

# Project Plan

## Group: Canvas

**Software Engineering Project  
Department of Computer Science  
University of Helsinki**

07/06/2005

Document version: 04

***Project Members:***

Ahlberg Peter Johan,  
Duku-Kaakyire Michael,  
Karppinen Tony Henrik,  
Lamsal Pragya,  
Välimäki Niko Petteri

***Instructor:***

Raine Kauppinen

***Supervisor:***

Juha Taina

## Table of Contents

1. Introduction.....	3
2. Organization.....	3
2.1 Description of the interest groups.....	3
2.2 Areas of responsibility.....	3
3. General description of the application.....	4
4. Size estimate.....	4
5. Project schedule.....	5
6. Working methods.....	8
6.1 Communication.....	8
6.2 Reporting.....	8
6.3 Tools.....	8
7. Risk Analysis.....	9

## **1. Introduction**

This is a project for design and development of a diagram-drawing software under the course named Software Engineering Project at the department of Computer Science at University of Helsinki.

The aim of this project is to develop a functional application that can realize the core requirements of the customer, while leaving a room for extension of the application in the future by other developers.

The homepage of this project group is <http://www.cs.helsinki.fi/group/canvas/>.

## **2. Organization**

### ***2.1 Description of the interest groups***

The customer of this project is Inkeri Verkamo, department of Computer Science, University of Helsinki.

The instructor of the project is Raine Kauppinen, department of Computer Science, University of Helsinki.

The supervisor of the project is Juha Taina, department of Computer Science, University of Helsinki.

Developers of the application are students of the Computer Science department at University of Helsinki. They are:

Ahlberg Peter Johan,  
Duku-Kaakyire Michael,  
Karppinen Tony Henrik,  
Lamsal Pragya,  
Välimäki Niko Petteri

### ***2.2 Areas of responsibility***

The developers are responsible for collection of requirements, design of the application, implementation of the application, integration and testing of the application, and installing the application on the customers system. The developers are also responsible of make sure that the requirements collected are consistent with the need of the customer. For this, the developers need to develop a prototype of the application and discuss with the customer about the validity of the requirements produced in the prototype.

The instructor of the project is responsible for monitoring the progress of the project group and give comments and suggestions about the project whenever necessary.

The organizer of the project is responsible for organizing the project. For this project group, the organizer is also the technical assistant who answers the questions of the customer.

Pragya Lamsal is the project leader. Peter Ahlberg and Tony Karppinen are mainly responsible for the requirements collection during the requirements engineering phase. Basically, all group members are responsible for the requirement engineering, design, implementation and the testing phase work. Work division in each phase is done as necessary. The homepage of the project group is updated by any of the members, as necessary.

### 3. General description of the application

The application to be developed by this project is a software used for drawing diagrams. The application will provide a set of basic elements like a box (rectangle), a circle, a connecting line, etc. The application will allow the users of the application to define new element types, define syntactic rules for the elements.

The description of the application given by the customer is: Within research in software engineering, there is often need to draw various kinds of diagrams, and the purpose of this project is to design and implement a tool for this purpose. To support drawing of multiple kinds of diagrams, the tool should be as generic as possible, and allow defining sets of diagram elements and rules for each type of diagram that the user wants to draw. Typical types of diagrams that the tool could be used for are queueing network models, dataflow diagrams or state diagrams.

### 4. Size estimate

Measurement parameter	Count		Weighting factor		Total
no. of user inputs	15	x	6 (complex)	=	90
no. of user outputs	17	x	5 (average)	=	85
no. of user inquiries	1	x	4 (average)	=	4
no. of files	2	x	7 (simple)	=	14
no. of external interfaces	3	x	7 (average)	=	21
<b>Total-count</b>				=	<b>214</b>

Function Point (FP) = Total-count x (0.65 + 0.01 x sum(Fi))

Fi (i = 1 to 14) are complexity adjustment values based on the responses to questions given below.

