НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ІПСА

КАФЕДРА СИСТЕМНОГО ПРОЕКТУВАННЯ

Розрахунково-графічна робота

з курсу: «Проектування інформаційних систем»

Виконав:
студент 4 курсу
групи ДА-71
Михайловин Р. Г.

Київ – 2020

Зміст

Automatic pet feeder

Software Requirements Specification Version

1.0
27.12.2020

Mikhailovyn Roman

Revision History

Name	Date	Reason for changes	Version
Mikhailovyn R. G.	2020-12-27	Document creation	1.0

Document Approval

Signature	Printed Name	Title	Date
	Mikhaylovin		27.12.2020

1. Introduction

1.1 Purpose

The purpose of this SRS is to define functional and nonfunctional requirements, use cases that describe user interactions with the application to provide them with a perfect experience.

1.2 Document Conventions

● Application - “Notes” app, a main object of this SRS.
● End User - any person who uses the application.
● Functional Requirements - requirements for a special feature: how it works, where stores data, how provides user experience.
● Nonfunctional Requirements - requirements that can not be described as a feature, for example Security requirements.
● Front-end - part of the application end user interacts with.
● Back-end - server-side of the application. Can not be seen by the end user, main calculations happen there.
● FAQ - frequently asked questions.
● Software Requirements Specification - a document which completely describes all of the functions of a proposed system and constraints under which it must operate.

➔ **X. Bold Text** - name of the paragraph
➔ **X.X Bold Text -** name of the subparagraph
➔ *Italic Text* - name of the resource

1.3 Intended Audience and Reading Suggestions

This document is written for developers, designers, marketing staff, testers and managers. Rest of the documents contain requirements for the application, its features and implementation details. Document is divided into paragraphs and subparagraphs which can be easily reached via Table of Contents.

1.4 Project Scope

The purpose of application the Automatic pet feeder mobile application. It’s features of this system, and user interface, what the system will do, constraints under which it must operate and how system will react. This document is intended for both stakeholders and developers.

This system is intend to be a mobile app for everybody who wants to feed their pets with automatic feeder. The software products name is “Automatic pet feeder”.

Whole system is a mobile app which is connected to database.

1.5 References

IEEE. *I EEE Std 829-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

2. Overall Description

2.1 Product Perspective

The application fulfills user needs in a way of providing intuitive user experience and providing an ability to calculate among of feed for their pets.

2.2 Product Features

1. Create, Edit, Delete pet profiles
2. Create, Edit, Delete food profiles
3. Record feeding history
4. Analyze feeding history

2.3 Operating Environment

Back-end of the app will be put into containers and launched at DigitalOcean cloud platform. MongoDB will be used as a main storage system.
Front-end part of the application can be launched at Android devices and iOS devices.
Minimal Android version requirement - Android 10.
Minimal iOS version requirement - iOS 10.3.4.

2.4 Design and Implementation Constraints

Back-end part programming language - Go
Android part programming language - Kotlin
iOS part programming language - Swift
Web version programming language - JavaScript
Database - MongoDB
Connection between front-end part and back-end part will be via REST architecture.

2.5 User Characteristics

There is one type of user:
User – can Create, Edit, Delete, View profiles via GUI.

2.6 User Documentation

Documentation for end users will be in the form of a web page. Also it will be a web page with FAQ and a support chat.

3. Specific Requirements

3.1 External Interfaces

3.1.1Software Interfaces

The mobile device has to have an iOS or Android operating system. Minimal requirements for operating systems see in 2.4.

3.1.2Hardware Interfaces

Hardware interface is represented in the form of a mobile device.

3.1.3User Interface

The application GUI provides different screens for viewing, creating and editing profiles.

3.2 Functional Requirements

3.2.1Create a profile

User enters the app and sees a “Create new profile” button in profile list. After the user presses the button one is presented with a window where a profile can be created. User need enter name, also he can add photo and proteins, fats, carbohydrates and some optional data.

3.2.2Edit a profile

User can edit profiles. He needs click field which he wants to edit and enter new data then click agreement button.

3.2.3Delete a profile

A profile can be delated if user holds profile button, there is open field with confirmation.

3.2.4View a Note

A profile can be viewed via app. After the user enters the app one sees a list of profile. After tapping on the profile a window with profile information is opened. The profile can be edited if the user presses one the fields To come back to the profile list user has to press the button “Back” in the upper left corner.

