# Notes on Deliverable 1

Massimo Felici

## Steps Towards Deliverable 1

- 1. Gathering Requirements
  - Writing a Requirements Specification Document (e.g., see the VOLERE template)
- 2. Capturing functional requirements into Use Cases
  - Describe use cases by a Use Case Template
- 3. Modelling a preliminary system design into Class Diagrams
- 4. Validating your design by CRC cards

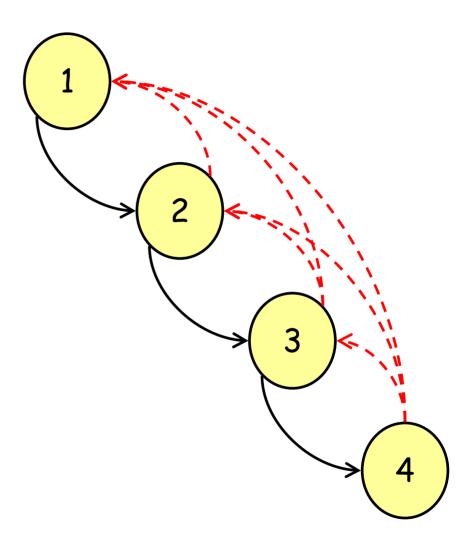
# Steps Towards Deliverable 1

1. Requirements

2. Use Cases

3. Class Diagrams

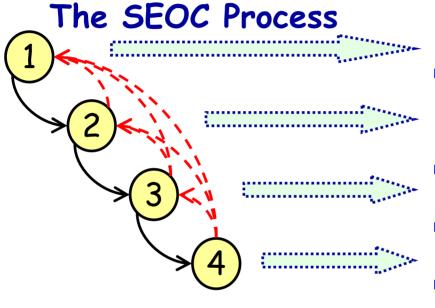
4. Validation



# The SEOC Project Deliverable 1

- 1. Requirements Specification
- 2. Use Case Model
- 3. Class Model
- 4. Validation of Class Model
- 5. Deliverable Assessment

# SEOC Activity Deliverables



### Deliverable 1

- Requirements Specification
- Use Case Model
- Class Model
- Validation of Class Model
- Deliverable Assessment

### Deliverable 1 Assessment

### Deliverable 1

- RequirementsSpecification
- Use Case Model
- Class Model
- Validation of Class Model

Part 3 - Deliverable Marking Scheme

	Deliverable Marking Scheme	
Deliverable Part	Questions	Marks
Requirements	Q1. Did you organise/collect the system requirements by using a Requirements Specification template (e.g., Volere)? Assess the quality of your Software Requirements Specification (SRS) docu- ment.	[ / 5]
Marks Limit: [20/100]	Q2. Did you distinguish different types of requirements (e.g., functional or non-functional)? Assess how your SRS identifies different types of requirements.	[ / 5]
	Q3. Do you believe you got most of the system requirements (requirements completeness)? Assess the extent to which you have elicited and gathered requirements from the main sources.	[ / 5]
	Q4. Have you identified/resolved conflicting requirements (requirements correctness)? Assess the extent to which you have resolved conflicting requirements among different types (e.g., functional and non-functional) or across teams.	[ / 5]
Use Cases	Q5. Did you graphically represent the functional requirements by Use Cases? Assess to which extent your use case diagram captures main system functionalities and actors.	[ / 10]
Marks Limit: [30/100]	Q6. Did you refine the use cases by generalization, include or extend relationships? Assess to which extent you have refined and structured use cases.	[ / 10]
	Q7. Did you use a template for describing use cases? Assess to which extent you have clarified and described use case information (completeness and correctness).	[ / 10]
Class Diagrams	Q8. Does your class diagram identify the main classes of the system? Assess to which extent your class diagram realizes system use cases.	[ / 10]
Marks Limit: [30/100]	Q9. Did you specify Attributes and Operations for each class?  Assess the completeness of class specification.	[ / 10]
[607 100]	Q10. Did you identify relationships (i.e., Dependency, Association, Aggregation, Composition and Inheritance or Generalization) between classes? Assess the object orientation quality of your class diagram.	[ / 10]
CRC Cards	Q11. Did you construct CRC cards for your system design? Assess the completeness and correctness of CRC cards.	[ / 10]
Marks Limit: [20/100]	Q12. Did you verify your Class Diagrams? Did you play any use case with the CRC Cards in order to verify your class diagram? Assess the quality and the coverage of your requirements and design verification by CRC cards.	[ / 10]
Deliverable N	Mark	[ /100]

