



# **Project Name: Yoshi Final Report**

**Version 1.0**

Yoshi	Version: 1.0
Final Report	Date: 2015-01-11

## Revision History

Date	Version	Description	Author
2015-01-11	0.1	Initial Draft	Martin Anev
2015-01-13	1.0	Added Work Distribution section, Captions on tables and figures	Martin Anev

Yoshi	Version: 1.0
Final Report	Date: 2015-01-11

## Table of Contents

**LIST OF TABLES .....3**

**LIST OF FIGURES.....3**

**1. INTRODUCTION.....4**

    1.1 PURPOSE OF THIS DOCUMENT ..... 4

    1.2 DOCUMENT ORGANIZATION ..... 4

    1.3 INTENDED AUDIENCE ..... 4

    1.4 SCOPE ..... 4

    1.5 DEFINITIONS AND ACRONYMS..... 4

        1.5.1 *Definitions*..... 4

        1.5.2 *Acronyms and abbreviations* ..... 5

**2. OVERALL DESCRIPTION .....6**

    2.1 PROJECT DESCRIPTION ..... 6

    2.2 PARTICIPANTS ..... 6

        2.2.1 *Customer*..... 6

        2.2.2 *Supervisors*..... 6

        2.2.3 *Team* ..... 6

**3. ORGANIZATION AND COMMUNICATION .....7**

    3.1 PROJECT ORGANIZATION ..... 7

    3.2 TOOLS AND COMMUNICATION ..... 7

        3.2.1 *Organization tools* ..... 7

        3.2.2 *GitHub*..... 7

        3.2.3 *Communication tools*..... 7

**4. WORK DISTRIBUTION .....8**

**5. PROJECT RESULTS.....10**

    5.1 PROJECT SCOPES ..... 10

    5.2 MISSING FEATURES ..... 11

    5.3 POSSIBLE IMPROVEMENTS ..... 11

    5.4 DELIVERABLES ..... 11

**REFERENCES:.....14**

Yoshi	Version: 1.0
Final Report	Date: 2015-01-11

### List of Tables

Table 1. Definitions .....	5
Table 2. Acronyms and abbreviations .....	5
Table 3. Working hours .....	8
Table 4. Use Case: Compute the community.....	10
Table 5. Use Case: Visualize computed characteristics.....	11
Table 6. Deliverables.....	13

### List of Figures

Figure 1. Working hours of members .....	7
Figure 2. Weeks Load.....	8
Figure 3. Work distribution .....	8

Yoshi	Version: 1.0
Final Report	Date: 2015-01-11

## 1. Introduction

### 1.1 Purpose of this document

The purpose of this document is to give an overview of the project delivered by Team Yoshi in the Distributed Software Development course done simultaneously in 'Politecnico di Milano' situated in Milan, Italy and 'Mälardalen University' situated in Västerås, Sweden. This document is a final document of the course, so it gives short recap of all the documents written to this point.

### 1.2 Document organization

The document is organized as follows:

- Section 1, *Introduction*
- Section 2, *Overall Description*, describes the project and participants
- Section 3, *Organization and communication*
- Section 4, *Work Distribution*
- Section 5, *Project results*

### 1.3 Intended Audience

The intended audience is:

- The supervisors of the project
- Yoshi Team
- Any developer with interest to continue or improve the project

### 1.4 Scope

The document summarizes the development process of the project 'Yoshi'. It includes a short background of the project which gives an overview of the project vision, requirements and team organization that are followed by the project plan. In the chapter 'Project results' is presented overview of the software developed for this course, documentation and possible improvements.

### 1.5 Definitions and acronyms

#### 1.5.1 Definitions

Basic definitions of the document terms are defined in Table 1.