3.3 Performance Requirements

The app itself does not have specific performance requirements except working smoothly and not to exit suddenly.

3.3 Security Requirements

All the user's pet personal data is secured and encrypted.

3.4 Availability Requirements

The app is absolutely free and can be downloaded via Google Play or App Store.

3.5 Portability Requirements

The app is ported to different systems as was mentioned above: Android, iOS.

3.6 Maintainability Requirements

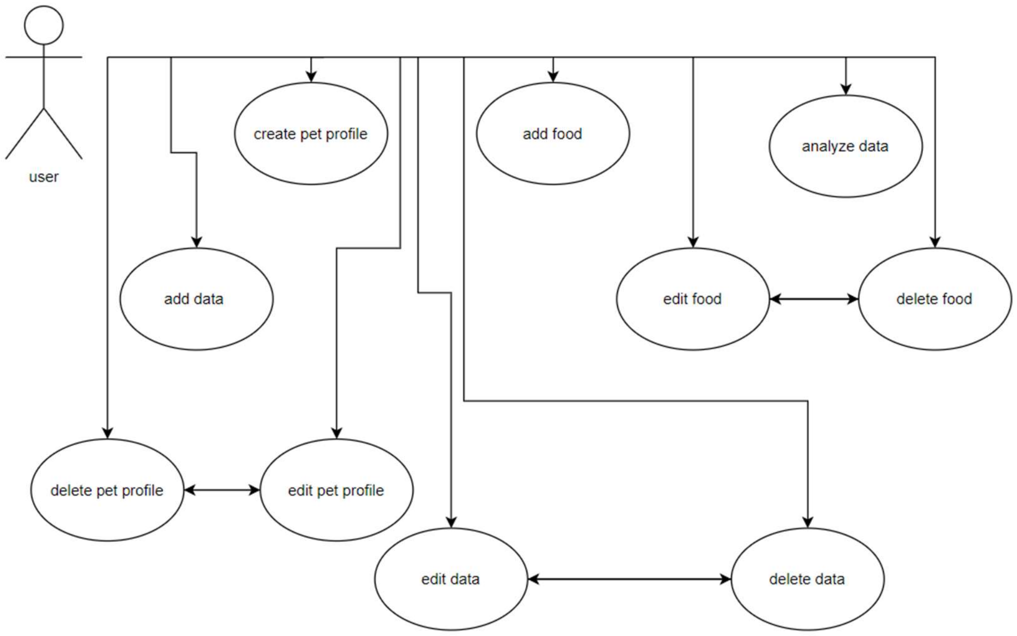
The app is always maintained and supported. New versions of the app will be supplied once two months.

3.7 Logical Database Requirements

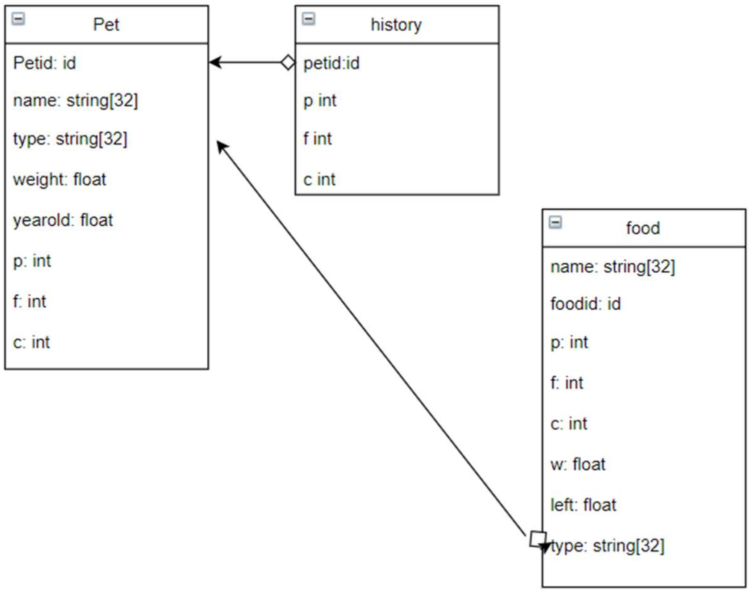
All the information will be stored in MongoDB. It has two main collections: pet profiles, food profiles, and history. profiles’ collection record contains the characteristics and name. History collection record contains feeding data.

