1) What is a requirements document (req. doc.)	?		
Requirements Document       2) How can we write requirements?         Chapter 4.2-4.3	<ol> <li>What is a requirements document (req. doc.)?</li> <li>How can we write requirements?</li> </ol>		
CMPT 276 © Dr. B. Fraser Based on slides from Software Engineering 9 <sup>th</sup> ed, Sommerville.	2		
Software requirements document       Users of a requirements document	Users of a requirements document		
<ul> <li>Software requirements document: <ul> <li>Req. doc. mainly part of a plan-driven method</li> <li>Agile methods argue requirements change too quickly for a req. doc. to be useful.</li> <li>a specification of the system requirements.</li> </ul> </li> <li>NOT a design document: <ul> <li>describes WHAT system should do not HOW it should do it.</li> </ul> </li> <li>Managers user for a req. doc.:</li> </ul> <li>Managers user for a req. doc.:</li> <li>Managers user for a req. doc.:</li>	the requirements and em to check that they eir needs. Customers changes to the ments. requirements ent to plan a bid for em and to plan the development process. requirements to and what system is eveloped. requirements to validation tests for em. requirements to and the system and tionships between its		
3 15-02-04	4		

		Requirements specification	
Requirements specification process		<ul> <li>Requirements specification is         <ul> <li>process of writing a requirements document.</li> <li>Includes the user and system requirements.</li> <li>User requirements must be</li> <li>System requirements are detailed requirements with</li> </ul> </li> <li>Important to be as complete as possible:         <ul> <li>may be the basis for system development contract.</li> </ul> </li> </ul>	
15-02-04	5	6	
Ways of writing a system req. specification		Requirements and design	
Notation Natural language sentences Structured natural language Mathematical specification	Description Requirements are in plain English. Requirements are in English, but written based on a Diagrams and text to describe the system. Ex: UML Ex: Unambiguous, but hard for customers to understand.	<ul> <li>In principle: state the system should do; describes it does this.</li> <li>In practice:</li> <li>Interaction with other systems may generate design requirements.</li> <li>A non-functional requirement may need a specific architecture design.</li> <li>Design constraints may be from regulatory requirements.</li> </ul>	
15-02-04 "English" only an exam	ple natural language. Could use Spanish, French, 7	8	

Natural language specification	Guidelines for writing requirements		
<ul> <li>Natural language: <ul> <li>Requirements often written in natural language:</li> </ul> </li> <li>Natural language used because it is</li> <li>It can be understood by customers and developers.</li> </ul>	<ul> <li>Create a standard format and use it for all requirements.</li> <li>Use language in a consistent way. <ul> <li>Use shall for</li> <li>Use should for</li> </ul> </li> <li>Avoid the use of computer jargon: <ul> <li>use domain terminology</li> </ul> </li> <li>Include an explanation (rationale) of</li> </ul>		
5-02-04 9	15-02-04 10		
Problems with natural language	Structured specifications		
Lack of clarity	Requirements written in		
<ul> <li>Requirements confusion <ul> <li>Functional and non-functional requirements tend to be mixed-up.</li> </ul> </li> <li>Requirements amalgamation <ul> <li>If several different requirements are expressed together.</li> </ul> </li> <li>See the cookie recipe: MIL-C-44072C <ul> <li>Ex: Sections 3.2.2, 3.3.11.</li> </ul> </li> </ul>	<ul> <li>Works OK for embedded control systems.</li> <li>Too restrictive for business systems.</li> <li>Form-based specification <ul> <li>Every requirement written in a standardized form.</li> </ul> </li> <li>Tables: <ul> <li>Good for showing different possible ranges of a condition.</li> </ul> </li> </ul>		
5-02-04 11	15-02-04 12		

Structured spec.: an insulin pump	Tabular specification: Insulin pump		
<ul> <li>Action</li> <li>CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing.</li> <li>If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result.</li> <li>If the result is rounded to zero then CompDose is set to the minimum dose that can be delivered.</li> </ul>	<b>Condition</b> Sugar level falling (r2 < r1) Sugar level stable (r2 = r1)	<b>Action</b> CompDose = 0 CompDose = 0	
Requirements Two previous readings so that the rate of change of sugar level can be computed.	Sugar level increasing and rate of increase decreasing (( $r1) < (r1 - r0)$ )	<sup>r2 -</sup> CompDose = 0	
Pre-condition The insulin reservoir contains at least the maximum allowed single dose of insulin.	Sugar level increasing and rate increase stable or increasing $((r2 - r1) \ge (r1 - r0))$	of CompDose = round ((r2 – r1)/4) If rounded result = 0 then CompDose = MinimumDose	
Post-condition r0 is replaced by r1 then r1 is replaced by r2.			
Side effects None.	15.02.04		14
Summary			
<ul> <li>The software requirements document is <ul> <li>an agreed statement of the system requirements.</li> <li>organized so that both system customers and software developers can use it.</li> </ul> </li> <li>Often written in natural language with diagrams <ul> <li>Numbered sentences;</li> <li>Conforming to uniform style.</li> </ul> </li> </ul>			
15-02-04 15			