UNIVERSITY OF ZAGREB

FACULTY OF ORGANISATION AND INFORMATICS

VARAŽDIN

Viktor Lazar, vlazar@foi.hr

Martina Šestak, msestak2@foi.hr

Goran Vodomin, gvodomin@foi.hr

Matej Vuković, mvukovic2@foi.hr

BLUETOOTH LE SHOWCASE

PROJECT DOCUMENTATION FOR SOFTWARE ANALYSIS AND DEVELOPMENT PROJECT

Varaždin, 2015.

UNIVERSITY OF ZAGREB

FACULTY OF ORGANISATION AND INFORMATICS

VARAŽDIN

Team number: T01

Team name: Heisenbug

Team members:

Viktor Lazar, 0108063551

Martina Šestak, 0016091250

Goran Vodomin, 0016092445

Matej Vuković, 0016094754

BLUETOOTH LE SHOWCASE

PROJECT DOCUMENTATION FOR SOFTWARE ANALYSIS AND DEVELOPMENT PROJECT

Mentors:

Dr.sc. Zlatko Stapić

Ivan Švogor, mag.inf.

Evolaris mentors:

Christian Adelsberger

Hermann Moser

Martin Schumann

Varaždin, January 2015.

Table of contents

1. Introduction

One of Evolaris' many projects is NFC bonus programme that was recently launched at the Shopping City Seiersberg located in Graz. Customers can, by using this application, register online and become „Friend of Seiersberg“. When registering every customer receives a traditional plastic card that contains a “Friend Chip”. When he comes to Shopping City Seiersberg he has to find a so called “Friends Kiosk” and place his “Friend Chip” over the terminal to sign in and begin using the application on his device. Every time he signs in, the user gets some points and vouchers which he can later spend in stores.

A “Friend Chip” which each customer has on his card supports Near Field Communication (hereinafter referred to as NFC) technology. NFC represents a form of contactless communication between devices. To start a communication session a user only needs to wave his device over another device that supports NFC technology. That device acts as a reader/interrogator and he is sending information to the passive user’s device using specific radio frequency.

Figure 1. NFC card,
Source: http://bestnfchardware.com/wp-content/uploads/2013/12/NFCBusinesscards.jpg

Currently Evolaris installed four terminals (readers) inside the shopping center so, whenever a user signs in with his card with NFC tag at that terminal, he can see how many points he has and he can also print his vouchers.

Even though the current system with NFC technology has proven to be successful in practice, they would like to use alternative technologies for the same functionality. This is the area our team will be dealing with; we will do some research about Bluetooth Low Energy (hereinafter referred to as Bluetooth LE) technology and build a mobile application that will mostly have the same functionality as Evolaris' current application, but instead, it will work using Bluetooth LE.

Everything regarding this project (project documentation, application code and notes from Scrum meetings) will be stored in our public Github repository available at https://github.com/MartinaSestak/AiR_BLE/tree/Phase-3/.

This document contains information about the implemented software development methodology and the way we implemented it in our project. Besides project documentation, this project characteristics are described in the technical (user requirements and product architecture specification, class diagram and generated source code specification) and user documentation (instructions for using the delivered application). In the writing of technical documentation we used a template published by FER (available at this address), for user requirements specification we used this template and for acceptance testing this template.

2. Bluetooth Low Energy Technology

Bluetooth Low Energy (also called Bluetooth Smart) is a technology that came public in 2010 and became popular because of lower battery power consumption compared to its predecessor Bluetooth. Thanks to this technology Bluetooth can also be used on smaller devices like watches and toys. It was first supported in mobile devices with Android version 4.3 (Android API 18), but nowadays it’s already available on iOS via iBeacon feature.

Figure 2. Bluetooth LE, Source:http://upload.wikimedia.org/wikipedia/en/thumb/2/20/Bluetooth_Smart_Logo.svg/1280px-Bluetooth_Smart_Logo.svg.png

Some features that made this technology stand out from classical Bluetooth technology are:

- Ultra-low peak, average and idle mode power consumption
- Ability to run for years on coin-cell batteries
- Lower implementation costs
- Multi-vendor interoperability
- Enhanced range

The communication between two devices is specified by so called GATT profile (Generic Attribute Profile), which is built on top of the Attribute Protocol (hereinafter referred to as ATT). This profile specifies the way small pieces of data (attributes) are exchanged between devices. Each attribute has its unique identifier (128-bit format for a string ID) and it can be either a characteristic or a service. A characteristic contains a single value and 0 or more descriptors used to describe that value and a service is a collection of characteristics.