UML Diagrams

Use Case Diagram



UML Class Diagram



Test Plan

1. Test Plan Identifier PIS-001

2. References

Automatic pet feeder App Software Requirement Specification Version 1.0

3.Introduction

This is a test plan for Automatic pet feeder App. The primary focus of this plan is to make sure that main processes of the app work as designed and provide maximum convenience for the end user. The project will have three levels of testing, Unit, System/Integration and Acceptance. The details for each level are addressed in the approach section and will be further defined in the level specific plans.

4.Test Items

a. User Interface Version 1.0
b. Main Functionality: API AddProfile, API GetProfile, API UpdateProfile, API DeleteProfile Version 1.0

5.Software Risk Issues

a. Backup and Recovery of all data
b. Ability to save progress in case the app stops in the middle of the process

6.Features To Be Tested

a. Creating a profile
b. Viewing a profile
c. Editing a profile
d. Deleting a profile

7. Features Not To Be Tested

Not specified

8. Approach

8.1 Testing Levels

The testing for the Automatic pet feeder app will consist of Unit, System/Integration (combined) and Acceptance test levels. It is necessary that there will be at least one full time independent test person for system/integration testing. UNIT Testing will be done by the developer and will be approved by the development team leader. Proof of unit testing (test case list, sample output, data printouts, defect information) must be provided by the programmer to the team leader before unit testing will be accepted and passed on to the test person. All unit test information will also be provided to the test person.

SYSTEM/INTEGRATION Testing will be performed by the test manager and development team leader with assistance from the individual developers as

required. No specific test tools are available for this project. Programs will enter into the System/Integration test after all
critical defects have been corrected. A program may have up to two Major defects as long as they do not impede testing of the program (I.E. there is a work around for the error).
ACCEPTANCE Testing will be performed by the actual end users with the assistance of the test manager and development team leader. Programs will enter into the Acceptance test after all critical and major defects have been corrected. A program may have one major defect as long as it does not impede testing of the program (I.E. there is a work around for the error).

8.2 Test Tools

a. Soap UI
b. Go built-in test util
c. Manual testing with IPhone 11 and Samsung Galaxy S20

8.3 Meetings

The test team will meet once every two weeks to evaluate progress to date and to identify error trends and problems as early as possible. The test team leader will meet with development and the project manager once every two weeks as well. These two meetings will be scheduled in different weeks. Additional meetings can be called as required for emergency situations.

8.4 Measures and Metrics

The following information will be collected by the Development team during the Unit testing process. This information will be provided to the test team at program turnover as well as be provided to the project team on a biweekly basis.
a. Defects by module and severity.
b. Defect Origin (Requirement, Design, Code)
c. Time spent on defect resolution by defect, for Critical & Major only. All Minor defects can be totaled together.

The following information will be collected by the test team during all testing phases. This information will be provided on a biweekly basis to the test manager and to the project team.

a. Defects by module and severity.
b. Defect Origin (Requirement, Design, Code)
c. Time spent on defect investigation by defect, for Critical & Major only. All Minor defects can be totaled together.
d. Number of times a program submitted to the test team as ready for

test.
e. Defects located at higher levels that should have been caught at lower levels of testing.

9. Item PASS/FAIL Criteria

The test process will be completed once the test team lead concludes that all conditions for success passing are completed.

10. Suspension Criteria And Resumption Requirements

a. If conditions for test level are not completed
b. Test team does not have appropriate environment for testing

11. Test Deliverables

a. Acceptance test plan System/Integration test plan
b. Unit test plans/turnover documentation Screen prototypes
c. Report mock-ups
d. Defect/Incident reports and summaries Test logs and turnover reports

12. Remaining Test Tasks

Task	Assigned To	Status
Create acceptance test plans	TM, PM, Client	
Create System/Integration Test Plan	TM, PM, Dev	
Define Unit Test rules and Procedures	TM, PM, Dev	
Define Turnover procedures for each level	TM, Dev	
Verify prototypes of Screens	Dev, Client, TM	

13. Environmental Needs

The following elements are required to support the overall testing effort at all levels:

a. iOS
b. Android OS
c. MongoDB

14. Staffing And Training Needs

It is preferred that there will be at least one (1) full time tester assigned to the project for the system/integration and acceptance testing phases of the project. This will require assignment of a person part time at the beginning of the project to participate in reviews etc... and approximately four months into the project they would be assigned full time. If a separate test person is not available the project manager/test manager will assume this role.

