

What is a Good Requirement Specification?

by *Gerrit Muller* University of South-Eastern Norway-NISE

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

Abstract

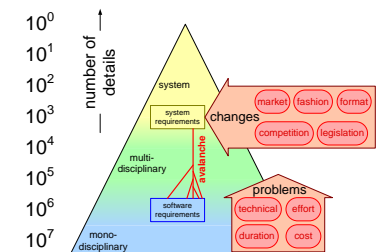
Requirements play a driving role during product creation. The requirements are captured in a requirements specification. How can we assess the requirements specification? What are the criteria for a good specification?

We discuss these aspects by positioning the requirements specification in the broader context of customers, market, product creation and product life-cycle. We zoom in to the software requirements specification, to discuss the criteria for this mono-disciplinary specification.

Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

September 6, 2020
status: concept
version: 0.2



Did you ever program a VCR?

A



depressed

B



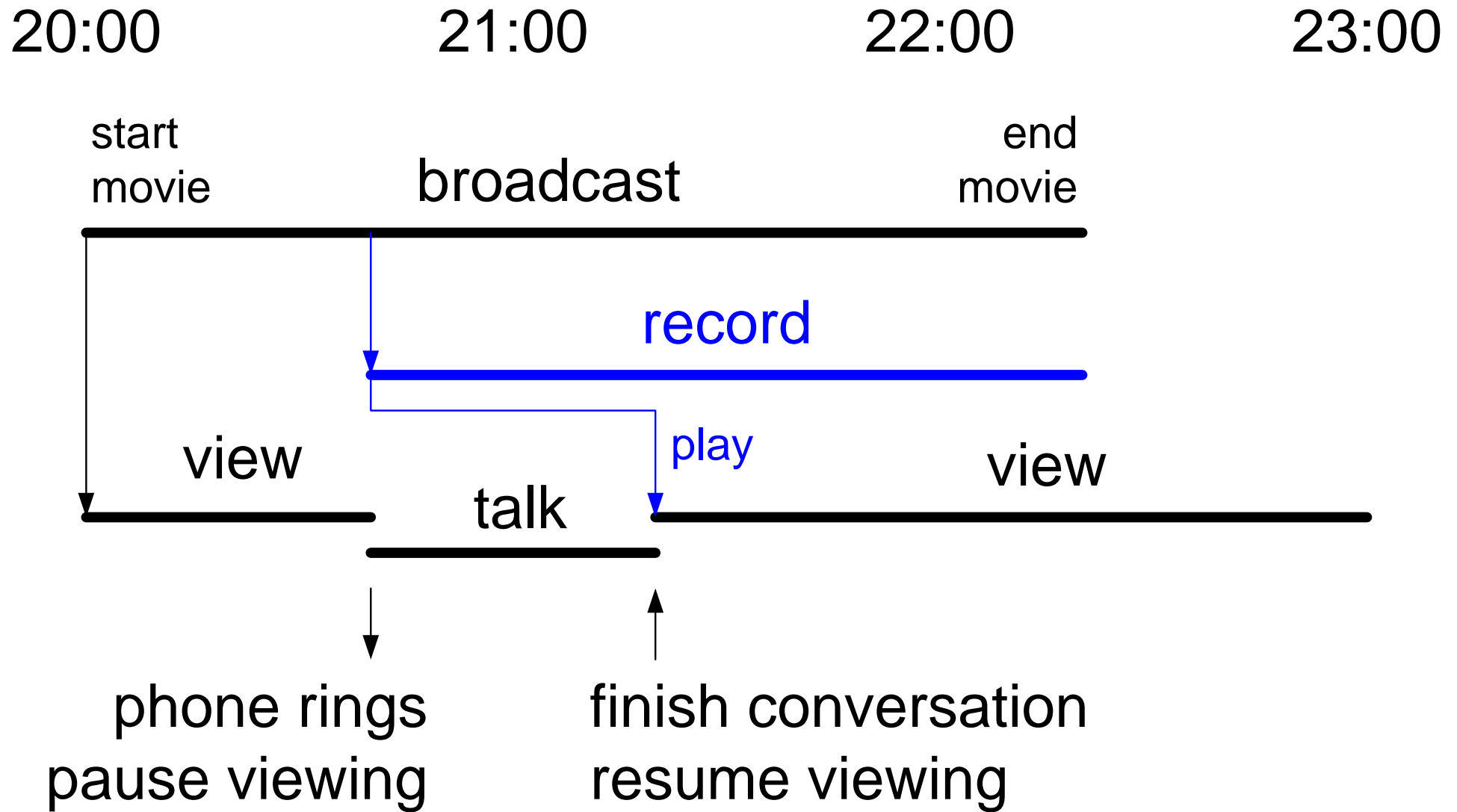
desperate

C



hysteric

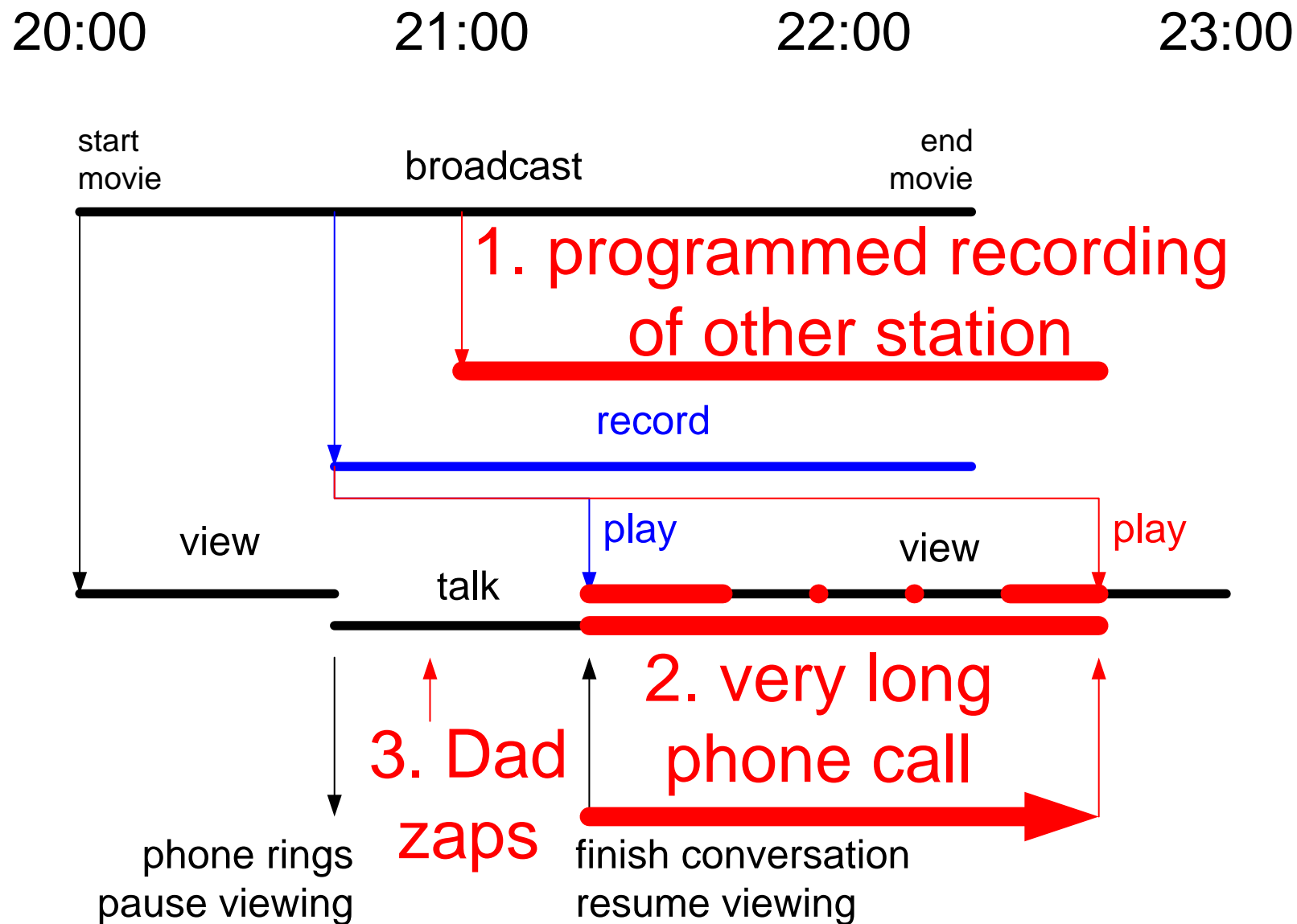
Example Time Shift recording



Construction limits intrude in User Experience

- number of tuners
- number of simultaneous streams (recording and playing)
- amount of available storage
- management strategy of storage space


What if?



