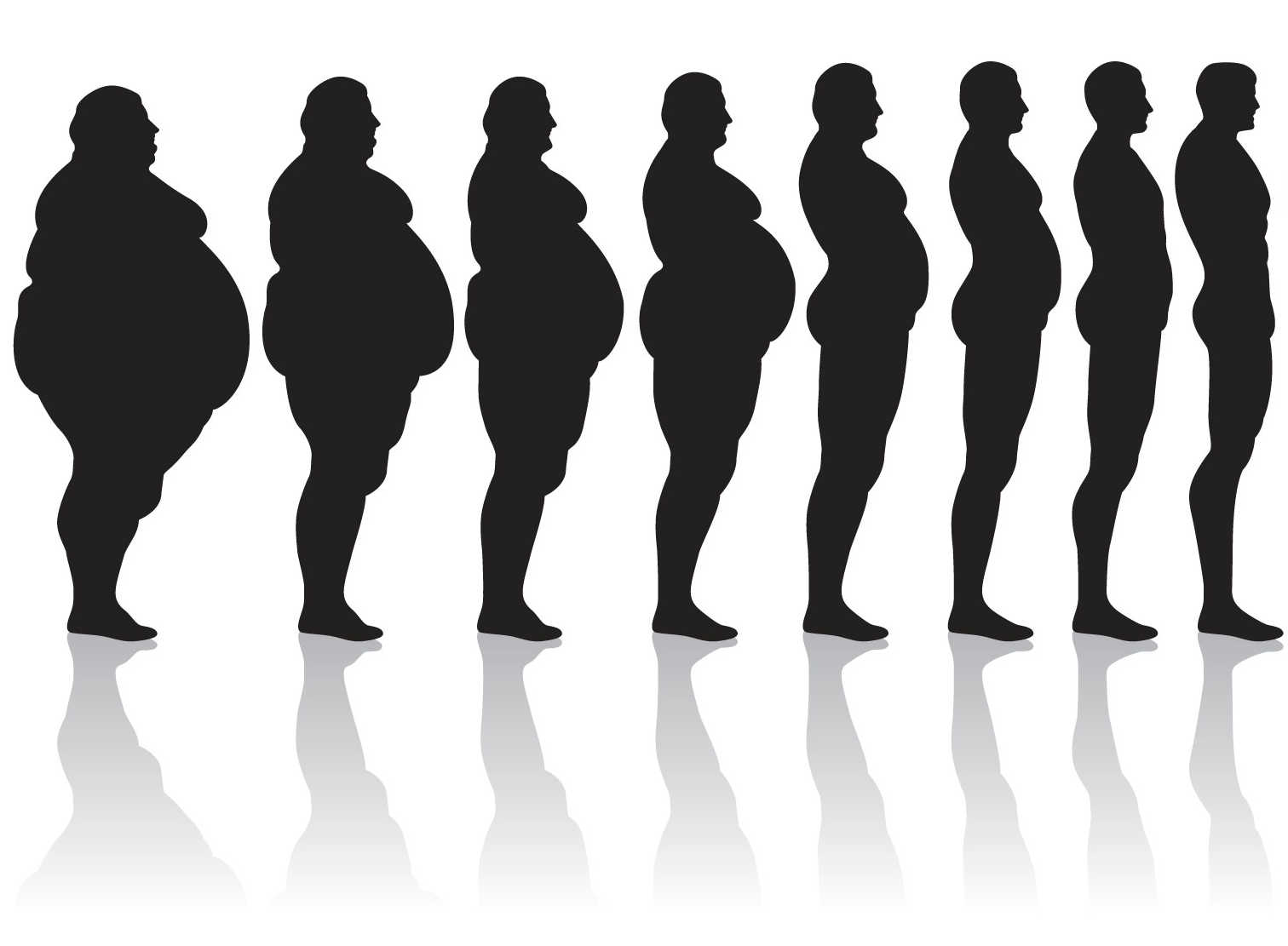
# Problem statement

Regarding a full refrigerator like below, sometimes we could forget the food until it becomes spoiled. It happens and happens a lot. What a waste! We would like to solve the problem through an application.



A full refrigerator

Preventing the food from spoiling is not enough, what’s more? Fat population is bigger and bigger. Exercise is important and a balance diet is also important, we would like everyone to have a good health. We could achieve this by our smart shopping list application.



Fat can be thin

Problem 1: People left food in their refrigerator until it become spoiled.