For example, when an Android device communicates with a BLE device, we can say that it supports the central role of scanning and looking for an advertisement sent by BLE device, which then supports a peripheral role.

When these two devices establish a connection, they first need to exchange GATT metadata to define data that will be exchanged later. Depending on the nature of that data, a device taking part in this communication can be called a GATT server or a GATT client. In our example, if a BLE device continuously sends data to Android device, we can declare it as a GATT server, where the Android device is a GATT client receiving data from BLE device.

Compared to NFC technology, both Bluetooth LE and NFC define protocols for establishing a radio link and transferring data between two devices, where one device is considered to be a master (reader, scanner) and the other one is a slave (card, advertiser).

However, when it comes to range Bluetooth LE is better than NFC because it works over several meters of distance, while NFC will only work over a couple of centimeters. Also, a NFC communication session between two devices will start only after the user himself holds his device above the other device (reader) and a bidirectional data transfer will start immediately. Unlike that, Bluetooth LE doesn't require the user to do anything, communication between devices can start when the user's device is in reader's range.

However, data transfer begins after the "pairing" process which establishes the connection between devices (for instance, exchanging keys). There is another difference that can be noticed when comparing these technologies, and it's regarding devices' power supply. Bluetooth LE demands both devices to be powered with less power consumption compared to the standard Bluetooth. On the other side, when using NFC one device (usually NFC tag) can be passive and inactive until it starts communicating with the other device (reader).

Bluetooth LE has the potential to change the way costumers and retailers interact, but only for costumers with the latest smartphones and tablets.

This new technology will be used in our project. When costumers come to the shopping center, their mobile devices will detect Bluetooth LE devices (i.e. beacons), which will enable them to be identified as application users and receive points and upgrade their *Friend* status.

3. Software development methodology

During this project we decided to use agile software development methodology, which emphasizes team communication and continuous collaboration, functional software product and the flexibility to adapt to emerging business needs. Agile methodology includes methodologies like Scrum, Extreme Programming, Dynamic Systems Development etc.

Since Scrum is currently very popular and enables teams to dynamically plan everything regarding the project (releases, resources and functionalities etc.) in cooperation with stakeholders (in this case Evolaris and project mentors), we decided to use this software development methodology.

Since we decided to implement Scrum methodology, we needed to make it more suitable for our course's demands. For example, in the Scrum methodology original description, the Scrum Master is responsible for ensuring that the Scrum team will achieve all sprint goals. In our project the Scrum Master will occasionally need to be a part of the development team and will be responsible for delivering the appropriate project documentation. Duration of our iterations depends mostly on the deadlines that were set by our course mentors or on the availability of information we need to receive from Evolaris mentors.

Among many tools and services, we decided to use Microsoft Team Foundation Service provided by Team Foundation Server (hereinafter referred to as TFS) to implement Scrum methodology in our project. We used Visual Studio Online (Microsoft Team Foundation Service) available at https://martina-sestak.visualstudio.com/DefaultCollection/EvolarisBLE

You can access our project by signing in with following credentials:

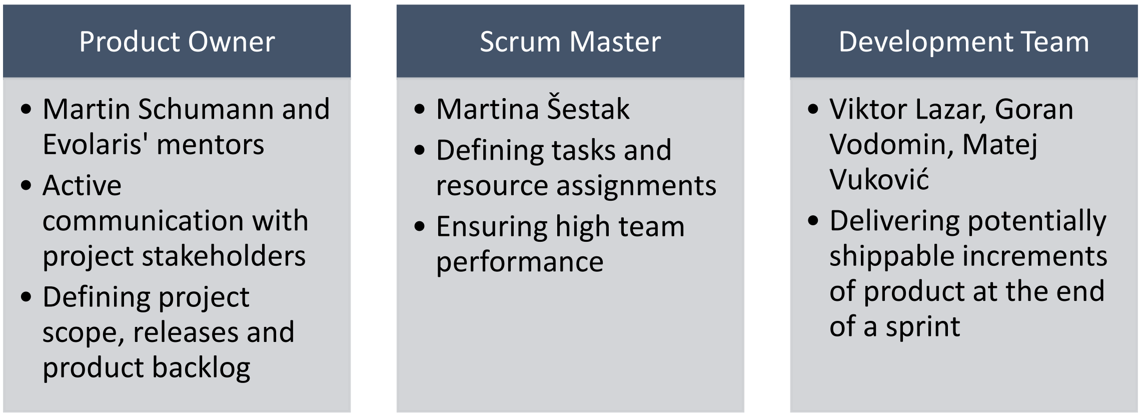
Email: **team.heisenbug@hotmail.com**

Password: **ProjektAiR**

Scrum methodology helped our team in better time manipulation and team members organization. We were able to rationally divide our entire project into several functionalities, and decide which one of them will be implemented in a specific sprint. Since we can see our sprint burndown charts, we were forced to follow our deadlines as planned, or at least improve ourselves if running late. By the end of it, we were able to publish our functional application using Scrum methodology.