Visual Basic Prototype:
enables "experiencing"

Requirements specification
Many tables, mostly addressing details

- 2.1.1 Real-time data requirements
- 2.1.2 Implementation detail
- 2.1.3 Non-real time data requirements



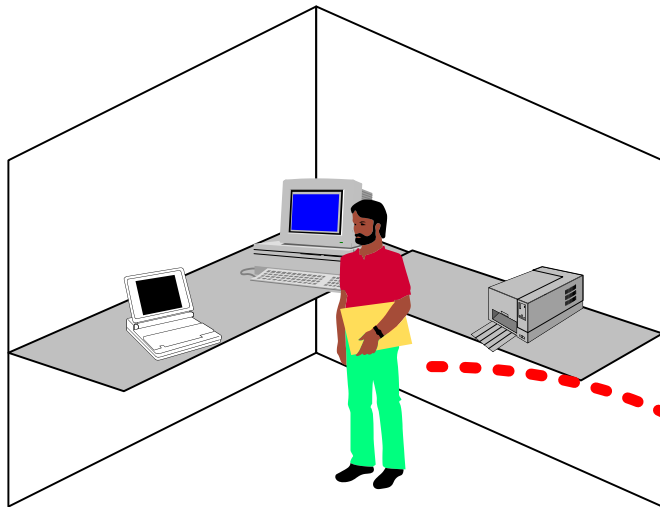
- play
- pause
- record
- EPG

1.1 Software Requirements	
1.1.1 Real-time data requirements	<p>1.1.1.1 Access to the non-real-time data must be done in such a way that it does not interfere with the real-time data</p> <p>1.1.1.2 There must be no disruptions in output of video signal during the operation of VCR</p> <p>1.1.1.3 Responsiveness for non real-time data is less than 150ms (the time for writing a block on HDD) for 2KB of non-video data</p>
1.1.2 Implementation detail	<p>1.1.2.1 Management of HDD content must only be possible through the TOC in order to prevent unauthorized access to content of HDD</p> <p>1.1.2.2 Visual feedback is provided to the user via On-Screen Display</p> <p>1.1.2.3 User input is provided via the RC</p>
1.1.3 Non-real time data requirements	<p>1.1.3.1 User must be able to pause and un-pause a title, played from HDD, while (s)he is watching it</p> <p>1.1.3.2 User can jump forward and backward in a title, from HDD, during watching of this title</p> <p>1.1.3.3 Names of titles should be derived from the information from the EPG (name of the program to be recorded, time and date of registration)</p>

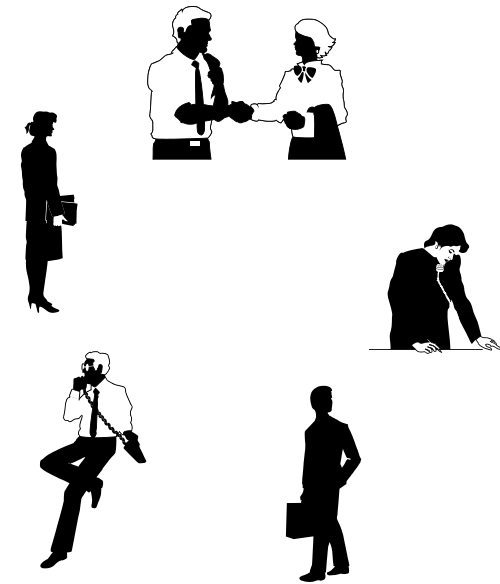
Key Success Factor: Feedback

Obtain feedback from real users:

- Observe
- (Dare to) Listen
- Experiment
- Use short development cycles

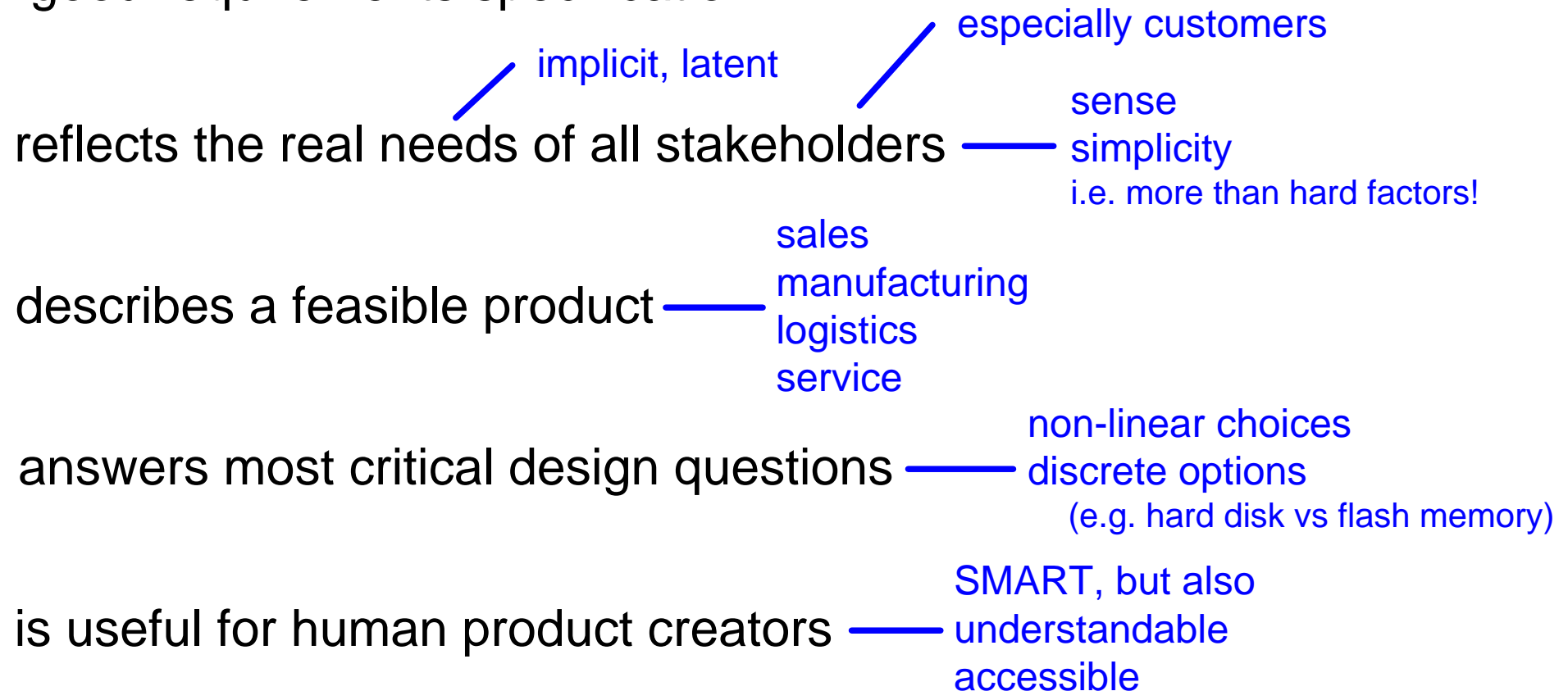


Don't stay in the development lab