Yoshi	Version: 1.0
Final Report	Date: 2015-01-11

<b>Keyword</b>	<b>Definitions</b>
Community	Social unit of any size that shares common values.
Open Source Community	Community that develops open source software.
Compute a Community	The action of observing a community, measuring metrics for Social Communities (e.g. collaboration between community members, common projects, list of followers of the members, collaborators, contributors to projects). The outcome of the action is a decision – what type is the observed community.
Visualize a community	The action of observing the output of the decision “What type is a community?”. The deliverable is text and images.
Software adaptor	In software engineering, the adapter pattern is a software design pattern that allows the interface of an existing class to be used from another interface.
Eclipse plug-in	Plugins are the smallest deployable and installable software components of ‘Eclipse’ software developing tool.
Community Decision Tree	Decision tree defining what type is an observed social community based on some characteristics. The Decision Tree is discussed in the document ‘Uncovering Latent Social Communities in Software Development’ [4].

**Table 1. Definitions**

1.5.2 *Acronyms and abbreviations*

<b>Acronym or abbreviation</b>	<b>Definitions</b>
W	Week
G	Goal
FR	Functional Requirement

**Table 2. Acronyms and abbreviations**

Yoshi	Version: 1.0
Final Report	Date: 2015-01-11

## 2. Overall description

This section describes the overall description of the project.

### 2.1 Project description

The project has two aims:

- Modify the existing “Yoshi” product.
- Develop missing components of the product.

A deliverable of the project is software product. Product’s name is “Yoshi Vis” and it is dependent on “Yoshi” [1] (Eclipse plug-in), providing a visualization and computation layer on top of “Yoshi”. In order to provide the adaptation, proper adaptors should be assigned to the output of “Yoshi”. Design and implementation of these adaptors will also be covered during this project.

### 2.2 Participants

#### 2.2.1 Customer

Damian A. Tamburri is a Ph.D. researcher at Vrije Universiteit - Amsterdam. Damian is the customer of Yoshi project. The customer wants a software for supporting social community awareness in open-source. He provided the source code of “Yoshi”. “Yoshi” has been built by Alexandra Leta, student in Information Sciences, Vrije Universiteit - Amsterdam for her graduation thesis project. The code was sent without any proper documentation. The team received only the thesis work of Alexandra Leta[9]. Yoshi, is an analytic software for open-source communities, helping different users to better understand open-source community and to get a good understanding for research and for practice. The team for the project did not succeed in running the prototype. A meeting with the customer was scheduled, but the prototype was still not being able to run. According to the output files, the Yoshi prototype has worked before and it had already made some output in a text form (i.e. ‘data.out’ file).

Email: [d.a.tamburri@vu.nl](mailto:d.a.tamburri@vu.nl)

#### 2.2.2 Supervisors

##### 2.2.2.1 Local Supervisor

Raffaella Mirandola

Email: [raffaella.mirandola@polimi.it](mailto:raffaella.mirandola@polimi.it)

##### 2.2.2.2 Remote Supervisor

Juraj Feljan

Email: [juraj.feljan@mdh.se](mailto:juraj.feljan@mdh.se)

#### 2.2.3 Team

##### 2.2.3.1 Group Members

- Martin Anev – Email: [martianev@gmail.com](mailto:martianev@gmail.com)
- Félix Baylac-Jacqué – Email: [baylac.felix@gmail.com](mailto:baylac.felix@gmail.com)
- Yuxing Chen – Email: [axinggu@qq.com](mailto:axinggu@qq.com)
- Rizwan Khalid – Email: [rkd14001@student.mdh.se](mailto:rkd14001@student.mdh.se)
- Hossein Sedighizadeh – Email: [hossein.sedighi161@gmail.com](mailto:hossein.sedighi161@gmail.com)

##### 2.2.3.2 Roles

- Scrum master: Hossein Sedighizadeh
- Product owner: Martin Anev
- Development team: Every member of the team