3.1. Scrum team

According to Scrum, a Scrum team consists of 3 roles: Product Owner, Scrum Master and Development Team. Each of these roles is assigned to a team member, so every team member has some responsibilities shown in Figure 3. However, in accordance with courses' policies, each team member will be working on developing some part of the final application.



Product Owner

- Martin Schumann and Evolaris' mentors
- Active communication with project stakeholders
- Defining project scope, releases and product backlog

Scrum Master

- Martina Šestak
- Defining tasks and resource assignments
- Ensuring high team performance

Development Team

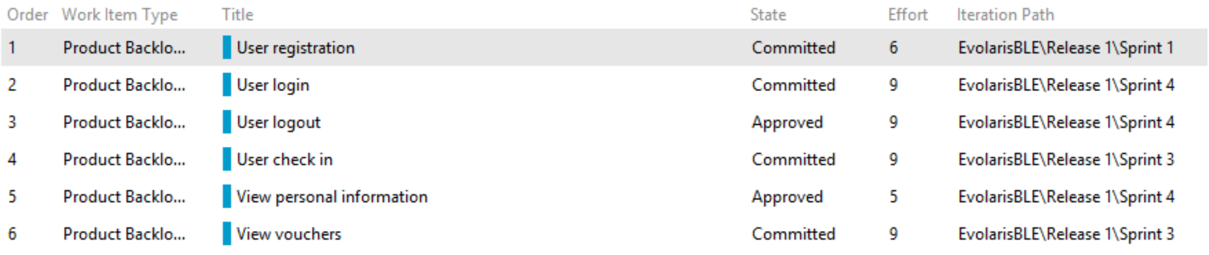
- Viktor Lazar, Goran Vodomin, Matej Vuković
- Delivering potentially shippable increments of product at the end of a sprint

Figure 3. Scrum roles and assignments

For this project we defined 4 sprints (iterations), so our first sprint called Sprint 1 started on 8th November 2014[1]. As mentioned before, all notes from Scrum team meetings can be found in our Github repository.

3.2. Product backlog

The Product owner defined and prioritized product backlog items (hereinafter referred to as PBI) that need to be implemented. On a product backlog refinement meeting the Scrum master defined which PBIs need to be implemented at the end of each sprint. Product backlog for this entire project is shown in Figure 4.

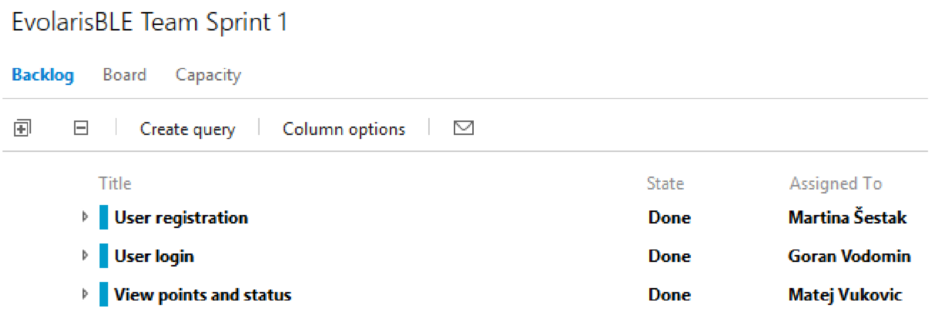


Order	Work Item Type	Title	State	Effort	Iteration Path
1	Product Backlo...	User registration	Committed	6	EvolarisBLE\Release 1\Sprint 1
2	Product Backlo...	User login	Committed	9	EvolarisBLE\Release 1\Sprint 4
3	Product Backlo...	User logout	Approved	9	EvolarisBLE\Release 1\Sprint 4
4	Product Backlo...	User check in	Committed	9	EvolarisBLE\Release 1\Sprint 3
5	Product Backlo...	View personal information	Approved	5	EvolarisBLE\Release 1\Sprint 4
6	Product Backlo...	View vouchers	Committed	9	EvolarisBLE\Release 1\Sprint 3

Figure 4. Product backlog

[1] Sprint 1 sprint started later because we didn't know our project scope before our meeting with Evolaris mentors, which took place on 31st October and because we were unofficially studying BLE technology earlier.