# CARMATCH

# CARMATCH Background

- CARMATCH is a franchising company that is being set up to promote car sharing
- Organizational goal: reduce carbon emissions
- CARMATCH seeks to promote car sharing
  - Matching potential car sharers
- CARMATCH consists of a three layer structure: (non-for-profit trust) global operation; national central operating company; local franchises
- In some countries, it offers insurances
- Main Profits: membership fees, consultancies, insurance commissions
- CARMATCH needs (has the requirements for) a computer system that can be used by its

## CARMATCH Requirements

- 1. To develop a system that will hold information about members of the CARMATCH scheme
- 2. To match members up with other members as car sharers
- 3. To record insurance sales
- 4. To record details of potential and actual consultancy in the area of operation
- 5. The system must be capable of future expansion to incorporate information about toll and road-pricing and equipment sold to and installed for members

## CARMATCH Requirements Specification

The ...... System Requirements Specification Version ...

#### Table of Contents

#### PROJECT DRIVERS

- 1. The Purpose of the Project
- 2. Client, Customer and other Stakeholders
- 3. Users of the Product

#### PROJECT CONSTRAINTS

- 4. Mandated Constraints
- 5. Naming Conventions and Definitions
- 6. Relevant Facts and Assumptions

#### **FUNCTIONAL REQUIREMENTS**

- 7. The Scope of the Work
- 8. The Scope of the Product
- 9. Functional and Data Requirements

#### NON-FUNCTIONAL REQUIREMENTS

- 10. Look and Feel Requirements
- 11. Usability and Humanity Requirements
- 12. Performance Requirements
- 13. Operational Requirements
- 14. Maintainability and Support Requirements
- 15. Security Requirements
- 16. Cultural and Political Requirements
- 17. Legal Requirements

#### PROJECT ISSUES

- 18. Open Issues
- 19. Off-the-Shelf Solutions
- 20. New Problems
- 21. Tasks
- 22. Cutover
- 23. Risks
- 24. Costs
- 25. User Documentation and Training
- 26. Waiting Room
- 27. Ideas for Solutions

### Project Drivers

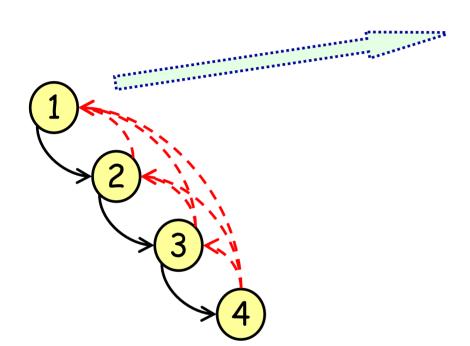
- CARMATCH background
- CARMATCH organization, Local governments, EU?, Locan franchises, car sharers, etc.
- · System Users?

### Project Constraints

- Budget, Deadlines, Laws, etc.?
- Functional Requirements
- To hold information about members
  - 1. 1
  - 2. 2
  - 3. 3
- To match car sharers
- 3. To record insurance sales
- 4. To record details consultancies
- Non-functional Requirements