Yoshi	Version: 1.0
Final Report	Date: 2015-01-11

### 3. Organization and Communication

#### 3.1 Project Organization

The team has decided to use the agile methodology and in particular the SCRUM framework, in order to manage the project. This project was divided into 5 sprints. At the beginning of each sprint, the backlog was discussed on the blackboard. Each sprint was lasting two weeks. During these two weeks, two to three meetings were being held, in order to setup and debrief the sprint: the first one during the first day, the last one during the last day. Sometimes the last week of one sprint was also the first of the next sprint. The tasks uncompleted during each sprint were reported to the next sprint backlog. The tool used for managing the tasks and assignments was Trello[2]. Each Sunday the group members were supposed to upload their summary week report in a Dropbox[3] folder visible to all team members. The scrum master was completing the week summary report into the same folder. Despite some delays and absences the team was able to understand who was doing what during the past week effectively. The decision of the SCRUM framework was driven because of the ambiguity in the project requirements. Using SCRUM the team managed to prevent of developing a product that does not fulfil the customer needs. The team benefited from the agile methodology as unexpected leaving of team members occurred during different times of the project.

#### 3.2 Tools and Communication

##### 3.2.1 Organization tools

###### 3.2.1.1 Trello

In order to manage this project using the SCRUM framework, the team decided to use Trello as blackboard. This website was very efficient as it contains cards – related for tasks which could be moved among different lines (e.g. the card “Final Report Document” is moving among lines “To Do”, “Doing”, “Done”). Each card can be assigned to specific member, and with a deadline. Notifications are coming when the deadline is approaching and a tagging technique was used to notify person for a comment on specific task. The team set a reminder (through Google Calendar and the Trello integration with emails) to notify weekly the members for the submission of the week reports.

###### 3.2.1.2 Dropbox

For file managing the team members were using Dropbox. Separate folders were assigned for documentation, related papers, and week reports.

###### 3.2.1.3 Google Docs

For presentations and temporary documents that needed to be agile and editable by all team members simultaneously the team was using Google Docs [4].

##### 3.2.2 GitHub

For implementation of the software product was used GitHub[6]. GitHub’s distributed revision control was everything that the team was needing for the implementation purposes.

##### 3.2.3 Communication tools

###### 3.2.3.1 Emails

Emails were used for communication with the team supervisors and with the customer.

###### 3.2.3.2 Skype

The team communicated with the customer and the project supervisors using instant messaging Skype [7].The team was using it for instant messaging for internal communication purposes (meetings, group coding, etc.) and for informal communication.



#### 4. Work distribution

On Table 3 is presented the working data from the team members, according to their reports.

Member	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	TOTAL
Hossein Sedighizadeh	15	12	9	17	10	16	12	12	8	5	2	4	122
Félix Baylac-Jacqué	7	8	14	12	10.5	6	3.1	12	6	0	2	13.5	94.1
Martin Anev	8	12	15	13	12	11	12	12	7	3	1	10	116
Yuxing Chen	7	8	17	10	8	11	16	14	5	1	1	9	107
Rizwan Khalid	15	9	9	8	5	14	12	15	11	11	10	7	126
<b>Total</b>	<b>52</b>	<b>49</b>	<b>64</b>	<b>60</b>	<b>45.5</b>	<b>58</b>	<b>55.1</b>	<b>65</b>	<b>37</b>	<b>20</b>	<b>16</b>	<b>43.5</b>	<b>565.1</b>

Table 3. Working hours

On Figures 1 and 2 are presented the data in graphical forms in weeks and working hours.