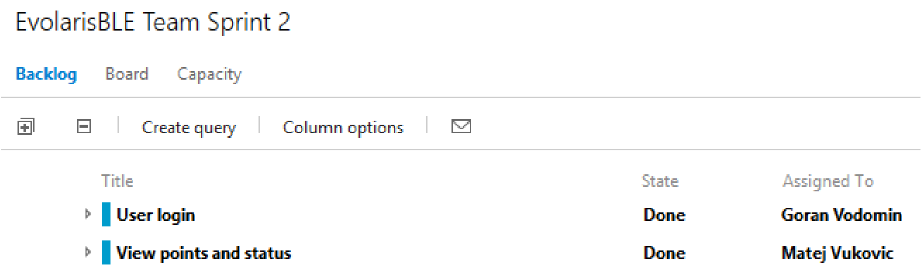
However, starting from Sprint1 our product backlog was continuously being refactored at the beginning of each sprint, so all versions of our product backlog can be found in following figures. In Sprint 1 our product backlog contained basic functionalities that our Evolaris mentors requested.



Title	State	Assigned To
User registration	Done	Martina Šestak
User login	Done	Goran Vodomin
View points and status	Done	Matej Vukovic

Figure 5. Product backlog for Sprint 1

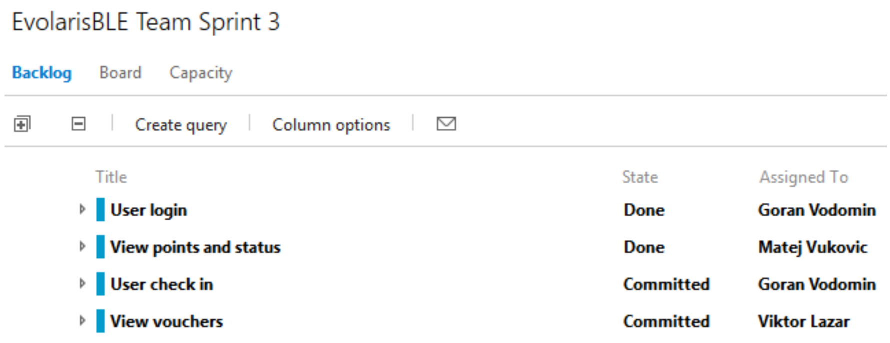
In Sprint 2, we decided to work on only two PBIs implementation: *User login* and *View points and status*.



Title	State	Assigned To
User login	Done	Goran Vodomin
View points and status	Done	Matej Vukovic

Figure 6. Product backlog for Sprint 2

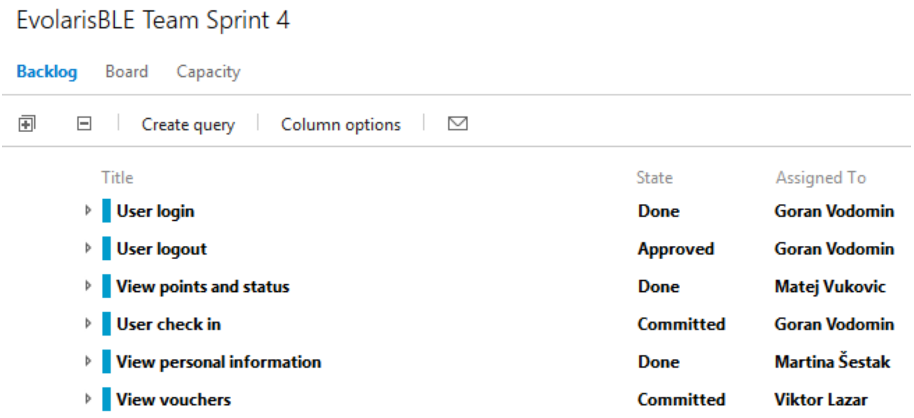
In Sprint 3 our product backlog was bigger because we started implementing new PBIs (*User check in* and *View* vouchers) and continued working on *User login* and *View point and status*.



Title	State	Assigned To
User login	Done	Goran Vodomin
View points and status	Done	Matej Vukovic
User check in	Committed	Goran Vodomin
View vouchers	Committed	Viktor Lazar

Figure 7. Product backlog for Sprint 3

Our product backlog for Sprint 4 consists of all PBIs and tasks we still needed to implement in Sprint 4.



