



# Software Engineering and Project Management

## INTRODUCTION

**Davide Vega D'Aurelio**

[davide.vega@it.uu.se](mailto:davide.vega@it.uu.se)

**Postdoctoral Researcher**

Division of Computing Science

Department of Information Technology

Uppsala University InfoLab, Sweden



## Proposed plan

- Random story to catch students attention (for true) [10']
- Software Engineering as a process [35']
- Break [15']
- V&V of functional requirements [30']
- Discovering activity [15']



## Part II

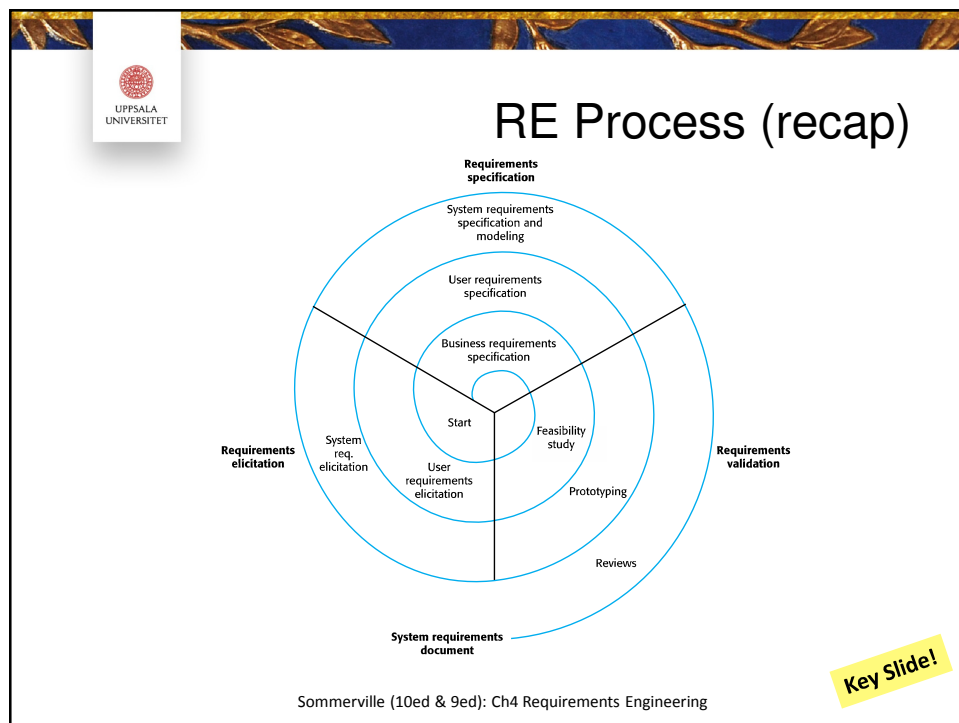
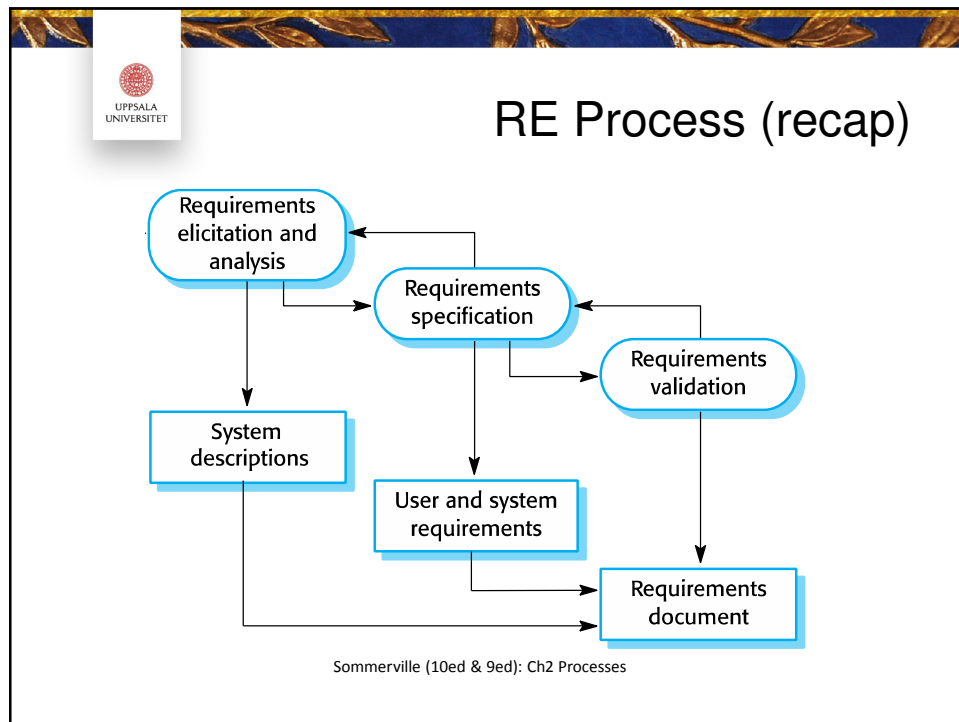
### V&V OF FUNCTIONAL REQUIREMENTS