No special training for using the App is required.

15. Responsibilities

	TM	PM	Dev team	Test team	Client
Acceptance test Documentation & Execution	X	X		X	X
System/Integration test Documentation & Exec.	X		X	X	
Unit test documentation & execution	X		X	X	
System Design Reviews	X	X	X	X	X
Detail Design Reviews	X	X	X	X	
Test procedures and rules	X	X	X	X	
Screen & Report prototype reviews			X	X	X
Change Control and regression testing	X	X	X	X	X

16. Schedule

Time has been allocated within the project plan for the following testing activities. The specific dates and times for each activity are defined in the project plan timeline. The persons required for each process are detailed in the project timeline and plan as well. Coordination of the personnel required for each task, test team, development team, management and customer will be handled by the project manager in conjunction with the development and test team leaders.

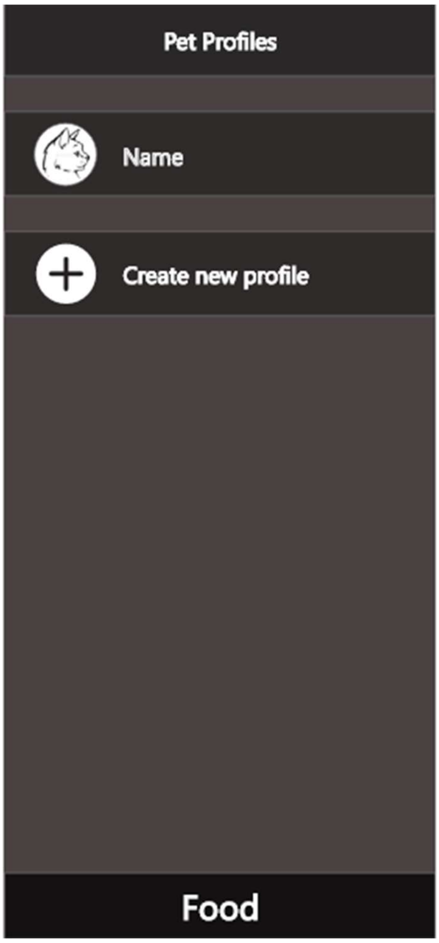
- Review of Requirements document by test team personnel (with other team members).
- Development of Master test plan by test manager and test with time allocated for at least two reviews of the plan.
- Review of the System design document by test team personnel.
- Development of System/Integration and Acceptance test plans by test manager and other essential personnel with time allocated for at least two reviews of the plans.

- Unit test time within the development process.
- Time allocated for both System/Integration and Acceptance test processes.

