

# Requirements Document

Chapter 4.2-4.3  
Slides #11

CMPT 276

© Dr. B. Fraser

Based on slides from Software Engineering 9<sup>th</sup> ed, Sommerville.

# Topics

---

- 1) What is a requirements document (req. doc.)?
- 2) How can we write requirements?

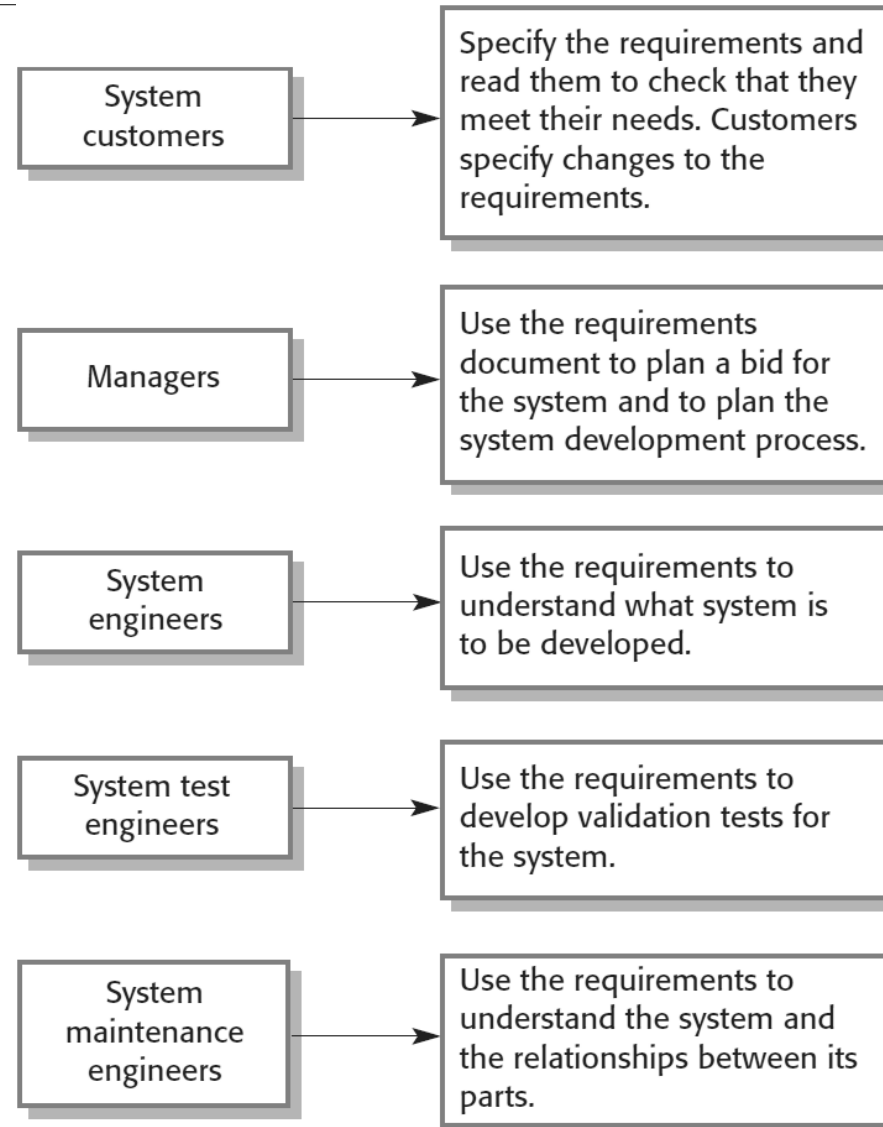
# Software requirements document

- Software requirements document:
  -
- Can include both:
  - user requirements and
  - system requirements.



# Users of a requirements document

- Req. doc. mainly part of a plan-driven method
  - Agile methods argue requirements change too quickly for a req. doc. to be useful.
  - Large systems and critical systems...
- Many uses for a req. doc.:



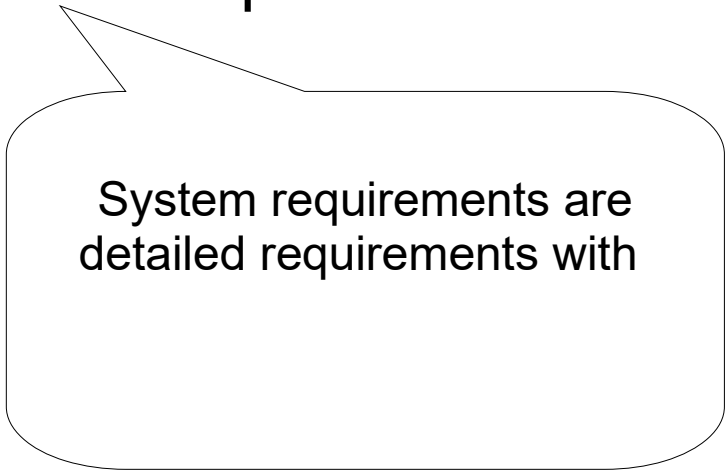
# Requirements specification process

# Requirements specification

- Requirements specification is
  - process of writing a requirements document.
  - Includes the user and system requirements.



User requirements must be



System requirements are detailed requirements with

- Important to be as complete as possible:
  - may be the basis for system development contract.

# Guidelines for writing requirements

- Requirements often written in natural language: sentences, diagrams, and tables.
  - Natural language used because it is..
  - Understood by customers and developers.
- Use language in a consistent way.
  - Use “shall” for...
  - Use “should” for...
- Avoid computer jargon: use domain terminology
- Include an explanation (rationale) of ..

# Exercise

- Write two natural language requirements for an automatic transmission shifter





# Problems with natural language

- Lack of clarity
  -
- Requirements confusion
  - Functional and non-functional requirements tend to be mixed-up.
- Requirements amalgamation
  - If several different requirements are expressed together.
- See **the** cookie recipe: MIL-C-44072C
  - Ex: Sections 3.2.2, 3.3.11

# Ways of writing a system req. specification

Notation

Natural language sentences

Mathematical specification

Description

Requirements are...  
in plain English.

Diagrams and text to describe the system.  
Ex: UML

Ex:...  
Unambiguous, but hard for customers to understand.

# Structured spec.: an insulin pump

## Action

- CompDose is zero if the sugar level is **stable or falling** or if the level is increasing but the **rate of increase is decreasing**.
- 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.
- If the **result is rounded to zero** then CompDose is set to the minimum dose that can be delivered.

## Requirements

Two previous readings so that the rate of change of sugar level can be computed.

## Pre-condition

The insulin reservoir contains at least the maximum allowed single dose of insulin.

Post-condition      r0 is replaced by r1 then r1 is replaced by r2.

Side effects          None.

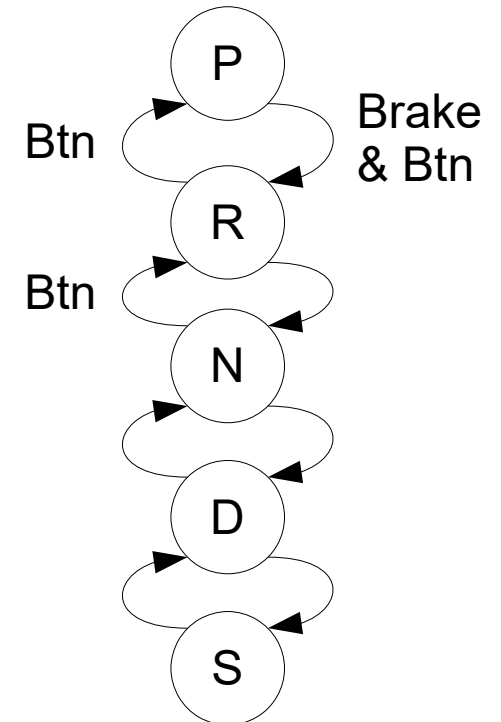
# Tabular specification: Insulin pump

<b>Condition</b>	<b>Action</b>
Sugar level falling ( $r_2 < r_1$ )	CompDose = 0
Sugar level stable ( $r_2 = r_1$ )	CompDose = 0
Sugar level increasing and rate of increase decreasing ( $(r_2 - r_1) < (r_1 - r_0)$ )	CompDose = 0
Sugar level increasing and rate of increase stable or increasing ( $(r_2 - r_1) \geq (r_1 - r_0)$ )	CompDose = round $((r_2 - r_1)/4)$ If rounded result = 0 then CompDose = MinimumDose

# Exercise

- Write a tabular specification for the shifter's state changes

Current Gear	Shift 'Up'	Shift 'Down'	Notes
P			
R			
N			
D			
S			



# Summary

- The software requirements document is
  - an agreed statement of the system requirements.
  - organized so that both system customers and software developers can use it.
- Often written in natural language with diagrams
  - Numbered sentences;
  - Conforming to uniform style.