### Requirements Engineering (recap)

Requirements are about **WHAT**,  
**not** HOW

Key Slide!





# User vs. System Requirements

Q: What is a cuckoo clock?

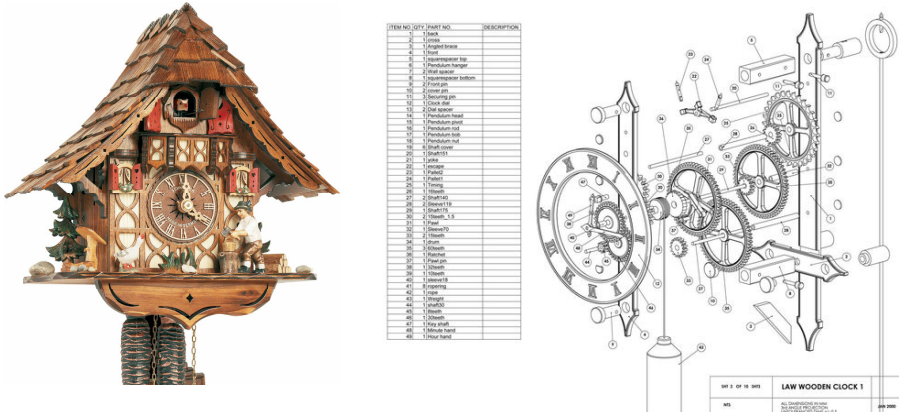


# User vs. System Requirements



UPPSALA UNIVERSITET

## User vs. System Requirements



The image illustrates the difference between user and system requirements using a cuckoo clock. On the left is a photograph of a finished cuckoo clock, representing the user's perspective. In the center is a table listing the parts of the clock, representing the system requirements. On the right is an exploded view diagram of the clock's internal mechanism, showing the gears and components that make it work.