Each factor (answer) is rated on a scale of 0 to 5:

0 - No

- 1 - Incidental
- 2 - Moderate
- 3 - Average
- 4 - Significant
- 5 - Essential

1. Does the system require reliable backup and recovery? (0)
2. Are data communications required? (0)
3. Are there distributed processing functions? (0)
4. Is performance critical? (0)
5. Will the system run in an existing, heavily utilized operational environment? (1)
6. Does the system require online data entry? (3)
7. Does the online data entry require the input transaction to be built over multiple screens or operations? (0)
8. Are the master files updated online? (5)
9. Are the inputs, outputs, files, or inquiries complex? (4)
10. Is the internal processing complex? (4)
11. Is the code designed to be reusable? (5)
12. Are conversion and installation included in the design? (1)
13. Is the system designed for multiple installations in different organizations? (4)
14. Is the application designed to facilitate change and ease of use by the user? (5)

$$\text{Sum}(F_i) = 32$$

$$\text{FP} = 214 \times (0.65 + 0.01 \times 32) = \mathbf{207.58}$$

$$\text{LOC/FP} = 30 \text{ (Given average value for object oriented languages)}$$

$$\text{LOC} = 30 \times 207.58 = \mathbf{6227.4}$$

## 5. Project schedule

Requirement engineering – week 23 (June 10)

- Requirement collection – wk 21 (May 26)
- Prototype development – wk 22 (June 2)
- Feedback from customer on prototype – wk 23 (June 6, Mon)
- Requirement doc ready – wk 23 (June 9, Thu)
- Test design – wk 23

Design – week 24, 25 & 26 (June 30)

- Architectural design
- Modular design
- Data design
- User Interface design
- Design documents
- Test design

### **Week 27 & 28: Summer Holiday**

Implementation – week 29, 30 & 31 (July 29)

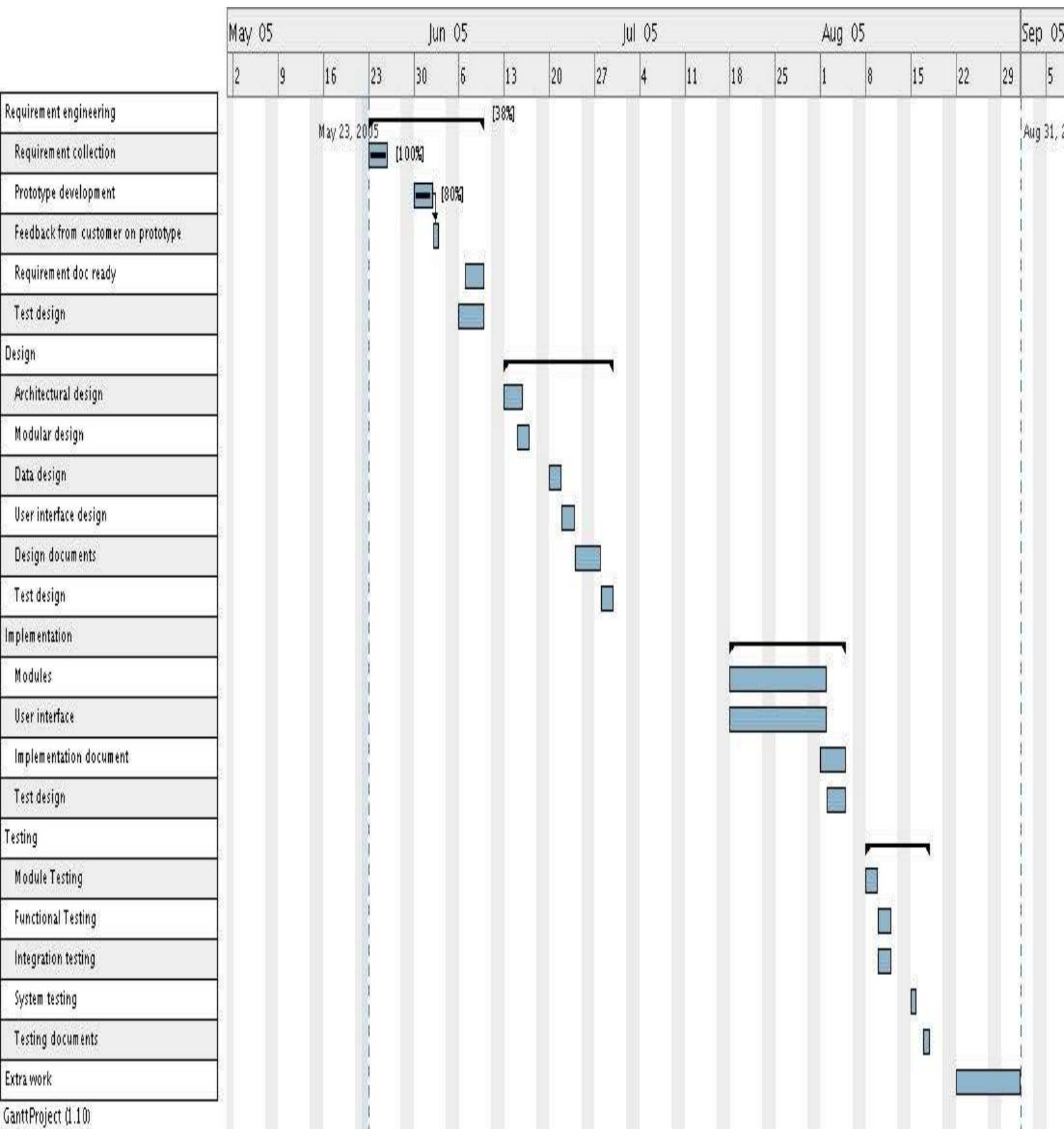
- Modules (Classes)
- User Interface
- Implementation document
- Test design