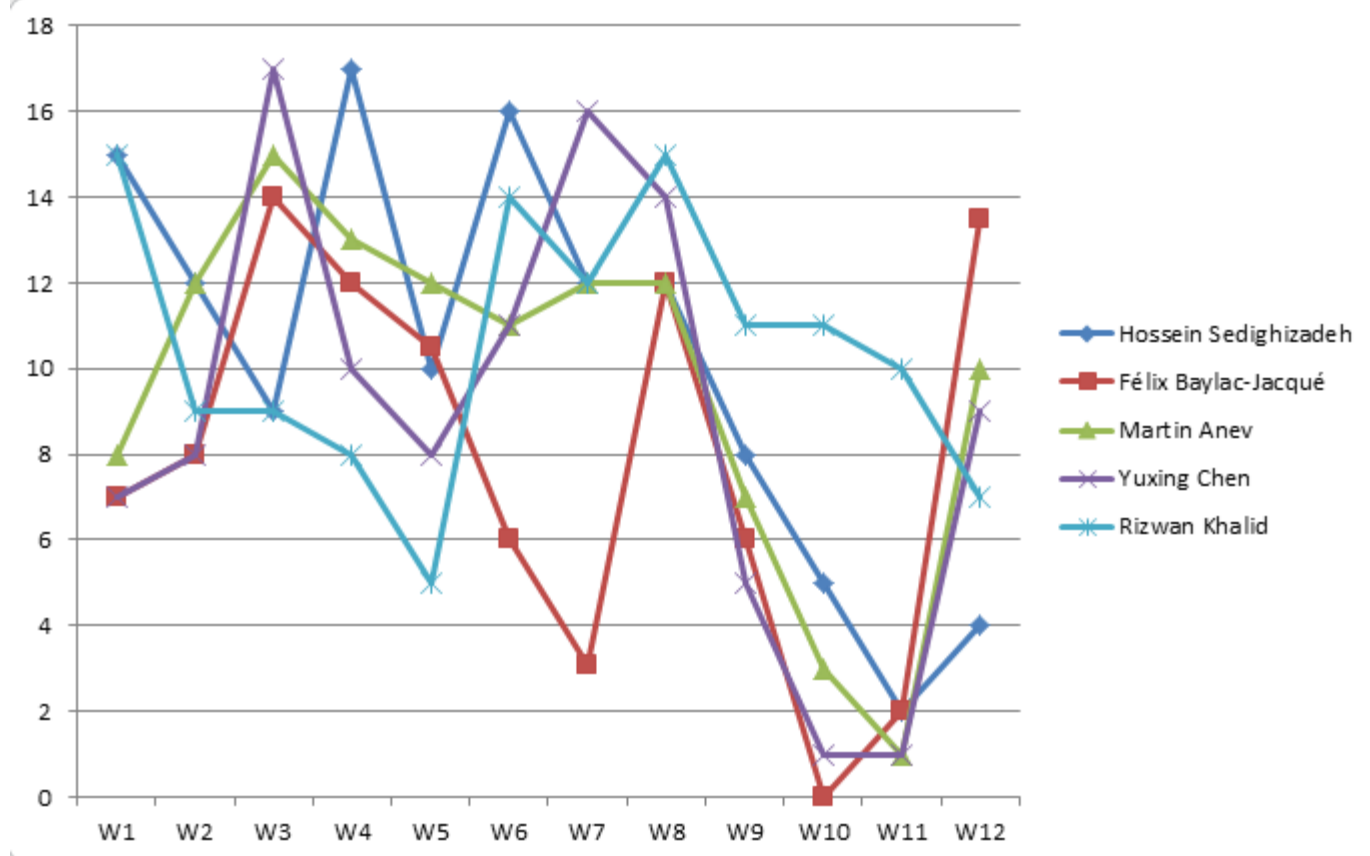
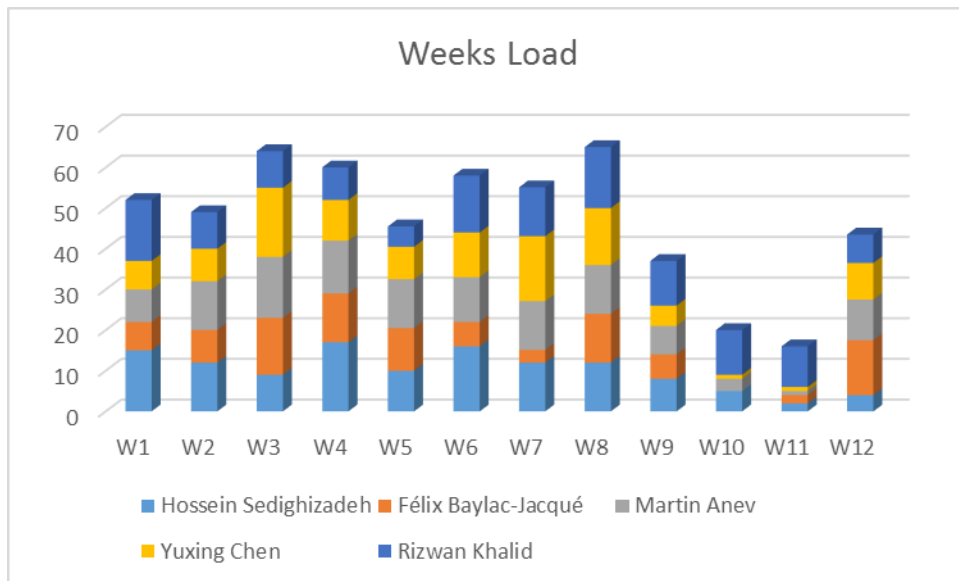