Problem 2: The fat population becomes bigger and bigger, people need a balanced diet.

The purpose of this project is to develop a mobile application which supports different mobile operating systems and analyzes the data generated by a smart refrigerator and reminds the user in case that the user may forget the food which is going to spoil. Generating a balance and health shopping list will save you a lot of time and gives you a healthier life.

# Business Case

## Profits

As smartphone and internet of things developing, our life has changed a lot. It is a trend that our life will be more intelligent and those companies which seize this opportunity will have a great business.

For the refrigerator company, selling intelligent refrigerator will make a great profit and make a revolution of the traditional refrigerator.

For the customers, investing on health is really worthy. They can now buy a smart refrigerator, prevent food from spoiling and have a healthy diet.

## Safety

The product is not a strange new thing that hard for customer to accept. The internet of things has developed itself for many years and lots of matured products have appeared in the market. We already live in a life surrounded by a lot of smartphone applications. We use applications to manager our bank account, to listen to music, to watch video, to communicate with friends, and even to do business.

Regarding the investment, it won’t be huge and the risks of the product are small. We are not going to develop a fresh new refrigerator, we are enhancing the current refrigerator hardware with sensors and mobile application. It is not as hard as invent a new kind of refrigerator but the benefit will be the same as developing a new refrigerator.

## Benefit to the community

The smart shopping list benefits the community of consumers in which every people belongs. Indeed, we don't have to create a shopping list anymore before heading out to the supermarket. The users will not have to change any behavior to maintain this inventory list and thanks to the application we never eat stale or spoiled food. Over time, we will be able to see the patterns of our food consumption and better manage our household budget.

# Risk Management

Every aspect of software project management involves some form of risk management. For our project there are several possible risks.

**Risk 1:** We are software engineers not familiar with the hardware

**Plan to mitigate:** Ask the refrigerator company RD members to train us.

**Risk 2**: The 2 main mobile platforms development are quite different, parallelly develop two different platforms mobile app will cost a lot of manpower.

**Plan to mitigate:** Focus on iOS platform first, after iOS app finished we begin the Android app development.

**Risk 3:** we don't have QA team, software quality may not meet the requirements.

**Plan to mitigate:** 1. Several phases test, Development Test, Alpha Test, Beta Test

2. Outsource the test.

**Risk 4:** if too many new features needed to be added and the requirements changes out of control.

**Risk 5**: if there is no enough time budget to adapt the software to so many different android phones.

The most effective means of managing risk is by using a combination of strategic (that is, planning to avoid risk) and tactical (that is, preparing to take remedial action if adverse events occur) methods. Here we will use two risk mitigation methods to manage the risks.

## IRE through Identify, Rank and Evaluate to mitigate risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Relative Severity** | **Relative Probability** | **Phase** | **Description** | **Mitigation** |
| 1 | 1 | Design | Not familiar with the hardware | Ask the refrigerator company RD members to train us. |
| 3 | 2 | Coding | Two main mobile platforms development are quite different, parallelly develop two different platforms mobile app will cost a lot of manpower | iOS platform first, later Android app |
| 2 | 3 | Testing | Don't have QA team | 1.Several phases test, Development Test, Alpha Test, Beta Test 2.Outsource the test |
| 5 | 4 | Requirements acquirement | Requirements changes out of control. | freeze the requirements after achieving the requirements acquisition |
| 4 | 5 | Testing | No enough time budget to adapt the software to so many different android phones | 1.Strict API level to Android v2.3 2.Online Android compatibility test suit |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Relative Severity** | **Relative Probability** | **Phase** | **(Relative) Severity X Probability =** | **Relative Ranking based on the equation Severity X Probability** |
| 1 | 1 | Design | 1 | 1 |
| 3 | 2 | Coding | 6 | 2 |
| 2 | 3 | Testing | 6 | 3 |
| 5 | 4 | Requirements acquirement | 20 | 5 |
| 4 | 5 | Testing | 20 | 4 |

Based on above IRE table, we will focus on the top 3 risks. We will use each mitigation way to mitigate corresponding risk. And we will still monitor the other risks and new occurring risks.

## Simple Risk Estimate/Breakeven Point based on Standish Report

• Pf = probability a project is a type 2 or type 3 or ~ 71%

• R0 = the ratio of these projects’ overrun to the original budget to be ~ 56%

• Rf = the ratio of the cost of these projects to the original or ~ 156%

• B is the initial budget of the software project

Ee = PfBRf + (1-Pf)B (formula 1)

B = Ee/(PfR0 + 1) (formula 2)

RJ as a fraction of B.