ITEM NO.	QTY	PART NO.	DESCRIPTION
1	1	1	Case
2	1	2	Case base
3	1	3	Case top
4	1	4	Case back
5	1	5	Case front
6	1	6	Case side
7	1	7	Case bottom
8	1	8	Case top
9	1	9	Case back
10	1	10	Case front
11	1	11	Case side
12	1	12	Case bottom
13	1	13	Case top
14	1	14	Case back
15	1	15	Case front
16	1	16	Case side
17	1	17	Case bottom
18	1	18	Case top
19	1	19	Case back
20	1	20	Case front
21	1	21	Case side
22	1	22	Case bottom
23	1	23	Case top
24	1	24	Case back
25	1	25	Case front
26	1	26	Case side
27	1	27	Case bottom
28	1	28	Case top
29	1	29	Case back
30	1	30	Case front
31	1	31	Case side
32	1	32	Case bottom
33	1	33	Case top
34	1	34	Case back
35	1	35	Case front
36	1	36	Case side
37	1	37	Case bottom
38	1	38	Case top
39	1	39	Case back
40	1	40	Case front
41	1	41	Case side
42	1	42	Case bottom
43	1	43	Case top
44	1	44	Case back
45	1	45	Case front
46	1	46	Case side
47	1	47	Case bottom
48	1	48	Case top
49	1	49	Case back
50	1	50	Case front
51	1	51	Case side
52	1	52	Case bottom
53	1	53	Case top
54	1	54	Case back
55	1	55	Case front
56	1	56	Case side
57	1	57	Case bottom
58	1	58	Case top
59	1	59	Case back
60	1	60	Case front
61	1	61	Case side
62	1	62	Case bottom
63	1	63	Case top
64	1	64	Case back
65	1	65	Case front
66	1	66	Case side
67	1	67	Case bottom
68	1	68	Case top
69	1	69	Case back
70	1	70	Case front
71	1	71	Case side
72	1	72	Case bottom
73	1	73	Case top
74	1	74	Case back
75	1	75	Case front
76	1	76	Case side
77	1	77	Case bottom
78	1	78	Case top
79	1	79	Case back
80	1	80	Case front
81	1	81	Case side
82	1	82	Case bottom
83	1	83	Case top
84	1	84	Case back
85	1	85	Case front
86	1	86	Case side
87	1	87	Case bottom
88	1	88	Case top
89	1	89	Case back
90	1	90	Case front
91	1	91	Case side
92	1	92	Case bottom
93	1	93	Case top
94	1	94	Case back
95	1	95	Case front
96	1	96	Case side
97	1	97	Case bottom
98	1	98	Case top
99	1	99	Case back
100	1	100	Case front

EXPLODED VIEW OF LAW WOODEN CLOCK 1

UPPSALA UNIVERSITET

## User vs. System Requirements

- **User**
  - Abstract
  - Natural Language
  - What users want
- **System**
  - More concrete, detailed
  - Natural + Formal language
  - What system provides
  - Used as contract or product description



## A Restaurant example

- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• <b>User</b></li> <li>– Food should look good</li> <li>– Food should taste good</li> <li>– I want enough food</li> <li>– I want fish</li> </ul> | <ul style="list-style-type: none"> <li>• <b>System</b></li> <li>– Chicken soup</li> <li>– Baked salmon with ...</li> <li>– Chocolate mousse</li> </ul> |
|---|--|




## A Software example

### User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

### System requirements specification

- 1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
- 1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
- 1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
- 1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc) separate reports shall be created for each dose unit.
- 1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.




## Functional vs. Non-functional Requirements

- **Functional**
  - what the system should do
- **Non-functional**
  - constrains on the system

Diagram illustrating the relationship between requirements and their constraints:

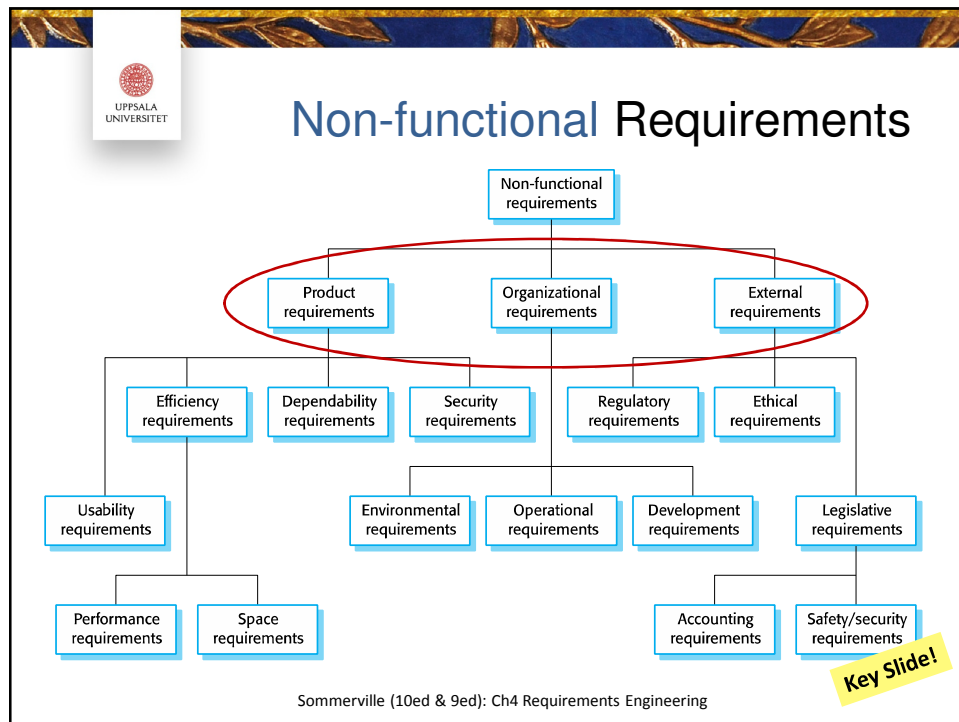
- Functional requirements lead to **Product behaviour**.
- Non-functional requirements lead to **Process** (constrains on the development process).