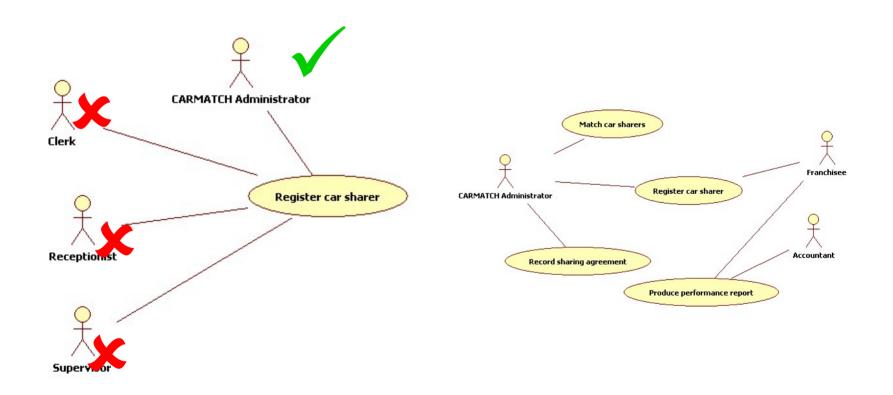
# Requirements Specification

### CARMATCH Requirements Specification

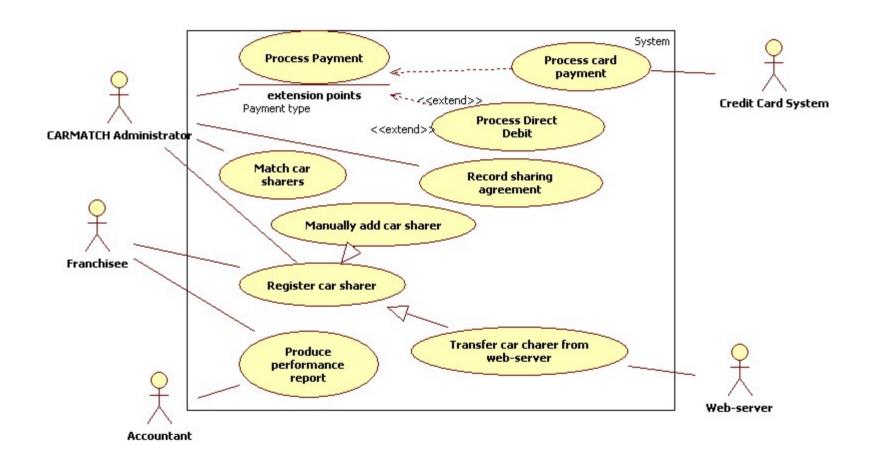


Project Constraints  4. Mandated Constraints  4. Mandated Constraints  5. Relevant Facts and Assumptions  7. The Scope of the Work  8. The Scope of the Work  9. Functional Requirements  10. Look and Feel Requirements  11. Usability and Humanity Requirements  12. Performance Requirements  13. Operational Requirements  14. Constant Requirements  15. Security Requirements  16. Cultural and Political Requirements  17. Legal Requirements  18. Open Issues  19. Off the Shelf Solutions  20. Neer Problems  21. To match car sharers  22. Cutover  23. Risks  24. Costs  25. User Documentation and Training  26. Watning Room  27. Heast for Solutions  28. User Documentation and Training  29. Watning Room  20. User Solutions  20. User Documentation and Training  20. Watning Room  20. User Solutions  20. User Documentation and Training  20. Watning Room  20. User Solutions  20. User Documentation and Training  20. Watning Room  21. To record details consultancies  22. Project Issues  23. Project Issues  24. Project Issues  25. Project Issues	The System Requirements Specifical Version able of Contents PROJECT DRIVERS 1. The Purpose of the Project 3. Sesser of the Project 3. Sesser of the Project	ion	<ul> <li>Project Drivers</li> <li>CARMATCH background</li> <li>CARMATCH organization, Local governments, EU?, Locan tranchises, car sharers, etc.</li> <li>System Users?</li> </ul>
9. Functional and Data Requirements 10. Look and Feel Requirements 11. Usability Requirements 12. Performance Requirements 13. Operational Requirements 14. Maintainability and Support Requirements 15. Lutarial and Political Requirements 16. Cultural and Political Requirements 17. Legal Requirements 18. Open Issues 19. Offer Shelf Solutions 20. New Problems 21. Tasks 22. Cutover 23. Risks 24. To record details consultancies 25. Waiting Room 27. Ideas for Solutions 26. Waiting Room 27. Ideas for Solutions 28. Waiting Room 27. Ideas for Solutions 29. Waiting Room 27. Ideas for Solutions 20. Project Issues 20. Project Issues 21. Tasks 22. Cutover 23. Risks 24. To record details consultancies 25. Waiting Room 27. Ideas for Solutions 28. Waiting Room 29. Waiting Room 29. Waiting Room 20. The Project Issues 29. Project Issues	PROJECT CONSTRAINTS 4. Mandated Constraints 5. Naming Conventions and Definitions 6. Relevant Facts and Assumptions FUNCTIONAL REQUIREMENTS 7. The Soope of the Work		<ul> <li>Budget, Deadlines, Laws, etc.?</li> </ul>
PROJECT ISSUES 18. Open Issues 19. Off-the-Shelf Solutions 10. New Problems 21. To record insurance sales 22. Risks 24. Costs 25. User Documentation and Training 26. Watning Room 27. Heast for Solutions 28. Watning Room 29. Watning Room 29. Watning Room 29. Watning Room 29. Watning Room 20. Project Issues 20. Project Issues	9. Functional and Data Requirements NON-FUNCTIONAL REQUIREMENTS 10. Look and Feel Requirements 12. Reformance Requirements 12. Performance Requirements 13. Operational Requirements 14. Security Requirements 15. Security Requirements 16. Security Requirements 16. Security Requirements 17. Cultural and Political Requirements		members 1. 1 2. 2
20. New Problems 21. Tasks 22. Cutover 23. Risks 24. To record details consultancies 25. Outst 26. Outst 27. Udating Room 27. Ideas for Solutions 28. Use of Solutions 29. Use of Solutions 29. Use of Solutions 29. Use of Solutions 20. Use of Solutions 21. Tasks 22. Cutover 23. To record insurance sales 24. To record details consultancies 25. Use of Solutions 26. Use of Solutions 27. Use of Solutions 28. To record insurance sales 29. To record insurance sales 29. To record insurance sales 29. To record insurance sales 20. Use of Solutions 20. Use of S	PROJECT ISSUES 18. Open Issues		2. To match car sharers
22. Risks 24. Costs 25. User Documentation and Training 26. Watering Room 27. Ideas for Solutions  4. To record details consultancies Non-functional Requirements  Project Issues	20. New Problems 21. Tasks		3. To record insurance sales
26. Waiting Room 27. Ideas for Solutions  Non-functional Requirements sectification prepared by	23. Risks 24. Costs	r.	4. To record details consultancies
pecification prepared by Date Project Issues  104-2007 SEOC - Lecture Note 05 10	26. Waiting Room 27. Ideas for Solutions		Non-functional Requirements
			Project Issues