Spending RJB on risk management activities reduces the average probability of failure Pf by a fraction, ∆Pf.

∆Pf is in units of the probability of failure

Ee = RJB + (Pf - ∆Pf)BRf + (1-(Pf - ∆Pf))B (formula 3)

∆Pf = RJ/RO (formula 4)

Because we don’t have our history data, we will use the data from Chaos report. Here we will set aside budget for the risk management activity.

According formula 2 and data from chaos report, B = 1.7Ee, Rj=0.39. We will prepare additional 0.39B to mitigate the risks.

Finally we must keep in mind that no matter how greatly increase our chances of mitigating the impact of adverse events, there are no guarantees that we will avoid them altogether. So we will always monitor the risks during the whole lifecycle of the development.

# Cost Estimate

# Communications Plan

## Meeting schedule

## Point of contact

## Communications media

# Legal Issues

# Staffing Plan

## Participants in the project

We work in a software company where basic infrastructure and developing techniques are available for this project. In this case we do not need to spend a lot of time thinking about best ways of storing data, communicating via TCP/IP and go through design principles.

The team is described as follow:

|  |  |
| --- | --- |
| **Name** | **Role** |
| **Stefan Viktoler** | Team leader |
| **Filip Michnik** | Developer / Designer |
| **Wang Yongjian** | Developer / Designer |
| **Udeshkumar Ganesan** | Developer / Tester backup |
| **Amélie Lagadec** | Developer / Tester |

In case of some problems, the team can be extended.

## Briggs Meyers compatibility matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Stefan ENTP** | **Filip ENFP** | **Wang ESFJ** | **Udesh ESTP** | **Amélie ISFJ** |
| **Amélie ISFJ** | 0.50 | 0.33 | 0.50 | 0.33 |  |
| **Udesh ESTP** | 0.83 | 1.00 | 0.50 |  |  |
| **Wang ESFJ** | 0.67 | 0.50 |  |  |  |
| **Filip ENFP** | 0.83 |  |  |  |  |
| **Stefan ENTP** |  |  |  |  |  |

# Strategic Plan

## Balanced Scorecard

The following balanced scorecard is a strategic planning and management system that is used to align business activities to the vision and strategy of our organization, improve internal and external communications and monitor organization performance against strategic goals.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Objective** | **Measurement** | **Target** | **Initiative** | **Owner** | **Supplier** |
| **Financial** | -Lower Support costs  -increasing longterm revenue  - maximize returns | -Support resource costs  -Revenue  - ROCE | < 20% of the  software costs  -5% growth  - 14% | ITIL  restructuring | IT Department | Service Desk  employees |
| **Customer** | -User satisfaction  -Fast response time | -Customer Market survey  -flow time | -#1  -<48 hours | -Quality  management  -Problem  manager | -QM  Department  -IT Department | -Quality manager  -Service Desk  employees |
| **Internal** | - Better customer  knowledge  -Develop innovative projects  - Max return on ressource allocation | -Customer  service  - % R&D project  - employee productivity improvement | - 10% growth of market  share  - 80%  -5% | -Product  evolvement  - Project manager | -Release and  Marketing manager  - Project manager | - Company members  -Employees |
| **Learning** | - Help Desk Team  education  -Ensure market driven skills  - Leading employee satisfaction | - % of team  trained on  the time  - strategic skill coverage ratio  - Employee satisfaction rating (5 point scale) | - 100% in 1  year  -65%  - 4 | - Consultant  from Development  team  - Consultants  - Project manager | - Dual Role  Help Desk  and  Development  team  -Development team | - Help Desk and Development  team  -Development team |

## SWOT

The following SWOT matrix has the aim to evaluate the strengths, weaknesses, opportunities and threats involved in our business venture.

|  |  |  |
| --- | --- | --- |
|  | **Positive** | **Negative** |
| **Internal** | * Innovative aspects (smart shpping list) * Experience * Knowledge * Data * Quality * Location and geography * Price, value * Financial reserves | * Continuity * Support robustness * Lack of competitive strength * Cash flow * Our vulnerabilities * Reliability of data * Commitment |
| **External** | * Competitors vulnerability * New partnerships * New markets * New technologies * Fashion influences * Technology development and innovation * Customer satisfaction | * Competitor intentions * Economy * Legislative effects * Shift in consumer tastes * Substainable financial backing * New regulations |

# Test Plan

The Test Plan is designed to prescribe the scope, approach, resources, and schedule of all testing activities of the Smart shopping list project.