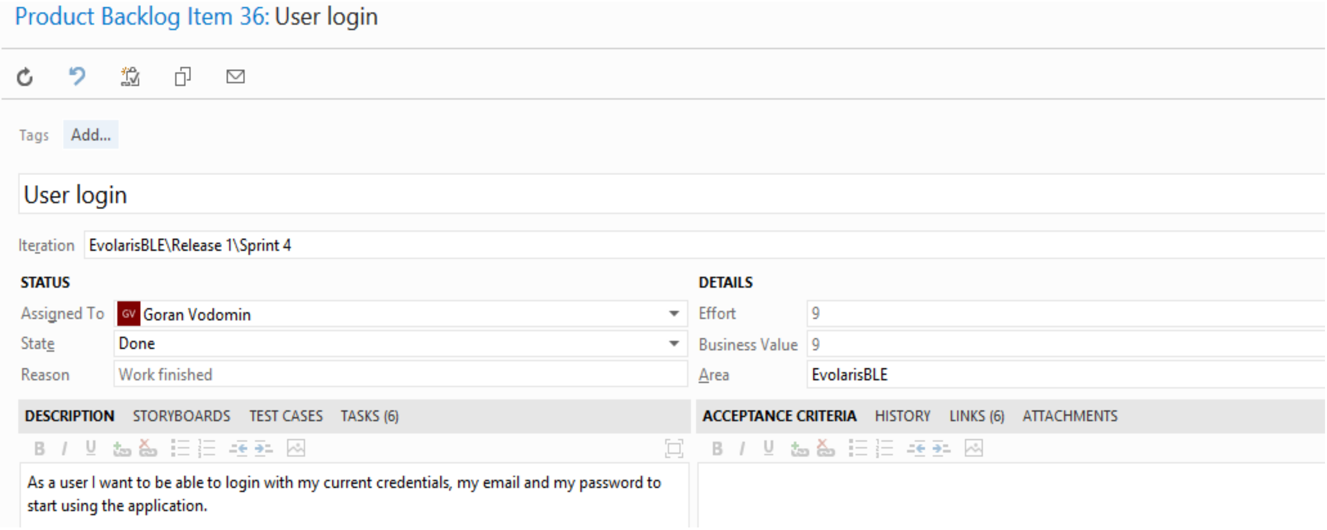
EvolarisBLE Team Sprint 4

Title	State	Assigned To
User login	Done	Goran Vodomin
User logout	Approved	Goran Vodomin
View points and status	Done	Matej Vukovic
User check in	Committed	Goran Vodomin
View personal information	Done	Martina Šestak
View vouchers	Committed	Viktor Lazar

Figure 8. Product backlog for Sprint 4

3.2.1. Product backlog item

Each PBI is declared to be in a state: *New*, *Approved*, *In progress*, *Commited* or *Done* and changes its state during the sprint. Also, when defining a PBI, a Product Owner sets its Business value and Effort measured in the number of hours the PBI will take to be implemented. He also needs to make a description of the PBI which contains a role, what this role wants to be able and why.



Product Backlog Item 36: User login

User login

Iteration EvolarisBLE\Release 1\Sprint 4

STATUS

Assigned To: GV Goran Vodomin

State: Done

Reason: Work finished

DETAILS

Effort: 9

Business Value: 9

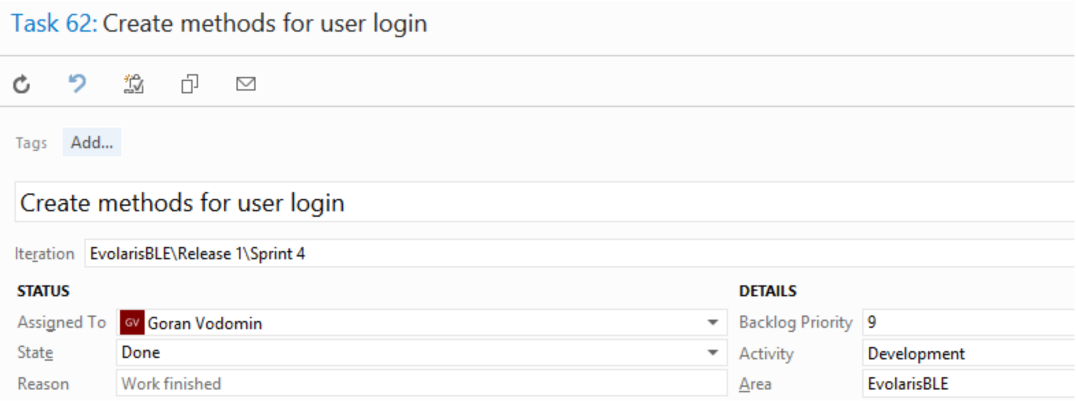
Area: EvolarisBLE

As a user I want to be able to login with my current credentials, my email and my password to start using the application.