### CARMATCH Actors and Use Cases



# CARMATCH System Use cases



13

## CARMATCH Use Case Description

Use Case: Register car sharer

**Description**: The registration of the car sharer information and the association with a membership number

Actors: CARMATCH Administrator, Car sharer

Assumptions: the CARMATCH Administrator has to confirm information, and the car sharer has to accept CARMATCH policy

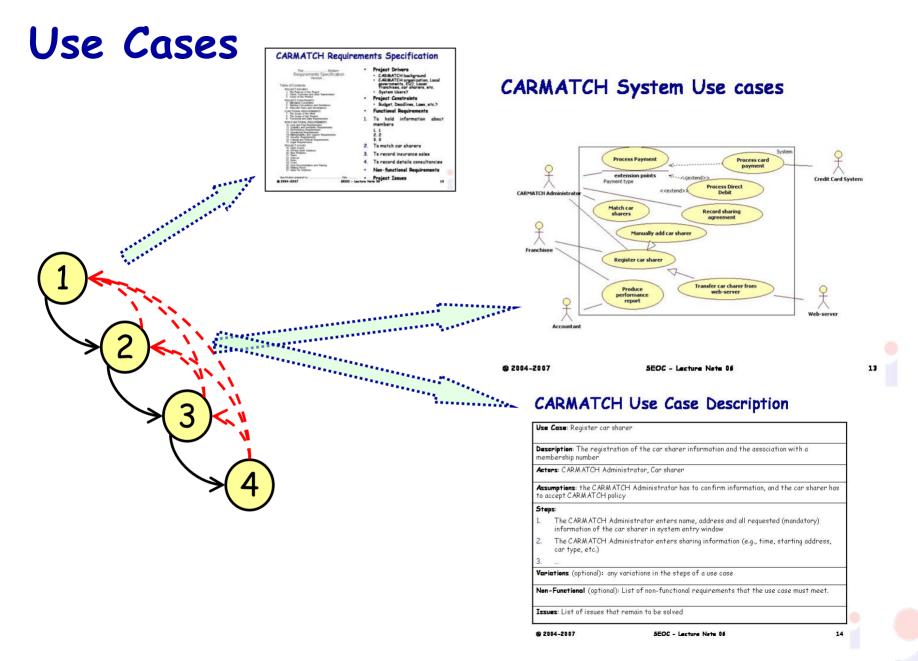
### Steps:

- The CARMATCH Administrator enters name, address and all requested (mandatory)
  information of the car sharer in system entry window
- The CARMATCH Administrator enters sharing information (e.g., time, starting address, car type, etc.)
- 3. ...