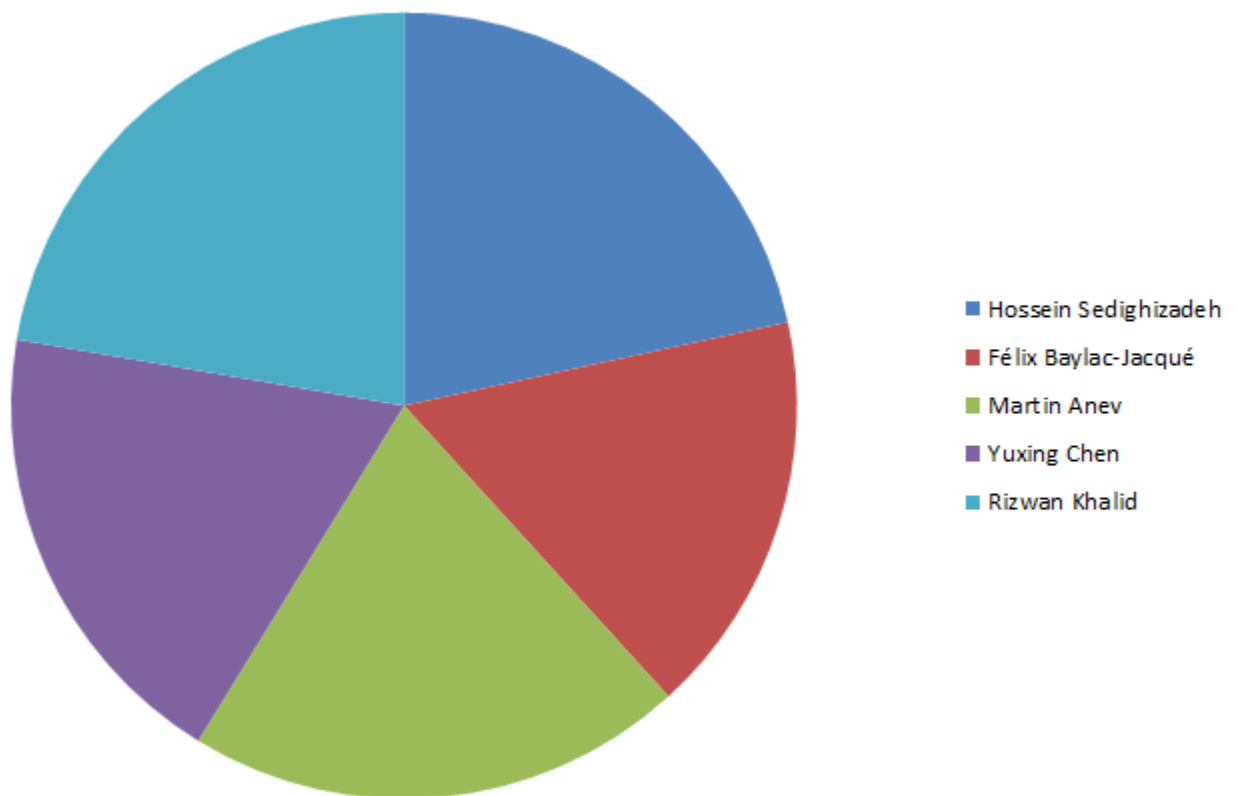