Figure 9. Product backlog item sample definition

3.2.2. Tasks

On a sprint planning meeting the Scrum Master in agreement with the entire team analyzed each product backlog item and defined tasks for each PBI. Each task and PBI was assigned to a team member and he has to continuously update task’s status and remaining work time. Tasks can have a backlog priority in range 5 (not so important, can be implemented in the end) to 10 (very important, must be implemented as soon as possible). Figure 10 shows a task *Create methods for user login* sample definition, which is assigned to Goran Vodomin, its state is *Done* and has a backlog priority 9 (important).



Task 62: Create methods for user login

Tags Add...

Create methods for user login

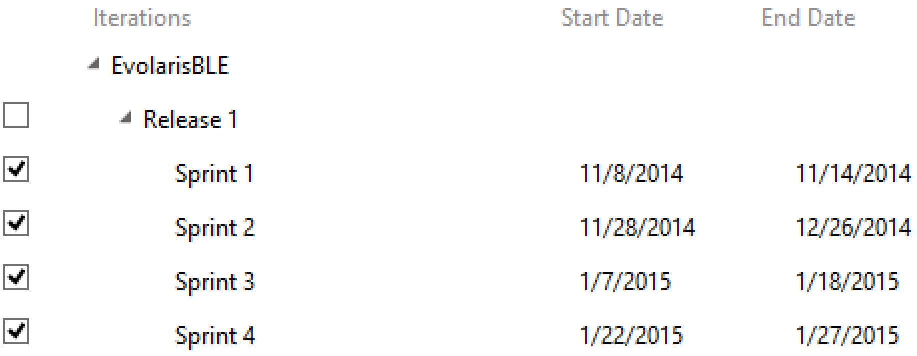
Iteration EvolarisBLE\Release 1\Sprint 4

STATUS		DETAILS	
Assigned To	GV Goran Vodomin	Backlog Priority	9
State	Done	Activity	Development
Reason	Work finished	Area	EvolarisBLE

Figure 10. Task "Create methods for user login" sample definition

3.3. Iterations

The Scrum team estimated that this project can be completed in 4 sprints with a duration of 1-4 weeks, as shown in Figure 5.

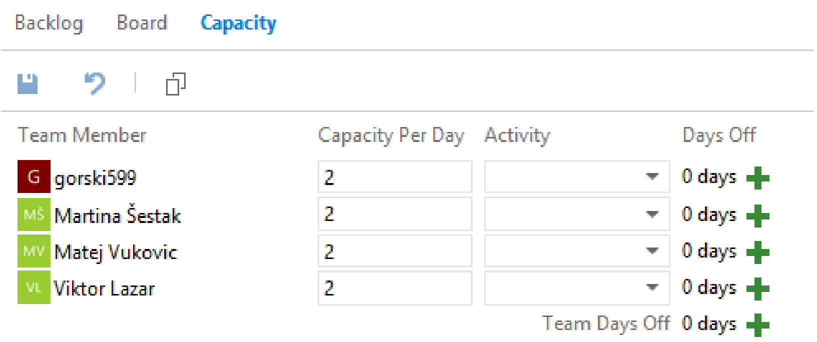


	Iterations	Start Date	End Date
	EvolarisBLE		
□	Release 1		
☑	Sprint 1	11/8/2014	11/14/2014
☑	Sprint 2	11/28/2014	12/26/2014
☑	Sprint 3	1/7/2015	1/18/2015
☑	Sprint 4	1/22/2015	1/27/2015

Figure 11. Iterations plan

3.4. Project team capacity

Each team member was assigned to a task while working 2 hours per day with no days off.



Team Member	Capacity Per Day	Activity	Days Off
G gorski599	2		0 days
MŠ Martina Šestak	2		0 days
MV Matej Vukovic	2		0 days
VL Viktor Lazar	2		0 days
		Team Days Off	0 days

Figure 12. Team capacity

3.5. Burndown charts analysis

As already mentioned, we can analyze our performance by looking at the generated burndown chart. Based on set project parameters (capacity, remaining work on tasks, effort etc.) the tool sets a line representing Ideal Trend. If our remaining work is bellow that line, this means all tasks are done on time, but if it's above the line, this means the project is late. Our project's final burndown chart for Sprint 1 is shown in Figure 11. It shows that the first sprint was at the beginning executed within timelines, which is opposite to a slight delay in the further course of this sprint.