The plan identifies the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and the schedule required to complete testing.

## Functions to be tested

1. Graphical User Interface

2. Reports Output/Data

3. Report Setup/Locations

## Functions not to be tested

Not other than mentioned above in section 10.1.

## Testing process overview

**Understanding Requirements**

Requirement specifications will be sent by the client.

Understanding of requirements will be done by the whole team.

Raised queries will be sent by lead to the client.

Response to queries will be sent by the client.

**Preparing Test Cases**

Amélie Lagadec will be preparing test cases based on the requirement specifications. This will cover all scenarios for requirements. Also, the performance of the application for its responsiveness, scalability, resource usage and stability will be tested.

The usability testing will be taken in count (usability issues, navigation and content) as well as the security and compliance.

**Preparing Test Matrix**

Amélie Lagadec will be preparing test matrix which maps test cases to respective requirement. This will ensure the coverage for requirements.

**Reviewing test cases and matrix**

Peer review will be conducted for test cases and test matrix by Stefan Viktoler

In certain cases for e.g. complex requirements, lead's help will be taken for conducting review

Any comments or suggestions on test cases and test coverage will be provided by Udeshkumar Ganesan

Re-worked improvements will be reviewed and approved by Stefan Viktoler

**Creating Test Data**

Test data will be created by Amélie Lagadec on client's developments/test device based on scenarios and Test cases.

**Executing Test Cases**

Test cases will be executed by Amélie Lagadec on client's development/test device based on designed scenarios, test cases and Test data.

Test result (Actual Result, Pass/Fail) will updated in test case document

**Retesting and Regression Testing**

Retesting for fixed bugs will be done by Amélie Lagadec once it is resolved by Filip Michnik and bug/defect status will be updated accordingly. In certain cases, regression testing will be done if required.

**Deployment/Delivery**

Once all bugs/defect reported after complete testing is fixed and no other bugs are found, report will be deployed to client’s test site by the developers.

Once round of testing will be done by Amélie Lagadec on client’s test site a report will be delivered along with sample output by email to Stefan Viktoler.

After reviewing, Stefan Viktoler will send the report delivery email to the client.

## Testing Strategy

We can already provide the following testing strategy. It will be reviewed to add the client’s requirements.

### Data and Database Integrity Testing

The databases and the database processes should be tested as separate systems. These systems should be tested without the applications (as the interface to the data).

We will ensure Database access methods and processes function properly and without data corruption. This will be done by invoking each database access method and process, seeding each with valid and invalid data (or requests for data). We will inspect the database to ensure the data has been populated as intended, all database events occurred properly, or review the returned data to ensure that the correct data was retrieved (for the correct reasons).

Completion Criteria : All database access methods and processes function as designed and without any data corruption.

### Function testing

The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. We will thus verify the application by interacting with the application via the GUI and analyzing the output (results).

We will ensure proper application navigation, data entry, processing, and retrieval by executing each use case or function, using valid and invalid data. This will verify that the expected results occur when valid data is used, the appropriate error / warning messages are displayed when invalid data is used and that each business rule is properly applied.

Completion Criteria: All planned tests have been executed and all identified defects have been addressed.

### User interface testing

The goal of User Interface Testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the applications. In addition, UI Testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.

We will verify that navigation through the application properly reflects business functions and requirements, including window to window, field to field, and use of access methods. The window objects and characteristics, such as menus, size, position or state will be conformed to standards.

We will create / modify tests for each window to verify proper navigation and object states for each application window and objects.

Completion Criteria: Each window successfully verified to remain consistent with benchmark version or within acceptable standard.

### Security and Access Control Testing

We will verify that the user can access only those functions / data for which their user type is provided permissions and that only those users with access to the application are permitted to access them.

For that we will list each user type and the functions / data each type has permissions for. We will create tests for each user type and verify permission by creating transactions specific to each user type. We will modify user type and re-run tests for same users. In each case, we will verify that those additional functions / data are correctly available or denied.

Completion Criteria: For each known user type the appropriate function / data are available and all transactions function as expected and run in prior Application Function tests.

### Configuration and installation Testing

We will validate and verify that the client application is installed and function properly on the prescribed client mobile device.