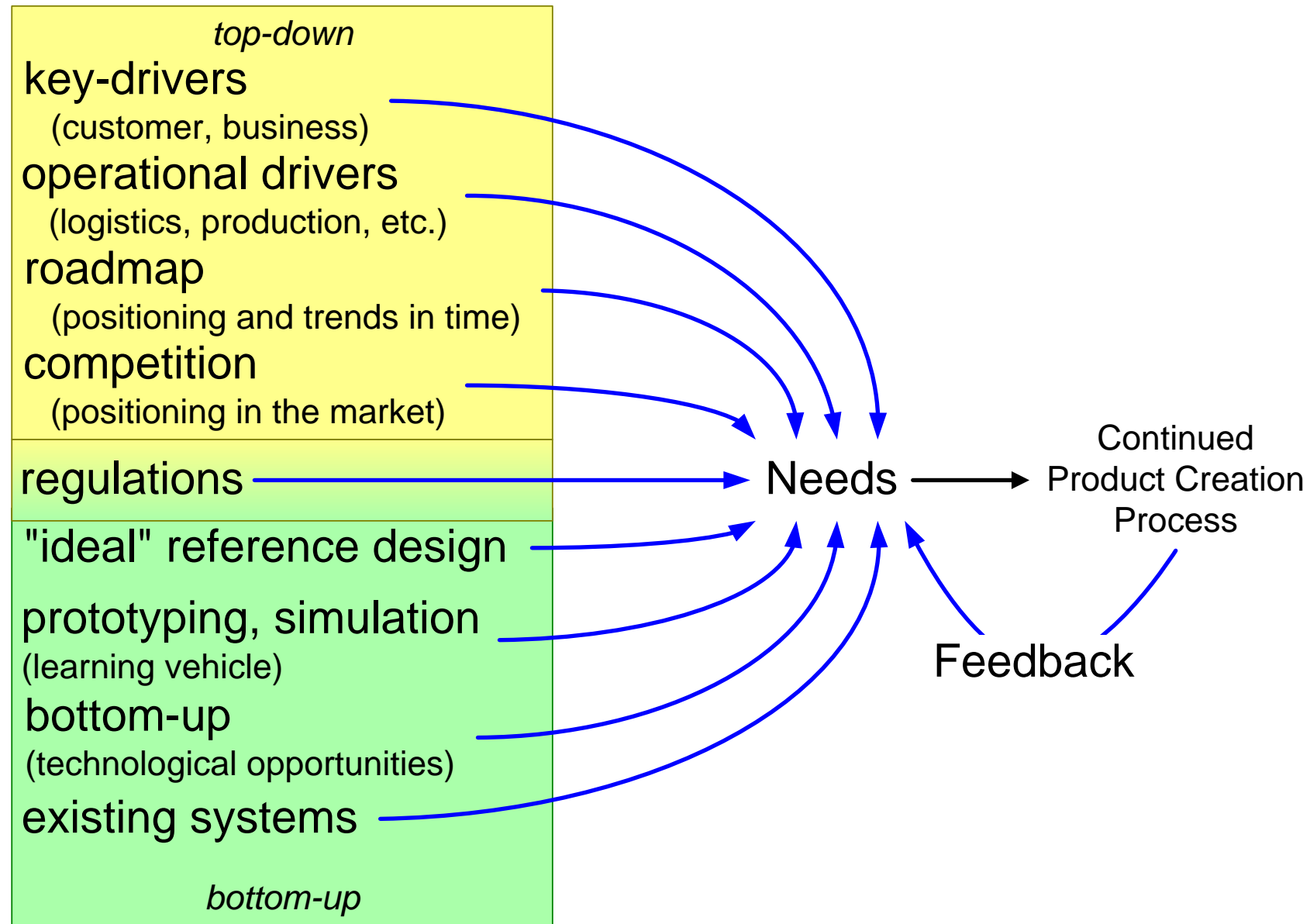


Criteria for a Good Requirements Specification

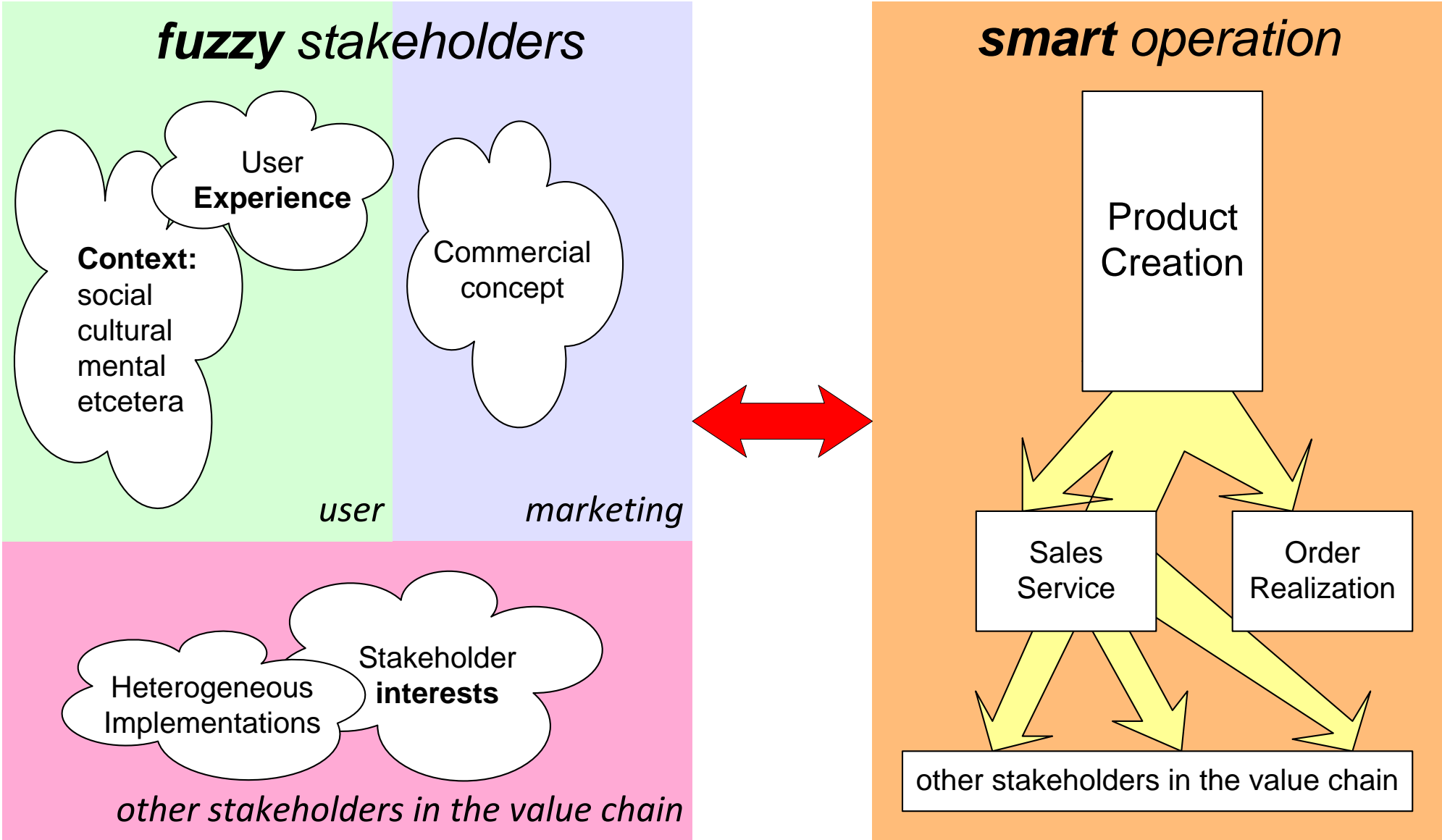
A good requirements specification:



Multiple Viewpoints to Understand Needs and Feasibility



How SMART can requirements be described?



fuzzy stakeholders

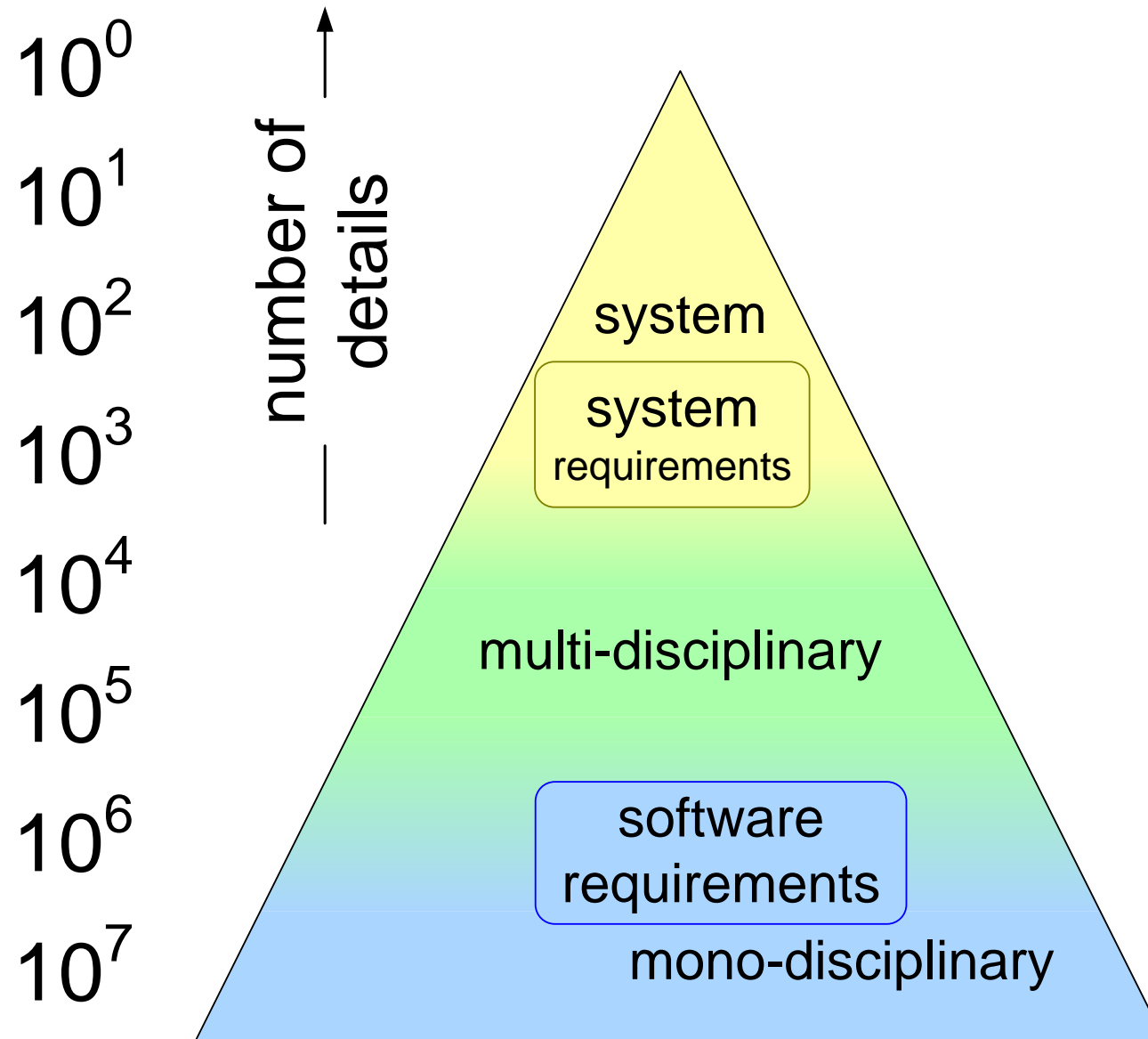
- Accessible
- Understandable
- Low threshold