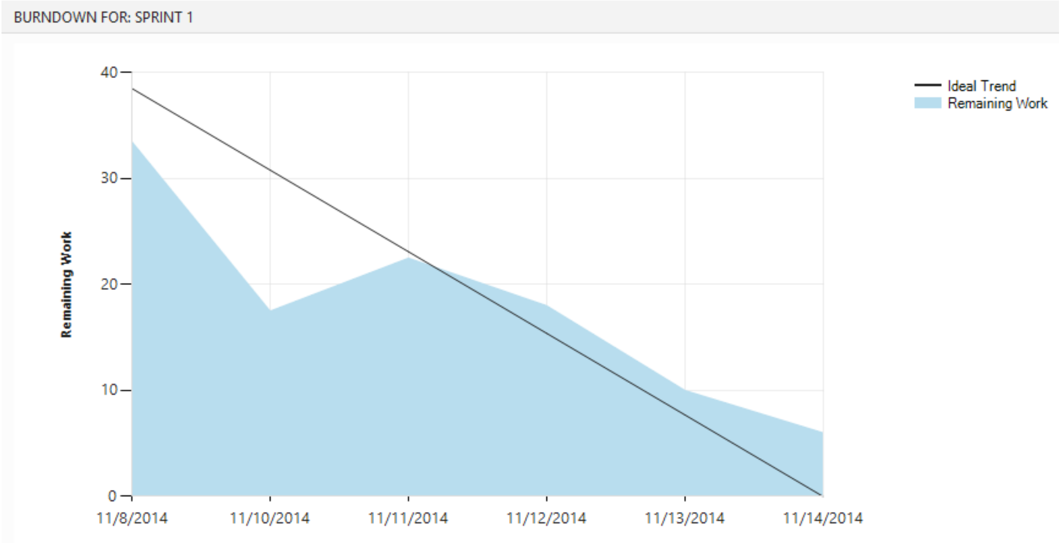


Figure 13. Burndown chart fort Sprint 1

If we take a look at the Sprint 2 burndown chart, we can notice that most of the tasks were completed at the beginning of the sprint so the project development was within time limits, except for the last week, but at the end of the sprint all tasks set for Sprint 2 sprint were still completed. Since Sprint 3 sprint is still running, the burndown chart for this sprint will be shown and analyzed in the final project Sprint.

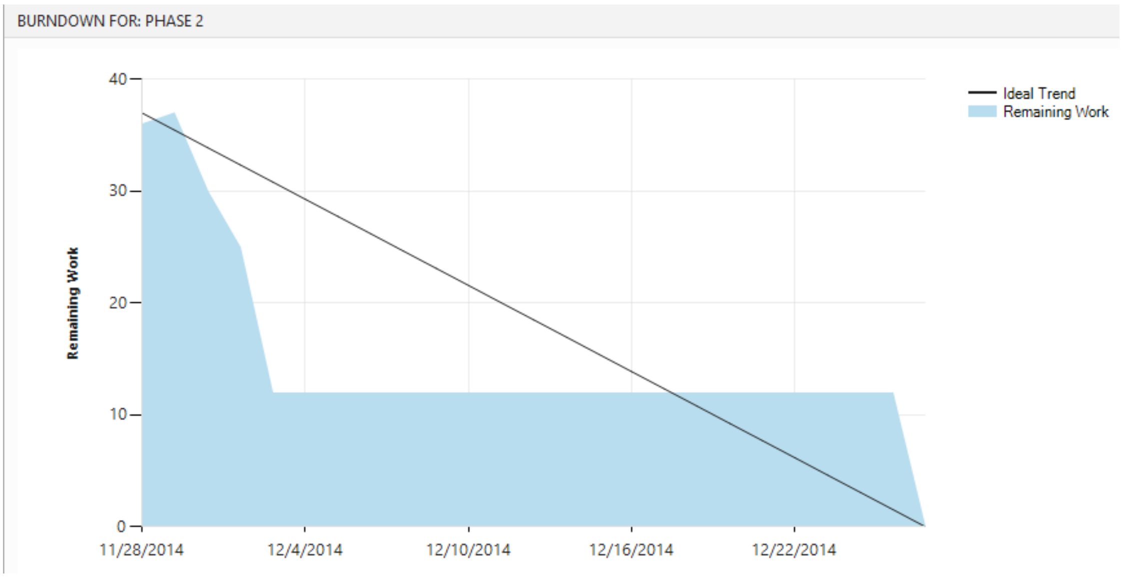


Figure 14. Burndown chart for Sprint 2 sprint

Sprint 3 burndown chart shows that half of this sprint was done within deadlines with no delay. However, at the half of this sprint the projects started to be late.

