MÄLARDALENS HÖGSKOLA

Car Gossip
Final Project Report

Version 1.0

Revision History

Date	Version	Description	Author
2013-01-19	0.01	Fill part of the document	AP
2013-01-20	1.0	Final version	DR

Doc. No.:

Table of Contents

1. Introduction

1.1 Purpose of this document

The purpose of this document is to display an overview of the results and metrics of the Car Gossip project, as performed during the Distributed Software Development course. This course is joint with the courses Mälardalen University (MDH) in Sweden and University of Zagreb (FER) in Croatia.

1.2 Intended Audience

USER	USE
Project Manager	To keep track of whether the development process follows the agreed development plan
Other project members	A guideline on how to implement components and define how they should interact
Supervisor and other stakeholders	Overview of the product to be developed

1.3 Scope

This document will only cover the results of the project, including: organization, milestones achieved, project results, project experiences and metrics for showing contribution from each team member.

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions
Gossip	A message containing information about the cars position and movement.
Alert	A default message sent by drivers to inform others of the traffic status.

1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
ETHZ	Federal Institute of Technology Zurich
FER	Faculty of Electrical Engineering and Computer Science
DSRC	Dedicated short range communication
GPS	Global Positioning System
App	Application
SCRUM	An agile software development method for project management

1.5 References

This document is based on the previously developed deliverables available at http://fer.unizg.hr/rasip/dsd/projects/cars_talk/documents

2. Background and Objectives

The main goal of this project is to provide a simulation of a message-based information exchange for vehicular traffic. The aim is to enrich the information that the driver has at his disposal regarding traffic conditions, hinders and generally his surroundings.

Such information can be provided through different means, mainly by visualizing other cars in the driver's surroundings on a map using the information from Gossips (car ID, GPS coordinates, speed, velocity) received via DSRC.

The system will also enable the driver to send and receive Alerts regarding traffic-related events (car crash, street closed for repair etc.). The driver will be able to report those events and distribute them via DSRC to other vehicles within reach.

Moreover, the messages (Gossips and Alerts) are stored in a database which is connected to a web service that is able to exploit them in order to provide predictions regarding traffic-related events. Such predictions will be distributed in the same way as for custom messages in the surrounding area via DSRC.

3. Organization

3.1 Project group

Name	Initials	Responsibility (roles)
David Reypka	DR	Project (team) leader, Documentation
Mohammed Alserr	MA	Testing leader, Persistence manager
Achile Penna	AP	Requirements engineering, Architecture manager
Nikola Vranešić	NV	Team Leader, GUI manager
Matija Korpar	MK	Lead developer, Algorithm leader
Dino Jurina	DJ	Protocolist, SVN manager

3.2 Customer

DSD Course Teachers
SCORE Contest Committee

3.3 Supervisor

Federico Ciccozzi

4. Development process

The Car gossip project will be developed following a SCRUM-like process through an iterative and incremental agile software development method. There are three core roles: product owner, scrum master and the development team.

The product owner is responsible for prioritizing work on the product and is a good communicator. David Reypka is the product owner in the Car gossip project since he is also the Project Leader.

The scrum master is responsible for supporting the Scrum Team, coaching and guiding them through this process, and removing any impediments blocking their progress. Nikola Vranešić is the scrum master in the Car gossip project.

All the team members of the project represent the development team which is responsible for delivering a potentially shippable product that is enhanced at the end of each Sprint.

“A sprint is the basic unit of development in Scrum. Each sprint is preceded by a planning meeting, where the tasks for the sprint are identified and an estimated commitment for the sprint goal is made, and followed by a review or retrospective meeting, where the progress is reviewed and lessons for the next sprint are identified. During each sprint, the team creates finished portions of a product. “

by Kelly Waters, 14 September 2007

How To Implement Scrum in 10 Easy Steps

This projects sprint duration is 3 weeks. After each sprint the team presents a new version of the product (alpha, beta and final). The sprint overviews are defined in the deliverables.

5. Milestones

To	Output	Planned week	Promised week	Late +/-	Delivered week	Rem
Customer	Project Plan (v.1)	W44	W44	-	W44	PDD
Customer	Requirements Definition (v.1)	W44	W44	-	W44	RDD
Customer	Design Description (v.1)	W45	W45	-	W45	DDD
Internal	Open Data Source	W47	W47	-	W47	ODS
Internal	Desktop App	W47	W47	-	W47	DA
Internal	Android App	W47	W47	+3	W50	AA
Customer	Alpha Prototype	W48	W48	-	W48	APT
Internal	Web Server	W50	W50	+5	W03	WS
Internal	Web App	W50	W50	+5	W03	WA
Customer	Beta Prototype	W51	W51	-	W51	BPT
Customer	Acceptance Test Plan	W01	W01	+1	W02	ATP
Customer	Test Report	W03	W03	-	W03	TR
Customer	Final Project Report	W03	W03	-	W03	FPR
Customer	Final Product	W03	W03	-	W03	FP