smart operation

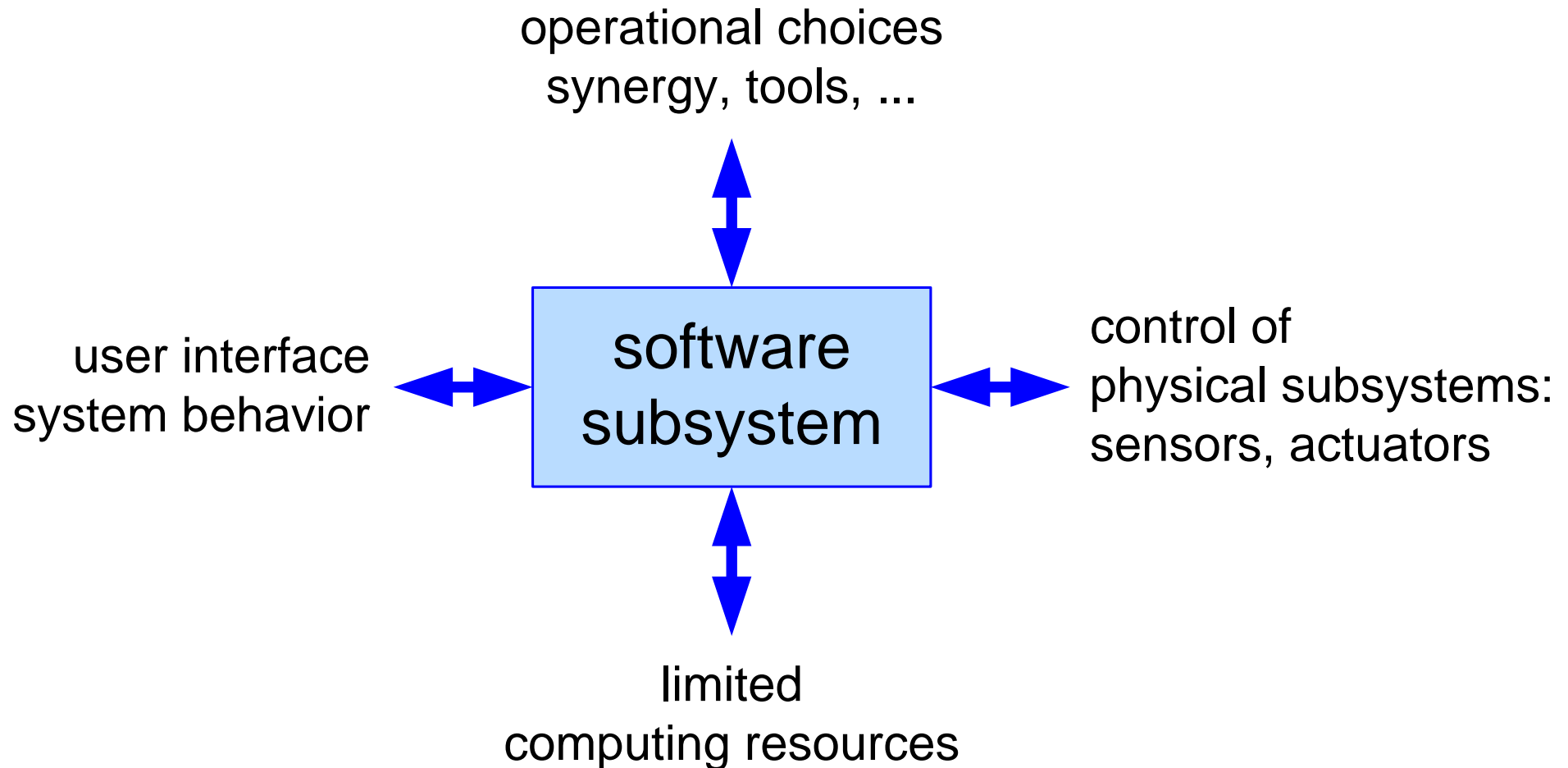
- Specific
- Unambiguous
- Verifiable
- Quantifiable
- Measurable
- Complete
- Traceable