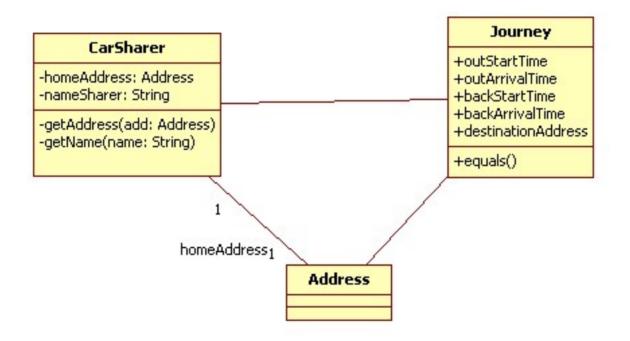
Variations (optional): any variations in the steps of a use case

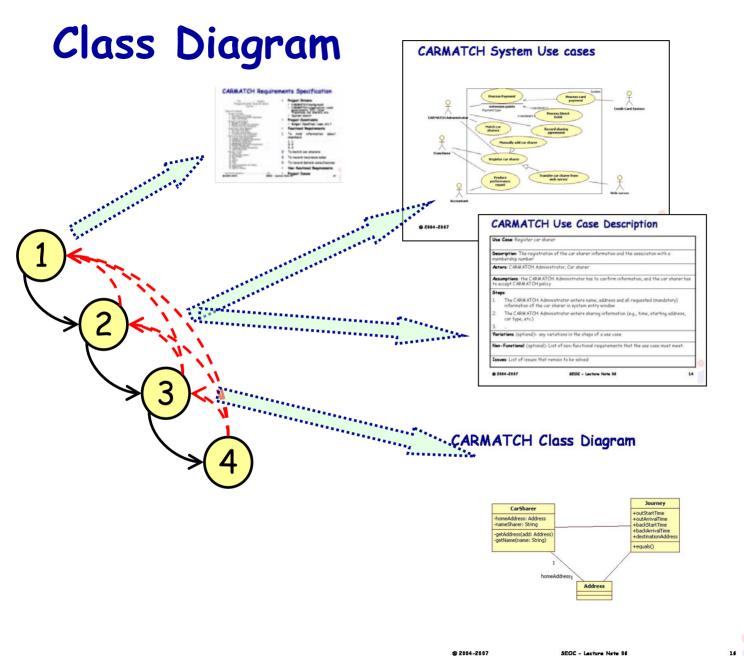
Non-Functional (optional): List of non-functional requirements that the use case must meet.

Issues: List of issues that remain to be solved



## CARMATCH Class Diagram





© 2004-2008 Notes on Deliverable 1

17

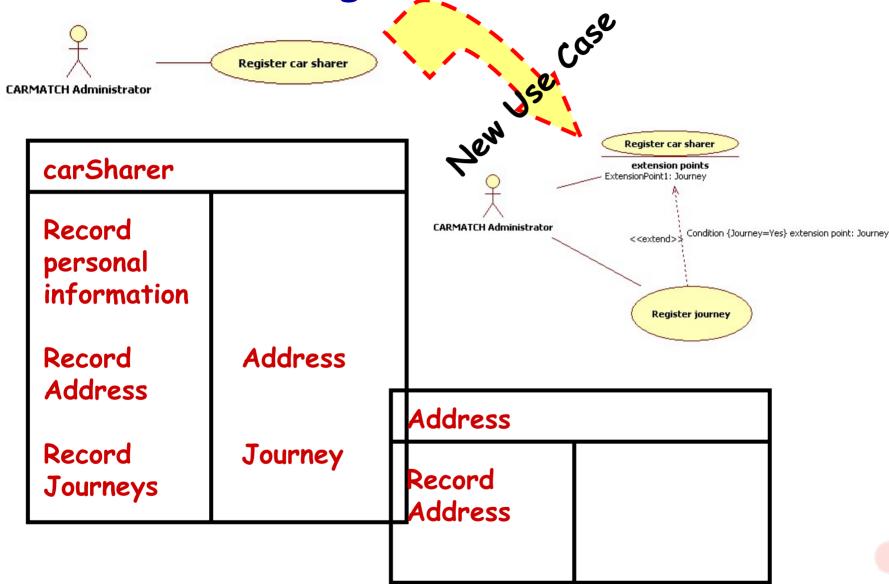
### CARMATCH Validation

carSharer		
Record personal information		
Record Address	Address	
Record Journeys	Journey	