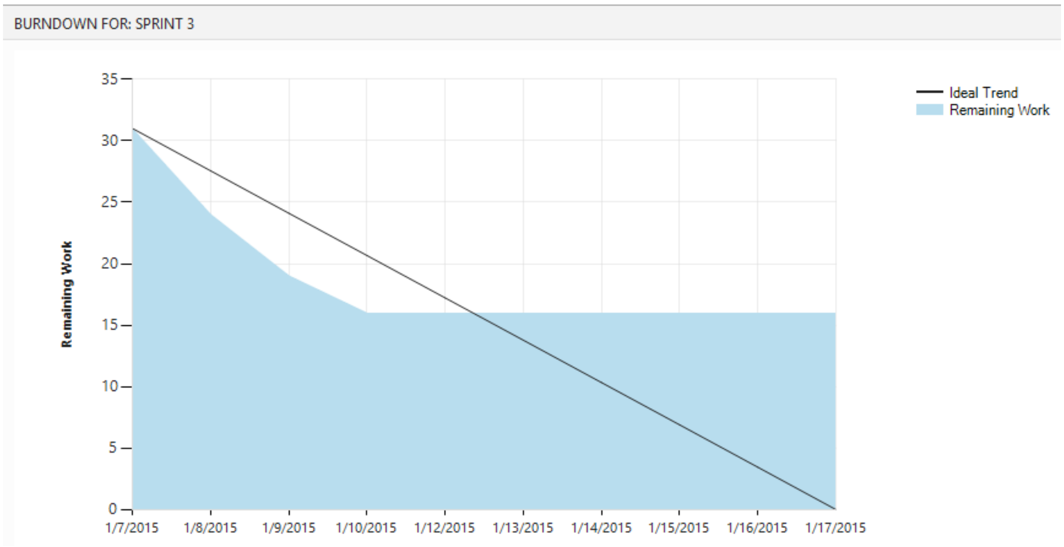


Figure 15. Burndown chart for Sprint 3

Finally, Sprint 4 burndown chart shows that the project was developed within defined deadlines and even ended one day before sprint end date.

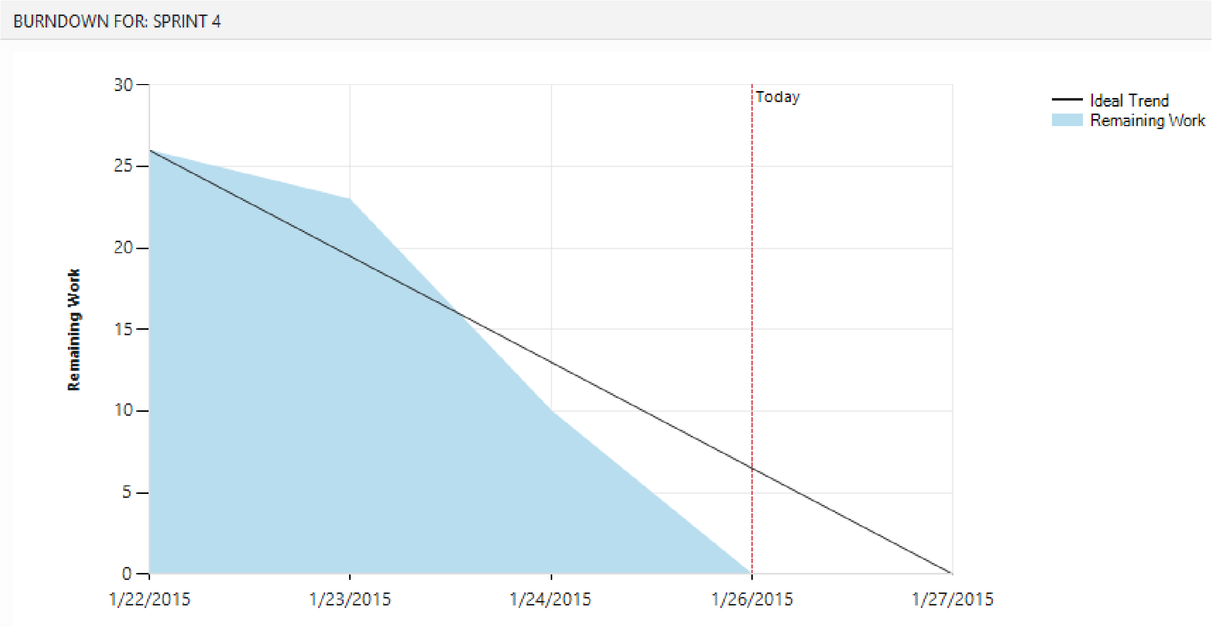


Figure 16. Sprint 4 burndown chart

4. Project references

1. T01-Bluetooth_LE_Showcase-Technical_documentation-Lazar,Šestak,Vodomin,Vuković.pdf
2. T01-Bluetooth_LE_Showcase-User_documentation-Lazar,Šestak,Vodomin,Vuković.pdf