# Schedule

To deliver a working application to display all required information (e.g. shopping list) which is connected via TCP/IP to the smart object at home, we assume that there is already an interface implemented to read the data given from the object. Therefore the main effort is in planning developing an intuitive design for the customer and user of the (shopping list) app.

The project will start the 4th of January 2016.

To achieve our goal we have to specify all requirements. This planning phase is estimated by 16 days, including all meetings with the refrigerator manufacturers, getting all information about the API to communicate, all needed functionalities for the app, the analysis of requirements and the documentation.

Thereafter we can start to design and prototype our application regarding to the specified requirements. For this task we calculate a time period of 40 days.

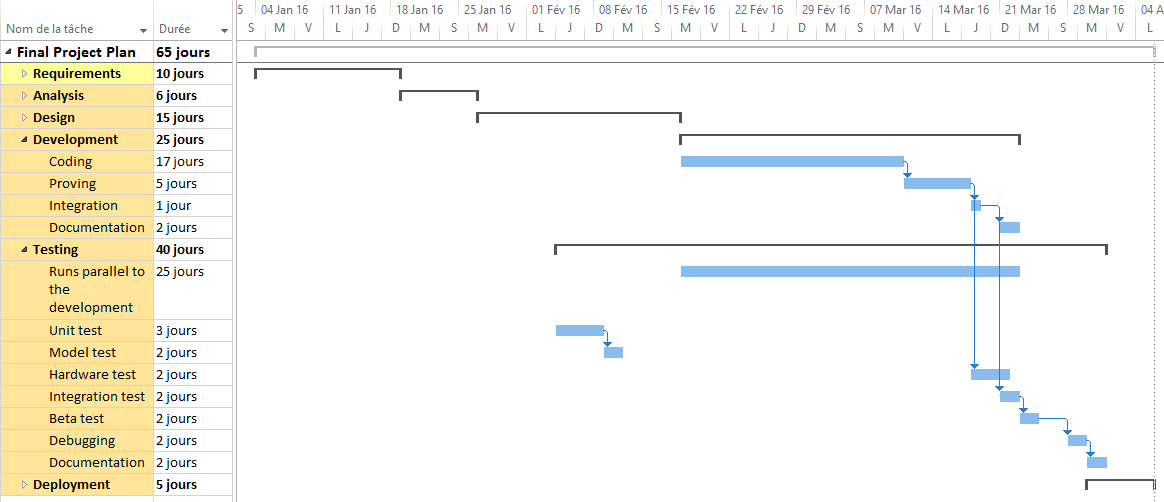
After finishing our first full functional app, we start to improve all designs and functionality, perform unit testing and acceptance testing until our final prototype and first productive application can be launched. This challenge is estimated by 40 days.

Then the project evolves to the support to maintain and possibly extend the application.

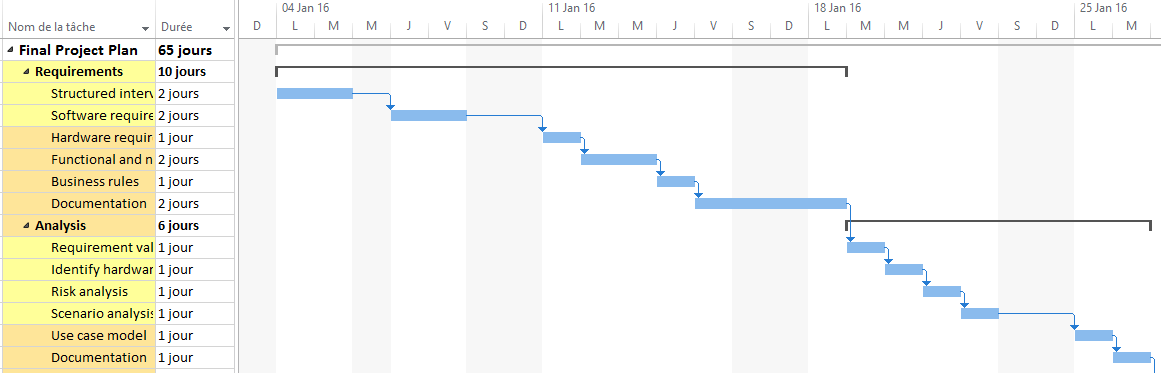
In total the development of an application suited for a smart home object requires 65 days.