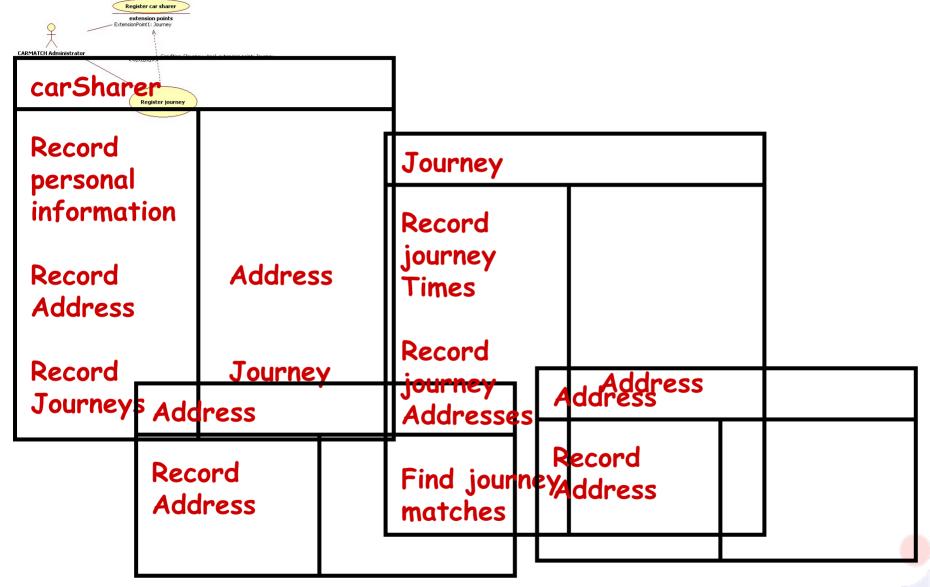
Journey		
Record journey Times		
Record journey Addresses	Address	
Find journey matches		

