

CITY TRAFFIC SIMULATION

PROJECT PLAN

Version 1.2



HYENA
Crossing



Fontys

University of Applied Sciences

CONTENTS

•	1.INTRODUCTION	1
•	1.1 PURPOSE OF PLAN	1
•	2.PROJECT STATEMENT	2
•	2.1 FORMAL CLIENT	2
•	2.2 TEAM	3
•	2.3 CURRENT SITUATION	4
•	2.4 PROJECT JUSTIFICATION	4
•	2.5 PROJECT PRODUCT	4
•	2.6 PROJECT DELIVERABLES & NON-DELIVERABLES	4
•	2.7 PROJECT CONSTRAINTS	5
•	2.8 RISK ASSESSMENT	5
•	3.PROJECT PHASING AND MILESTONES	7
•	3.1 SKILLS	8
•	3.2 QUALITY	8
•	3.3 Team roles and responsibilities	8
•	3.4 ORGANIZATION	9
•	3.5 PROJECT APPROACH	9
•	3.6 DEVELOPMENT APPROACH	10
•	3.7 OVERALL PROJECT APPROACH	10
•	3.8 MOSCOW	12

• 1.INTRODUCTION

1.1 PURPOSE OF PLAN

The SDM Project Plan will provide a definition of the project, including the project's goals and objectives. Additionally, the Plan will serve as an agreement between the following parties: Project Sponsor, Project Manager, Project Team, and other personnel associated with and/or affected by the project.

The Project Plan defines the following:

- Project purpose
- Project statement
- Business and project goals and objectives
- Management plan
- Risk assessment
- Roles and responsibilities
- Assumptions and constraints

• 2.PROJECT STATEMENT

2.1 FORMAL CLIENT

Formal client:

Name: Andrius Kuprys

Email: a.kuprys@fontys.nl

Telephone number: 0885076989

Tutor:

Name: Andrius Kuprys

Email: a.kuprys@fontys.nl

Telephone number: 0885076989

2.2 TEAM

Team name: Hyena Crossing

Team members:

Name	Role	Occupation	Email	Student Number
Yousef Abu Zahra	Team Leader	ICT & Software Engineering student at Fontys	y.abu.zahra@student.fontys.nl	3519449
Alina Baci	Team Secretary	ICT & Software Engineering student at Fontys	a.baci@student.fontys.nl	3485161
Bozhidar Bonev	Team member	ICT & Software Engineering student at Fontys	b.bonev@student.fontys.nl	3277550

Borislav Pavlov	Team member	ICT & Software Engineering student at Fontys	b.palvov@student.fontys.nl	3664198
Ghazi Abdul Fattah	Vice Leader	ICT & Software Engineering student at Fontys	g.fattah@student.fontys.nl	3205010

2.3 CURRENT SITUATION

Our client works for Sim Software Inc. which is a company for building simulation software. Currently there are numerous car accidents in the city because of badly made configuration of the roads and no traffic lights.

2.4 PROJECT JUSTIFICATION

In order to make the city a safer environment, Mr. Kuprys hired us to build a simulation software in which they can test out different scenarios to optimize the traffic.

2.5 PROJECT PRODUCT

The product of our project is going to be a software application that simulates real life traffic. This application is going to run in windows environment and will be built using C# as a programming language.

2.6 PROJECT DELIVERABLES & NON-DELIVERABLES

Deliverables:

- A simulation application for real life traffic.
- Project plan.
- Test plan.
- URS (User Requirements Specification).
- DDS (Design Document Specification), which is going to include UML class diagrams.
- Manual
- Source code
- Working prototypes at the end of each iteration.
- Presentation.

- Process report.

Non-Deliverables:

- Training for the use of application.
- Any additional software that the application might need.

2.7 PROJECT CONSTRAINTS

These are the limitations that our project will have to abide by:

- **Time**
We need to finish the project in a window of 18 weeks.
- **Programming language**
The application will be written in C# mainly.
- **Version control software**
We need to have a version control software in place.
- **Running environment**
The application should run on Windows environment.