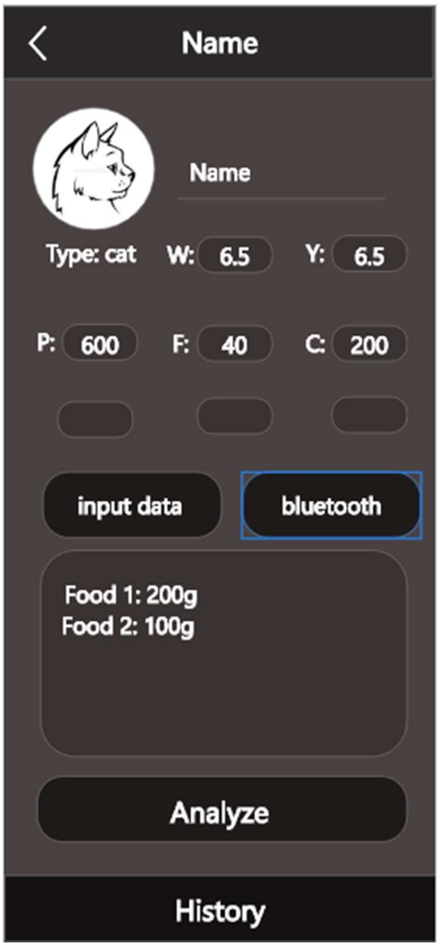
17. Approvals

CEO – Roman Mikhailovyn	

Wireframe Diagrams



Pet profile list page



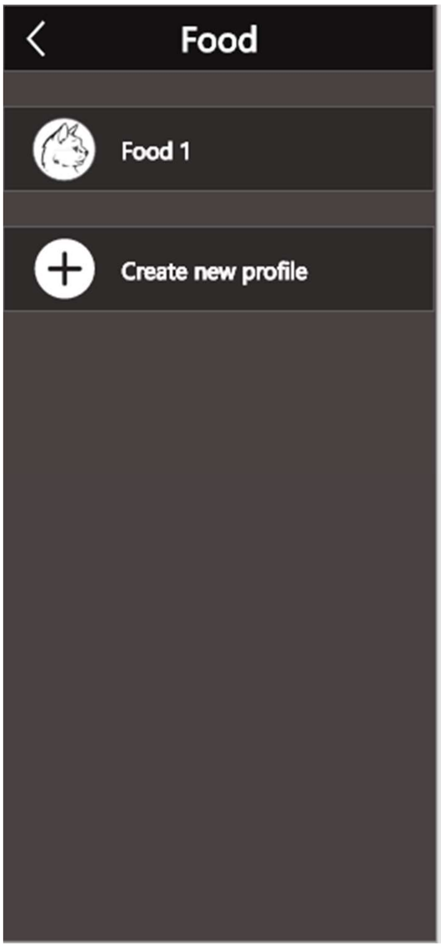
Name

Type: cat W: 6.5 Y: 6.5

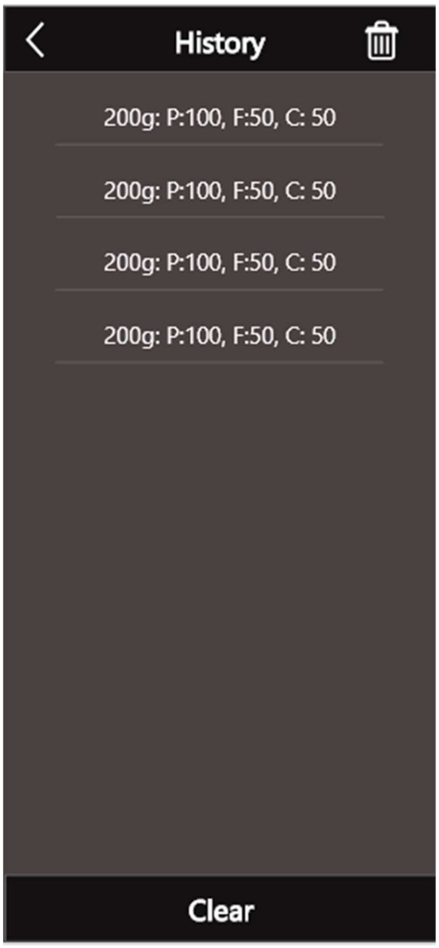
P: 600 F: 40 C: 200

Food 1: 200g
Food 2: 100g

Pet profile



Food list page



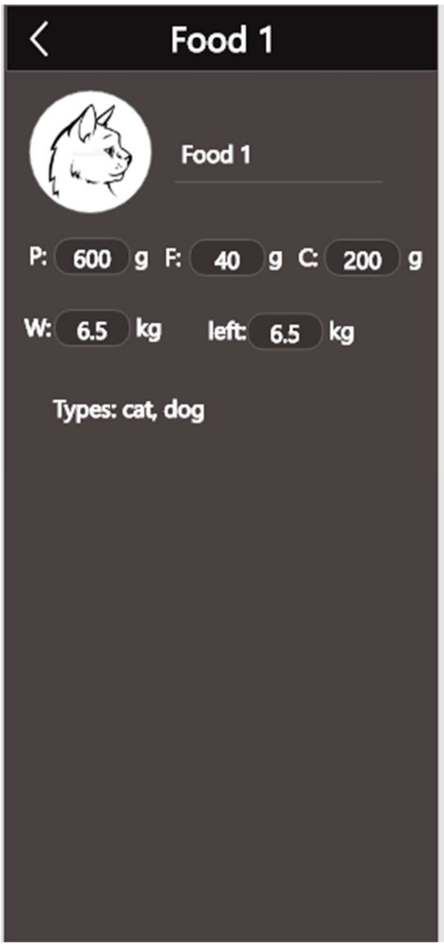
200g: P:100, F:50, C: 50

200g: P:100, F:50, C: 50

200g: P:100, F:50, C: 50

200g: P:100, F:50, C: 50

History page



Food 1

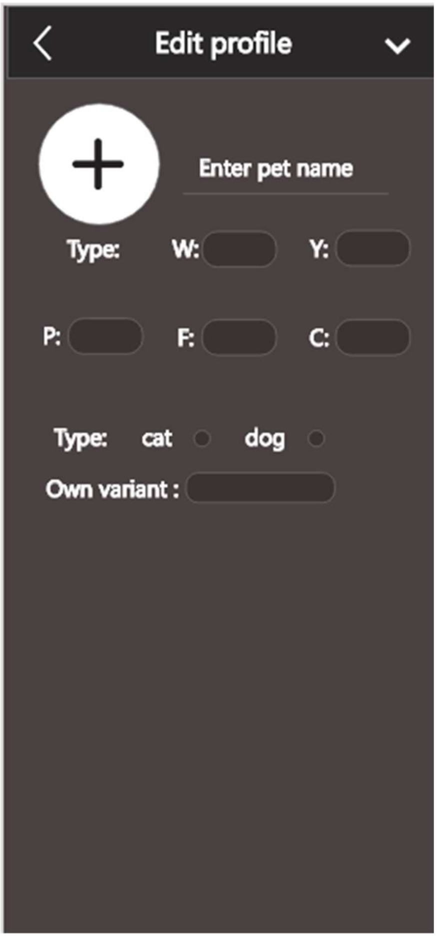
Food 1

P: 600 g F: 40 g C: 200 g

W: 6.5 kg left: 6.5 kg

Types: cat, dog

Food profile



Edit profile

Enter pet name

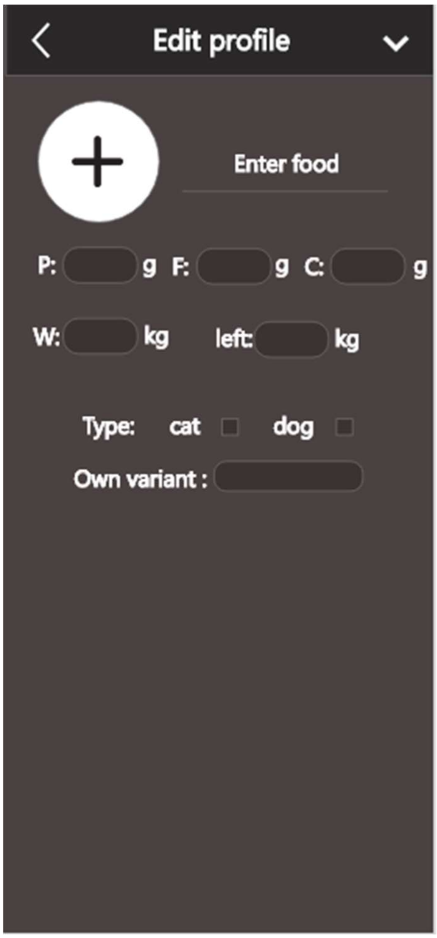
Type: W: Y:

P: F: C:

Type: cat ○ dog ○

Own variant :

Edit pet profile



Edit food profile

Висновок

У ході виконання розрахунково-графічної роботи були створені Software Requirement Specification, UML діаграми, Wireframe діаграми та тест план для проекту Automatic pet feeder App. З результатів роботи можна зробити висновок, що дані етапи проектування є дуже важливими і корисними, оскільки, виконавши їх, можна чітко побачити, що з себе буде являти продукт, який час потрібно закласти на його створення та що від нього очікувати користувачам. Однак, наявність SRS та решти перелічених пунктів ніяк не відміняє того, що етапи створення продукту можуть змінюватись, ставати довшими, виконувати інші цілі. Саме це закладене у підхід гнучкої розробки - швидка адаптація до змін. До того ж, не слід забувати, що одним із принципів Agile є “Працюючий продукт важливіше за докладну документаіцю”, тому не слід витрачати забагато часу на створення наддокладної документації, а починати створювати продукт.