Figure 1. Working hours of members



**Figure 2. Weeks Load**

On Figure 3 is represented the work distribution among members.



**Figure 3. Work distribution**

Yoshi	Version: 1.0
Final Report	Date: 2015-01-11

## 5. Project results

### 5.1 Project Scopes

The project was delivered successfully.

The aims of the project:

- Modify the existing “Yoshi” product.
  - Develop missing components of the product.
- ✓ Have been achieved.

The goals of the project:

- [G0] Allow users to observe community by gathering organizational characteristics.  
 [G1] Allow users to measure community social and organizational characteristics.  
 [G2] Allow users to study type and characteristics against performance metrics.  
 [G3] Allow users to understand the characteristics of an observed community.  
 [G4] Format the current output of “Yoshi”.  
 [G5] Define the type of the observed community based on the metrics.  
 [G6] Provide output with the defined social types and motivate the decision.
- ✓ Have been achieved.

The Functional Requirements:

- [FR1] The current output of Yoshi should be prepared for adaptation.  
 [FR2] The product should be able to provide proper adapter for the formatted output of Yoshi.  
 [FR3] The product should be able to take decision what type is an observed social community based on the received metrics from “Yoshi”.  
 [FR4] The product should be able to visualize as output the formatted output of Yoshi.  
 [FR5] The product should be able to visualize as output the evaluated decision of the community type in text form.  
 [FR6] The product should be able to visualize as output motivation of the taken decision.  
 [FR7] The product should be able to visualize as output the community type on the Community Decision Tree in graphic form.
- ✓ Have been achieved.

The use cases:

Name	Compute the community
Actors	User
Entry conditions	The user select specific community to compute
Flow of events	<ol style="list-style-type: none"> <li>1. The user starts the system</li> <li>2. System provides choices for community</li> <li>3. The user selects the community to compute</li> <li>4. The user clicks “compute community”</li> <li>5. The system shows the result</li> </ol>
Exit conditions	The user quits the activity
Exceptions	No exception

**Table 4. Use Case: Compute the community**

Yoshi	Version: 1.0
Final Report	Date: 2015-01-11

Name	Visualize computed characteristics
Actors	User
Entry conditions	After the system computed the system shows the result page
Flow of events	<ol style="list-style-type: none"> <li>1. The user starts the system</li> <li>2. System provides choices for visualizing already computed results</li> <li>3. The user selects the desired option <ol style="list-style-type: none"> <li>3.1 The user clicks “visualize the result”</li> </ol> </li> <li>4. The system visualizes the result</li> </ol>
Exit conditions	The user quits the activity
Exceptions	No exception

**Table 5. Use Case: Visualize computed characteristics**

✓ Are verified.

Although, different names of the buttons and the layout is provided, the use cases follow the same flow and deliver the same output.

The aims were fulfilled within the time limits. The customer is content with the delivered product and confirms the objectives have been met.

## 5.2 Missing features

*It was not team’s task to deliver runnable (stable) version of the “Yoshi” prototype. Up to date it is not yet runnable.* Besides this, the features and requirements are met.

## 5.3 Possible improvements

- Running the “Yoshi” prototype would be major improvement for the whole system. Re-implementing “Yoshi” from scratch should be evaluated as there is significant chance to be easier task. In case of such option, the team would recommend using the same technologies: Python version 3.3 or higher and sticking to the current architecture. The work on this project was managed so that the components of “Yoshi” and “Yoshi Vis” (refer to the Design Documentation [8]) are replaceable.
- Significant improvements could be achieved by improving the thresholds and methodologies of defining a type of the community. Deep research for the community types should be done for this purpose.
- Possible improvements in the graphical user interface could be applicable as well.

## 5.4 Deliverables

The deliverables are the results of the project that are delivered to the Customer, Supervisor and/or the Team itself. Table 6 below represents all the expected deliverables of the project and the expected date that they should be handed to the specified recipient.