2.8 RISK ASSESSMENT

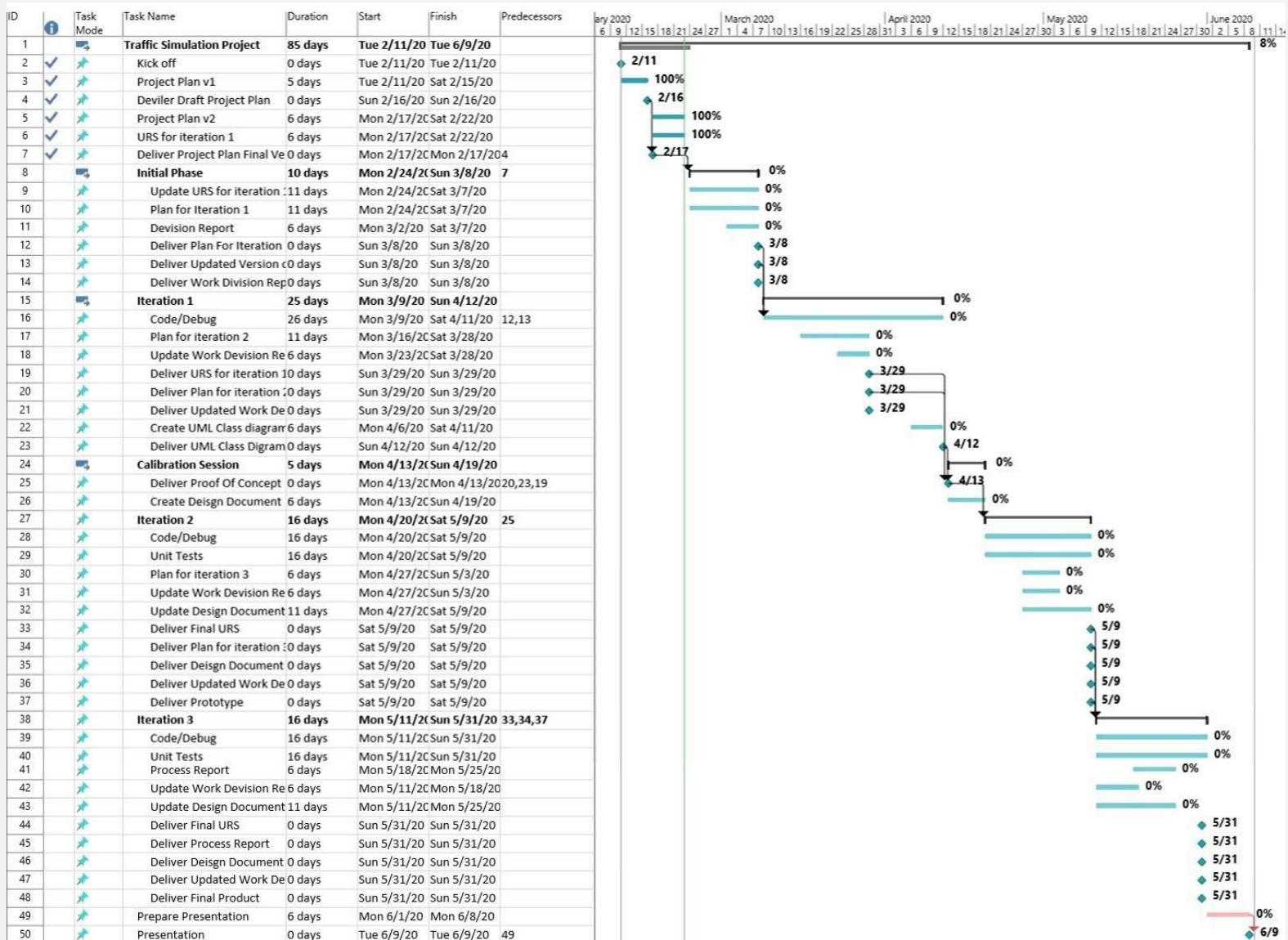
Risks

In every project, there always will be risks involved. Here's a list of the risks that we think our project might face:

RISK	DESCRIPTION	PROBABILIT Y	IMPAC T	PREVENTIVE MEASURES
Falling behind on deadlines/milestones	Activities taking longer than expected	Medium	High	Having frequent meetings to keep up with each other and the client. Also having milestones in place which have a deliverable for each one of them.