6. Project Results

6.1 Requirements

6.1.1 Requirement Compliance Matrix

Id	Requirement Description	completed	Rem
ODS-1	Parses data from NS-2 movement format files downloaded into CSV formatted data with new calculated data as well	Yes	ODS
TS-1	Simulates vehicular movement (traffic behavior) based on input parameters from the Operator and retrieved data from the Open Data Source.	Yes	OPRT, ODS
TS-2	Simulates a DSRC cloud (gossip spreading) by forwarding messages to other cars that are in range of the message sender.	Yes	ICD
TS-3	Visualizes the traffic behavior on a map as a real time event.	Yes	TS
TS-4	Visualizes the gossip spreading on a map as a real time event.	Yes	TS
ICD-1	Creates Gossips out of data retrieved from the ODS simulating ICS's supplying the data.	Yes	TS
ICD-2	Sends messages to the DSRC cloud simulating broadcasting messages via DSRC.	Yes	ICD
ICD-3	Receives messages from the DSRC cloud simulating reception from other cars via DSRC.	Yes	TS
ICD-4	Sends messages to an connected Android device running the Car Gossip App via BT.	Yes	ICD
ICD- 5	Receives messages from an connected Android device running the Car Gossip App via BT.	Yes	AApp
ICD-6	Filters received messages using a priority queue.	Yes	ICD
AApp-1	Receives messages from a connected ICD via BT.	Yes	ICD
AApp-2	Visualizes traffic movements on a map.	Yes	AApp
AApp-3	Enables the Driver to send Alerts by pressing an icon button on the screen.	Yes	AApp
AApp-4	Sends messages to a connected ICD via BT.	Yes	DRV
AApp-5	Sends messages to Web Server via HTTP(S).	Partially	AApp
AApp-6	Receives messages from Web Server via HTTP(S).	Partially	WS
WS-1	Receives messages from an Android device running the Car Gossip App via HTTP(S).	Partially	AApp
WS-2	Stores the received messages into a DB which can be accessed via a public API.	Partially	DB
WS-3	Validates Alerts to prevent spamming.	Partially	WS
WS-4	Pushes messages to all Android devices running the Car Gossip App via HTTP(S).	Partially	WS
WS-5	Responds to Web App requests using Web services.	Partially	WApp
WApp-1	Sends requests to Web Server to access and process stored traffic data.	Partially	WS

6.1.2 Requirements Compliance Summary

Summarize the requirements compliance data.

Total number of requirements	23
Number of requirements implemented	15
Requirements partially fulfilled	8
Requirements not fulfilled	0
Requirements dropped	0

6.1.3 Remarks

Remark Id	Description
ST	Simulate traffic
SICD	Simulate an ICD
SRM	Send and receive messages
SGA	Spreading Gossip Algorithm
CS	Control spam
ODS	Open Data Source
DRV	Driver
AS	Android Smartphone
SOICD	Simulator of ICD
SOT	Simulator of traffic
WS	Web Server
DB	Data Base

6.2 Deliverables

To	Output	Planned week	Promised week	Late +/-	Delivered week	Rem
Customer	Project Plan (v.1)	W44	W44	-	W44	PDD
Customer	Requirements Definition (v.1)	W44	W44	-	W44	RDD
Customer	Design Description (v.1)	W45	W45	-	W45	DDD
Internal	Open Data Source	W47	W47	-	W47	ODS
Internal	Desktop App	W47	W47	-	W47	DA
Internal	Android App	W47	W47	-	W47	AA
Customer	Alpha Prototype	W48	W48	-	W48	APT
Internal	Web Server	W50	W50	-	W03	WS
Internal	Web App	W50	W50	-	W03	WA
Customer	Beta Prototype	W51	W51	-	W51	BPT
Customer	Acceptance Test Plan	W01	W01	+1	W02	ATP

Customer	Test Report	W03	W03	-	W03	TR
Customer	Final Project Report	W03	W03	-	W03	FPR
Customer	Final Product	W03	W03	-	W03	FP

7. Risks

Look at the risk table from the Project Plan document and list and comment:

- *risks that have appeared but their impact was low because of preventive actions*
- *risks that appeared and had a significant impact on project work*
- *risks that appeared but were not foreseen and listed in the table (describe them and their impact on project work)*

New risks that appeared:

Likelihood	Risk	Preventive action	Significance
Low	A member leaves the group	Redistribute the work when it was highly likely that he leaves	High
Medium	A team member does not work enough	Motivate the person and give him clear orders, if not successful try to give lower priority tasks	Medium
High	Miscommunication	Constant communication through several channels between the two groups. High information flow between Team Leaders.	High
Medium	Lack of knowledge in used technologies	Decide on technologies the majority is familiar with.	High
Medium	Unspecific software design	Agile and iterative design and development	Low

It was not expected that team members leave and people don't work their expected hours.
This was hard to compensate, since the work was divided in an equal way and it was found out only later that the group members wore not working with the same attitude.
The significance for the member leaving was high, since he had a big part of the implementation and a great influence on the chosen technologies.
On the other hand it was clear at a relatively early stage, that one person would not work equal, while the member leaving was not expected and at a much later stage.