When SW engineers demand "requirements",
then they expect *frozen* inputs
to be used for
the design, implementation and validation
of the software

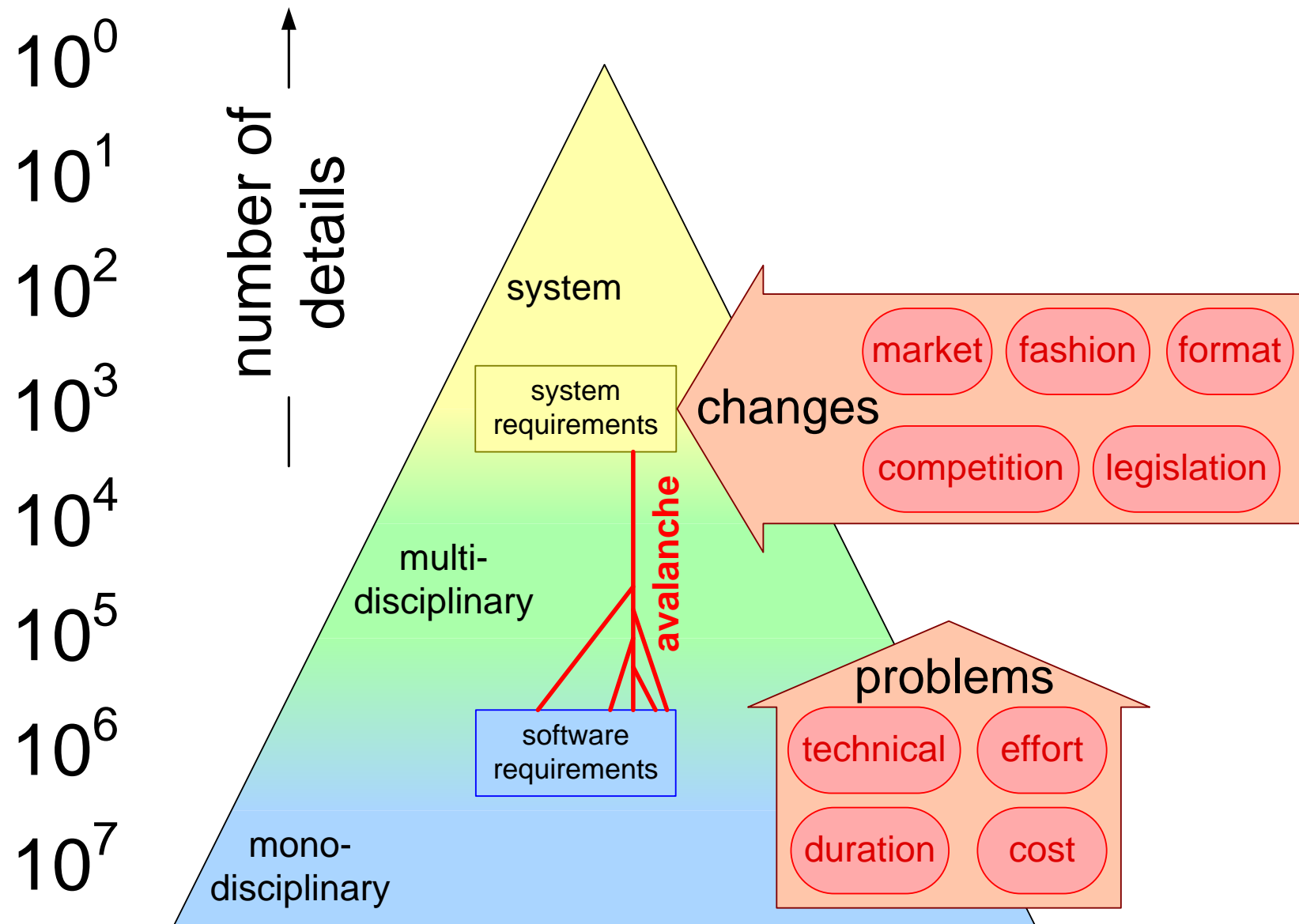
System vs Software Requirements



Why is the Software Requirement Specification so Large?



And why is it never up-to-date?



Conclusions and Recommendations

Never wait for the software requirements specification to be complete

1) it is never complete

2) it is never up-to-date

3) the product will be too late.

Be creative to cope with uncertainty and dynamics

for instance, use prototype as specification "WYSIWYG"

use incremental development strategies (XP, EVO, ...)

focus on most important and critical issues