Communication problems	Conflict within team members. Absence and other disciplinary issues.	Low	Medium	The team leader has to make sure that everyone is on the same page.
File loss/corruption	Losing some or all of the files, or having them corrupted.	Low	High	Using version control software to make sure that everything is backed up.
Final product not meeting all promised features	The final containing wrong requirements or not all of the requirements we promised to deliver.	Medium	Medium	Keeping track of all the requirements in our documents. Also getting feedback from our client after each iteration and during periodic meetings.
Final product lacking quality	Final product is buggy or inaccurate	Medium	High	Performing periodic testing during and after each iteration.
Lack of experience within our team	Our team lacking the background knowledge needed to finish the project, whether software or real life knowledge.	Medium	Medium	Doing research before and during the project.

3.PROJECT PHASING AND MILESTONES



3.1 SKILLS

In order to complete our project, these are the skills that our team needs to have:

- Project Management
- Programming skills in C#
- Testing skills
- Risk Management
- Database Administration
- User interface design skills.
- Quality Management
- Administrative / Secretarial Skill
- Communication skills

3.2 QUALITY

Our software application must be able to fulfill all the requirements specified by our client. It also needs to be delivered on time.

The interface of our application should be interactive and easy to use. It must implement a form of assistance to help guiding new users.

After each iteration, we need to have a working prototype that adheres to our client's and test users' expectations and feedback.

Our team is going to ensure that any bugs we might face during development will be dealt with, and that no errors will be displayed during runtime.

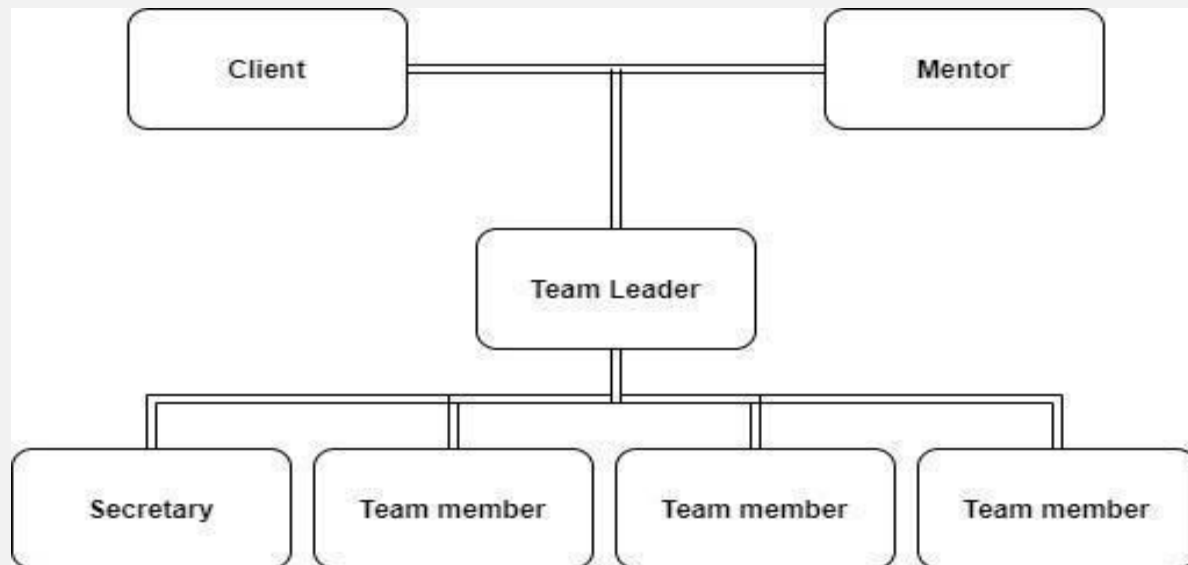
Alongside all of this, the documentation that we will provide should be clear and easy to follow.

3.3 Team roles and responsibilities

ROLE	RESPONSIBILITIES	PARTICIPANT(S)
Team leader	<ul style="list-style-type: none">• Make some project decisions• Communicating with the client• Resolve conflict within team• Divide the workload• The same responsibilities as a team member	Yousef Abu Zahra
Vice Leader	<ul style="list-style-type: none">• Replaces the leader in case of absence/sickness• The same responsibilities as a team member	Ghazi Abdul Fattah

Secretary	<ul style="list-style-type: none"> • Keep track of the meeting minutes • Update the process document • The same responsibilities as a team member 	Alina Baci
Team member	<ul style="list-style-type: none"> • Create applications • Design interfaces • Documentation 	Bozhidar Bonev Borislav Pavlov
Tutor	<ul style="list-style-type: none"> • Give feedback • Track the project progress 	Andrius Kuprys
Client	<ul style="list-style-type: none"> • Approve project decisions • Provide the requirements • Gives feedback 	Andrius Kuprys

3.4 ORGANIZATION



3.5 PROJECT APPROACH

In this section, we are going to specify which approach we are going to take for both the development process and managing the project:

3.6 DEVELOPMENT APPROACH

For this project we will be following the Agile methodology to develop the software. We will be working in iterations and deliver a working prototype at the end of each iteration.