Address	
Record Address	

CARMATCH: Register car sharer



# CARMATCH: Register Journey

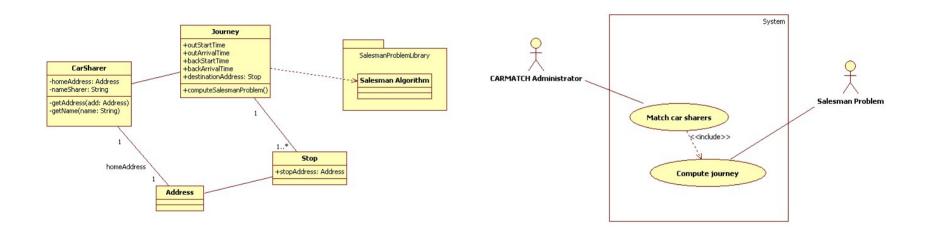


© 2004-2008

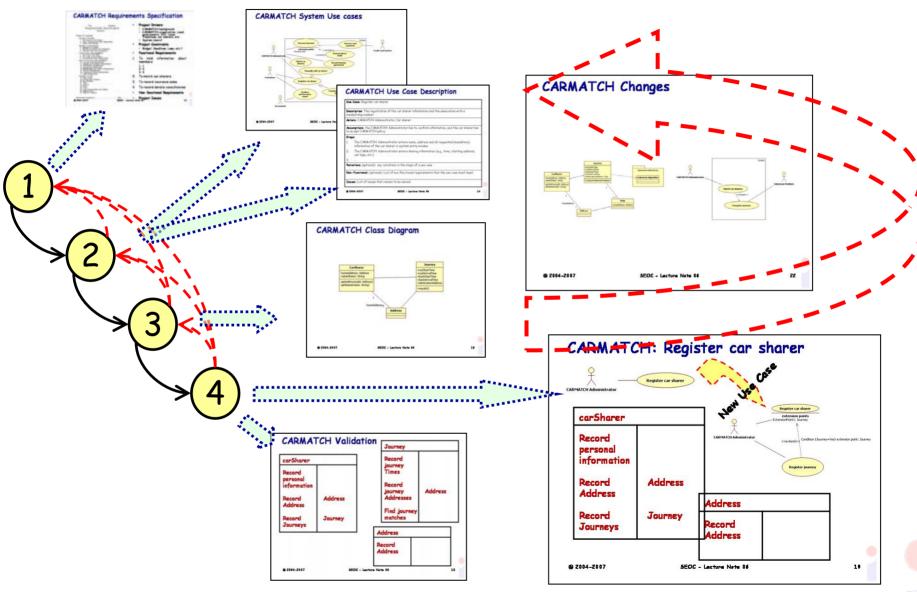
## CARMATCH new requirements

- Efficiency: maximize the combination of journeys by combining multiple stops (i.e., journeys)
- Note that it is a non-functional requirements
- Are there any implications? How does it affect your preliminary design?

# CARMATCH Changes

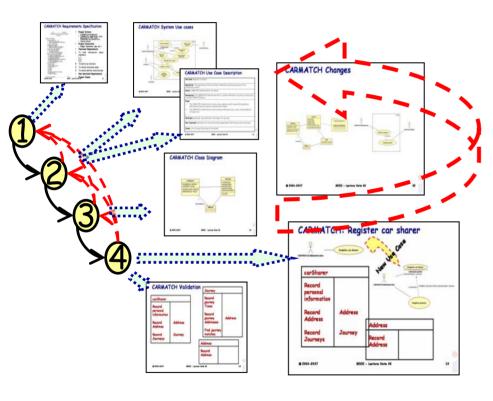


### CARMATCH validation



© 2004-2008

### **Assessment**



Part 3 - Deliverable Marking Scheme

	Deliverable Marking Scheme	
Deliverable Part	Questions	Marks
		[ / =1
Requirements	Q1. Did you organise/collect the system requirements by using a Requirements Specification template (e.g., Volere)? Assess the quality of your Software Requirements Specification (SRS) docu- ment.	[ / 5]
Marks Limit: [20/100]	<b>Q2.</b> Did you distinguish different types of requirements (e.g., functional or non-functional)? Assess how your SRS identifies different types of requirements.	[ / 5]
	Q3. Do you believe you got most of the system requirements (requirements completeness)? Assess the extent to which you have elicited and gathered requirements from the main sources.	[ / 5]
	Q4. Have you identified/resolved conflicting requirements (requirements correctness)? Assess the extent to which you have resolved conflicting requirements among different types (e.g., functional and non-functional) or across teams.	[ / 5]
Use Cases	Q5. Did you graphically represent the functional requirements by Use Cases? Assess to which extent your use case diagram captures main system functionalities and actors.	[ / 10]
Marks Limit: [30/100]	Q6. Did you refine the use cases by generalization, include or extend relationships? Assess to which extent you have refined and structured use cases.	[ / 10]
	Q7. Did you use a template for describing use cases? Assess to which extent you have clarified and described use case information (completeness and correctness).	[ / 10]
Class	Q8. Does your class diagram identify the main classes of the	[ / 10]
Diagrams	system? Assess to which extent your class diagram realizes system use cases.	[ / 10]
Marks Limit: [30/100]	Q9. Did you specify Attributes and Operations for each class? Assess the completeness of class specification.	[ / 10]
[/ ===]	Q10. Did you identify relationships (i.e., Dependency, Association, Aggregation, Composition and Inheritance or Generalization) between classes? Assess the object orientation quality of your class diagram.	[ / 10]
CRC	Q11. Did you construct CRC cards for your system design? Assess	[ / 10]
Cards	the completeness and correctness of CRC cards.	
Marks Limit: [20/100]	Q12. Did you verify your Class Diagrams? Did you play any use case with the CRC Cards in order to verify your class diagram? Assess the quality and the coverage of your requirements and design verification by CRC cards.	[ / 10]
Deliverable Mark		[ /100]