Key Slide!



## Functional Requirements

- Data operations, presentation...
- Domain requirements: *“everyone knows that...”*
- **Complete**
  - All desired functionality is covered
- **Consistent**
  - They do not contradict each other




UPPSALA UNIVERSITET

## Metrics for Non-functional Requirements

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems





UPPSALA  
UNIVERSITET

# Questions

# ?



UPPSALA  
UNIVERSITET

## Part II.b

VERIFICATION, VALIDATION, INSPECTION



## Requirements checklist (Verification)

- **Understandable** *(properly explained)*
- **Valid** *(is this required, rationale)*
- **Verifiable** *(test cases included)*
- **Realistic** *(feasibility, prototype)*
- **Complete** *(all cases covered, exceptions)*
- **Consistent**
- **Traceable**

Key Slide!



## Requirements checklist (Verification)

- **Valid**
  - *Real users*
- **Realistic**
  - *On time / Under budget / Current tech*
- **Complete**
  - *Is all functionality captured? / Are all cases covered?*
- **Consistent**
  - *Internal to requirements*
- **Verifiable**
  - *Can requirements be tested?*



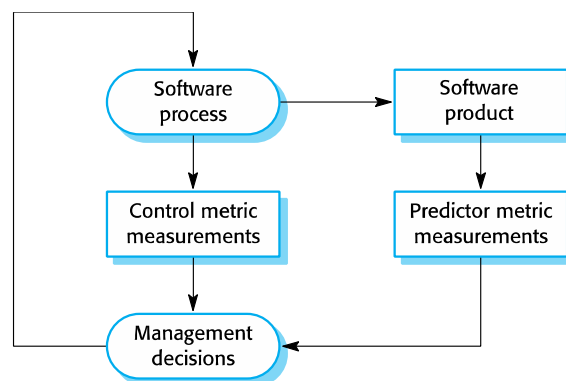
## How do we get the requirements right? (*Validation*)

- **User review of**
  - *Requirements*
  - *Models*
  - *Prototype*
  - *Test cases*
- **Comparison with similar systems**

Key Slide!



## Measurement Feedback

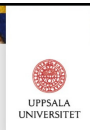


Sommerville (10ed & 9ed): Ch24 Quality management




## Measurement Process

- **Choose measurements**
  - *Goal → Question → Metric (GQM)*
- **Select components**
- **Measure**
- **Identify anomalous values**
  - *Compare to normal product / company values*
- **Analyze anomalous components**



## Process measurement


- **Resources required**
  - *Time*
  - *Money*
- **Occurrence of events**
  - *Failed system builds*
  - *Missed deadlines*
  - *Missing process documents*



UPPSALA  
UNIVERSITET

# Questions

# ?



UPPSALA  
UNIVERSITET

## Part II.c


CHECK ACTIVITY

## Example 1



## Example 2





UPPSALA  
UNIVERSITET

# Questions

# ?