For version control, we will initiate the project with a master branch. When we start working on the application iterations, we will create a “developer” branch which also is going to contain individual sub-branches for each member of the team. At the end of the iteration, we will merge developer with master, creating an application release.

The reason we’re taking this approach is to keep the master branch free from merging conflicts and have a clear checkpoint in case something goes wrong with future iterations.

3.7 OVERALL PROJECT APPROACH

The overall project approach will be the scrum methodology. The work will be completed in short cycles and the team will meet weekly to discuss tasks and roadblocks that need clearing. All of the work required will be stored in the product backlog or on the scrum board.

The work will be done in an incremental, iterative approach.

FUNCTIONALITY	PROGRESS	START DATE	END DATE
ITERATION 1			

Start simulation

Stop simulation

Pause simulation

Reset the map layout

Add/Remove one-way roads

Add/Remove two-way roads

Add/Remove intersections

Add/Remove traffic lights

Save the simulation

Save as a file

Specify the number of cars generated

Generate report with information about current simulation

ITERATION 2

Add/remove highways

Add/Remove roundabouts

Add/Remove non-priority roads

Add/Remove turn right signals

Change the traffic light time

Open file

Add/Remove pedestrians

Cars follow Ford-Fulkerson algorithm for maximum flow

Cars follow Dijkstra algorithm for shortest path

ITERATION 3

Simulate traffic flow based on road sensors

Add/Remove Bridges

Add/Remove Joins

Add/Remove stop signs

Add/Remove maximum/minimum speed signs

Add special types of cars

Add surprise elements

Increase/Decrease pedestrian traffic

Change the speed of the simulation

Help section

3.8 MOSCOW

Our project is going to contain many features, these features carry different weights: some of them are a must have, while others would make a nice addition but would not cause our project to fail if they were to be left out. And some others will not be implemented at all due to the limitations/scope of the project.

In the following table, you can see the list of the features our project is going to contain, and how important each one of them is to our project' success:

Legend for MoSCoW Table: **M** = must have
S = should have

C = could have
W = won't have

FEATURE	M	S	C	W
Simulation runtime controls				
Start simulation	✓			
Stop simulation	✓			
Pause simulation	✓			
Generate report with information about current traffic	✓			
Map layout				
Note: The road features we will create are going to be presented in pre-made, puzzle-like pieces (e.g. a road with a stop sign on it that you can place on the grid, the user cannot separate the road and the stop sign on it).				
Add/remove one-way roads	✓			
Add/remove two-way roads	✓			
Add/remove highways		✓		
Add/remove bridges			✓	
Add/remove joins			✓	
Add/remove intersections	✓			

Add/remove roundabouts

Add/remove traffic lights & zebra crossings

Add/remove stop signs

Add/remove non-priority road

Add/remove turn right signals

Add/remove maximum/minimum speed signs

Reset the map layout

Create traffic templates with number of cars per hour

Change the traffic light time

Extra traffic signs

File options

Save

Save as a file

Open saved file

Record the simulation

Dynamic elements

Add/Remove pedestrians

Increase/Decrease pedestrian traffic

Specify the number of cars generated

Inserting different sizes of vehicle

Change the speed of the simulation

Add surprise elements (e.g. car crashes)

Add special types of cars (e.g. police cars)

	✓		
✓			
		✓	
	✓		
		✓	
		✓	
✓			
	✓		
	✓		
			✓
✓			
	✓		
		✓	
✓			
			✓
		✓	
		✓	
		✓	

User Account & Help

User can sign-up for an account

User can log-in/out

Display tips

Cars Movement Algorithms

Cars follow Ford-Fulkerson algorithm for maximum flow

Cars follow Dijkstra algorithm for shortest path

			✓
			✓
	✓		
	✓		
		✓	