The miscommunication between the group members was a constant threat, especially during the holiday period.
It was tried to counter with more meetings and the inclusion of voice chat.

Since a group member with experience in the chosen technologies left, it was hard to compensate with less experienced people. So we had to face difficulties in certain areas of our development (Android and web server development).

Since we aimed to refine our design at later stages of the project and have more of an agile approach to development, it was not always clear to which extent features will be implemented and how the interfaces will fit together.

8. Project Experiences

8.1 Positive Experiences

We found out that one component could be developed in a fast way and finished earlier than expected.
So there was enough time to refactor, polish and test this part. This part achieved a very satisfying level of completeness.

The group work between some small sub groups was efficient and produced greater results than working on each own.

It was an exciting project with potential and a lot of room for creativity and own ideas, since there was no customer involved to specify which direction the project will take.

Some meetings were successful in creating a good work motivation and overview of the current state of the project and each member had the possibility to bring in their own ideas.

8.2 Improvement Possibilities

8.2.1 Communication:

The communication could be improved a lot. The project started out by using five different kinds of communication (mail, Skype, Google drive, Google groups, live meetings) but some of those tools were reduced during the project (live meetings relatively rare) and some were not used at all anymore (Google groups).
During the alpha phase there was also an project management tool called Trac introduced, which provided some added benefits (ticketing system) but did not really improve the communication overall.

The meetings were a good source of information for every team member and this even increased through the use of voice chats, but outside of the meetings the information flow was not steady and declining after longer periods without meetings.

So in retrospect a constant time for meetings every day or every two days would provide a good information flow between the members.
But it is also recommended to improve the group bonding, so that the group members know each other and work better and in a more group-benefitting way.

8.2.2 Timeliness:
The timeliness could also be improved in way to set strict deadlines before official deadlines outside of the project group.
If deadlines are not met and there are no good reasons to justify it, the group should discuss and make the necessary changes so it does not happen again.

9. Metrics

9.1 Work per Member

Member	W43	W44	W45	W46	W47	W48	W49	W50	W51	W52	W1	W2	W3	Total
David Reypka	18	20	16	16	14	18	19	26	16	8	8	30	32	241
Nikola Vranesic	10	19	18	16	16	10	27	20	6	0	22	0	30	194
Achille Penna	10	14	12	15	15	7	16	25	12	21	11	26	24	198
Mohammed Alserr	13	14	21	21,5	25	16	17	0	0	0	0	0	0	127,5
Matija Korpar	10	10	14	20	13	18	19	23	14	10	10	10	20	191
Dino Jurina	8	10	8	2	16	13	9	17	11	0	10	10	0	114
Total	69	87	89	90,5	99	82	105	111	57	39	61	76	106	1065,5

Mohammed Alserr left the group after W49 and so did not contribute anymore.
Dino's worked hours have not been up to par with the other members.
Nikola Vranesic had unfortunately no steady workflow and some weeks where he was absent.

9.2 Milestone Metrics

Completed as planned or earlier	Total	Timeliness
10	14	71,4%

9.3 Effort Metrics

ID	Activity	Actual Effort	Planned Effort	Deviation (%)
	Architecture Draft	18	18	0
	Project Plan	6	6	0
	Requirements Definition	9	6	50
	Design Description	20	20	0
	Open Data Source	4	7	57
	Desktop App	25	30	83
	Android App	26	18	69
	Alpha Version	26	42	61
	Web Server	20	42	47
	Web App	8	42	19
	Beta Version	30	42	71
	Component refactoring	4	14	28
	Documentation revision	25	14	56
	Acceptance Test Plan	14	7	50
	Test Report	2	7	28
	Final Product	32	84	38

Effort estimation accuracy (%) *(100*(1 - abs(Actual – Planned)/Actual))*	41%

We had trouble with the Android application and could not work as much as we wanted on the web server and web application part, since that was delayed a few times and had a low priority for us.

We overestimated the effort for the final product and overall did not put in as much work as we wanted to.

Also due to one person working not as expected and one member leaving we had troubles keeping up with our schedule and did not make the necessary decisions to account for these less worked hours.

The expected peak work hours before the Alpha version and to a slighter extend the Beta version and most of all the Final version were not as big as expected, due to some members not putting in enough effort and not having the most efficient work distribution.