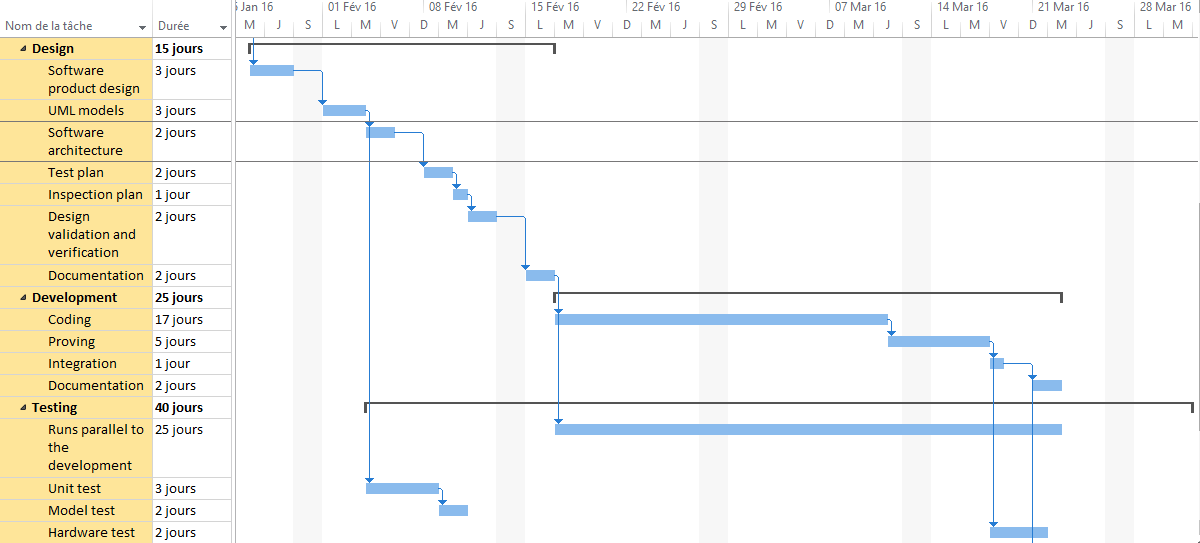
The project will end up on the 5th of April 2016.

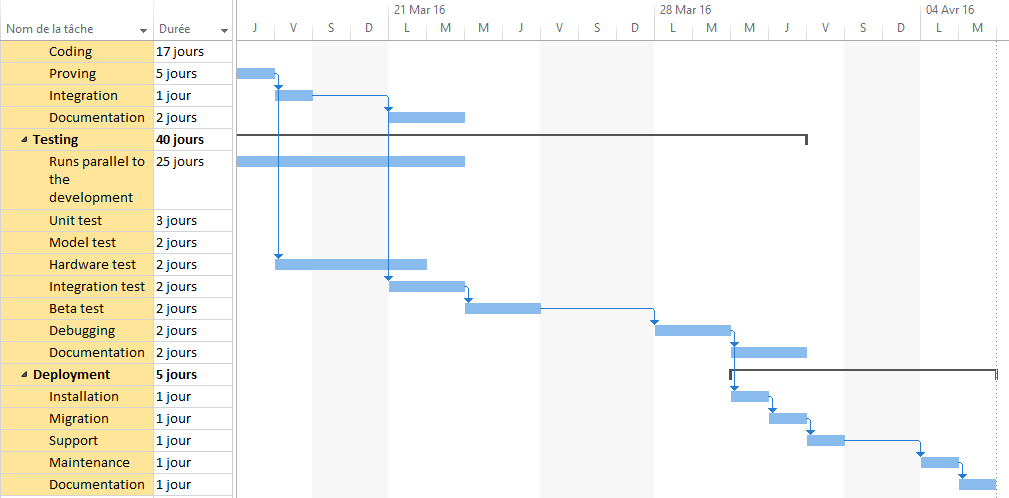
Here is the overview of the project in a Gantt chart format :



Here is the detailed different steps of the project :







# Design structure matrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**References**

Getting Results from Software Development Teams – Chapter 4: Developing and Maintaining the Project Plan: Section: Risk Managment - Microsoft Press 2008 Lawrence J. Peters

A Simple Estimate of the Cost of Software Project Failures and the Breakeven

Effectiveness of Project Risk Management - Stephen P. Masticola

Course Registration System, Test Plan for the Architectural Prototype http://sce.uhcl.edu/helm/RUP\_course\_example/courseregistrationproject/artifacts/test/plans/test\_plan\_arch.htm#\_Toc449511176

How to Create a Test Plan : http://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html