Yoshi	Version: 1.0
Final Report	Date: 2015-01-11

ID	Deliverable	Responsible	Recipient	Deliver Date	Actual Date	+/-	Comments
D01	Team Logo	Felix	T	2014-10-27	2014-10-27	0	
D02	Project Vision & Project Plan Presentation	Yoshi Team	C,S,T	2014-10-28	2014-10-28	0	
D03	Project Plan Document (v.1)	Yoshi Team	S,T	2014-11-03	2014-11-03	0	
D04	Requirement, Design Presentation	Yoshi Team	C,S,T	2014-11-11	2014-11-11	0	
D05	Project Requirements Document (v.1)	Martin, Yuxing	C,S,T	2014-11-14	2014-11-10	+4	Version 1.0 was delivered on 2014-11-03, version 1.1 on 2014-11-10
D06	Design Description Document (v.1)	Rizwan, (Johan – not part of the team since Week 2)	C,S,T	2014-11-14	2014-11-21	-7	
D07	Status Report Presentation	Yoshi Team	C,S,T	2014-11-18	2014-11-18	0	
D08	Project Requirements Document (v.2.0)	Martin, Yuixng	C,S,T	2014-11-26	2014-11-26	0	
D09	Design Description Document (v.2.0)	Rizwan	C,S,T	2014-11-30	2014-12-01	-1	
D10	Formatted <b>data.out</b> file from “Yoshi”	Hossein	T	2014-12-05	2014-12-05	0	
D11	Alpha Prototype Presentation	Yoshi Team	C,S,T	2014-12-02	2014-12-02	0	
D12	Build input for the algorithm	Yuxing	T	2014-12-16	2014-12-16	0	
D13	Build the algorithm	Yuxing	T	2014-12-14	2014-12-14	0	
D13	Build the output of Yoshi Vis	Felix	T	2014-12-14	2014-12-14	0	
D14	Include the images in the output	Rizwan	T	2014-12-14	2014-12-14	0	
D15	Beta Prototype Presentation	Yoshi Team	C,S,T	2014-12-16	2014-12-16	0	

Yoshi	Version: 1.0
Final Report	Date: 2015-01-11

<b>D16</b>	Project Plan Document (v.1.1)	Felix, Hossein, Martin	C,S,T	2015-01-16	2014-01-05	+11
<b>D17</b>	Design Description Document (v.3.0)	Martin	C,S,T	2015-01-16	2014-12-28	+19
<b>D18</b>	Acceptance Test Plan Document (v.1.0)	Rizwan	C,S,T	2014-12-31	2014-12-30	+1
<b>D19</b>	Test Report	Yuxing	C,S,T	2014-01-12	2014-01-12	0
<b>D20</b>	Final Project Presentation	Yoshi Team	C,S,T	2015-01-13	2014-01-13	0
<b>D21</b>	Final Report Document (v.1.0)	Martin	C,S,T	2015-01-16	2014-01-16	0
<b>D22</b>	Final product	Yoshi Team	C,S,T	2015-01-23	2015-01-11	+12
<b>D23</b>	Final version of documents	Yoshi Team	C,S,T	2015-01-16	2015-01-16	0

**Table 6. Deliverables**

Yoshi	Version: 1.0
Final Report	Date: 2015-01-11

## References:

- [1] Yoshi - <https://github.com/maelstromdat/YOSHI>
- [2] Trello - <https://trello.com/>
- [3] Dropbox – <https://dropbox.com/>
- [4] Google Docs – <https://docs.google.com/>
- [5] LucidChart - <https://www.lucidchart.com/>
- [6] GitHub - <https://github.com/>
- [7] Skype - <http://www.skype.com/en/>
- [8] Yoshi Vis Documentation - <https://github.com/NinjaTrappeur/yoshi-viz/blob/master/YoshiViz/Documentation/Design%20Description%20v.3.pdf>
- [9] ‘Discovering Open-Source Community Types: An Automated Approach’ – A. Leta