Testing – week 32 & 33 (August 18)

- Module testing
- Functional testing
- Integration testing
- System testing
- Testing documents

**Extra week (for extra work): wk 34 & 35**

The figure in the next page presents the schedule as a Gantt Chart.



## **6.1 Communication**

The main methods of communication between the project members are group meetings and email. The email address of the group's mailing list is [ohtuv05-canvas-list@cs.helsinki.fi](mailto:ohtuv05-canvas-list@cs.helsinki.fi). The group is divided into two subgroups and the members of the subgroups meet in person for project work as necessary. Peter Ahlberg, Niko Välimäki and Pragya Lamsal form one subgroup, who are able to meet during office hours. Michael Kaakyire and Tony Karppinen form another subgroup, who are not always able to meet during office hours.

## **6.2 Reporting**

The project group meets with the instructor twice a week for two hours to discuss about the project. The progress of the project is reported and the instructor gives comments and suggestions.

The documents produced during the development of the application are put on the homepage of the project group mentioned in the introduction section of this document. The instructor and the customer can check the document at any time, if needed.

## **6.3 Tools**

### **Process Model:**

This project will use *waterfall model* for the development. Waterfall model contains following stages:

1. Requirement Engineering phase
2. Design phase
3. Implementation phase
4. Integration and Testing phase

As mentioned above, a prototype will also be developed during the requirement engineering phase in order to compare with the requirement specification document.

This model produces the following documents:

1. Project Plan document
2. Requirement Specification document
3. Design document
4. Implementation document
5. Testing document
6. User Manual



**Programming tools:**

Java programming language on Linux operating system will be used for implementation.

Eclipse is used for implementation and testing.

OpenOffice will be used for the writing all the documents of this project. The documents will also be available in pdf format.

For version control, CVS is used.

**7. Risk Analysis**

1. Project not meeting the requirements  
Prototype is built and shown to the customer in order to get feedback on the requirements. The customer can be consulted during any stage of the development process in order to get feedback, if needed.
2. Project running overtime  
Schedule has to be analyzed and checked with the ongoing development in order to estimate if the project will run overtime. If the project is likely to be late, then only the core requirements are implemented and optional requirements will be left out.
3. Illness of a group member  
The group is divided into subgroups and members work in as a pair so that if someone gets ill, the other member of the pair can handle the work without difficulty or losing time in understanding the work being done by the ill member.
4. Group member leaving the group  
Priorities of the requirements are checked and optional requirements left out in order to reduce the workload of the remaining members and finish the project on time.
5. Late changes to requirements  
Not likely to occur.