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.
Did you ever program a VCR?

A  

B  

depressed

C  

desperate

hysteric
Example Time Shift recording

20:00  21:00  22:00  23:00

- start movie
- broadcast
- end movie

- record
- view
- talk
- play
- view

- phone rings
- pause viewing
- finish conversation
- resume viewing
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?

20:00

- start movie
- phone rings
- pause viewing
- view
- talk
- record
- 1. programmed recording of other station
- 2. very long phone call
- 3. Dad zaps

21:00

- broadcast
- play
- view

22:00

- end movie
- finish conversation
- resume viewing
- play

23:00

- version: 0.2
- March 6, 2013
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

<table>
<thead>
<tr>
<th>1.1 Software Requirements</th>
<th>1.1.1 Real-time data requirements</th>
<th>1.1.2 Implementation detail</th>
<th>1.1.3 Non-real time data requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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</td>
<td>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</td>
<td>1.1.3.1 User must be able to pause and unpause a title, played from HDD, while (s)he is watching it</td>
</tr>
<tr>
<td></td>
<td>1.1.1.2 There must be no disruptions in output of video signal during the operation of VCR</td>
<td>1.1.2.2 Visual feedback is provided to the user via On-Screen Display</td>
<td>1.1.3.2 User can jump forward and backward in a title, from HDD, during watching of this title</td>
</tr>
<tr>
<td></td>
<td>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</td>
<td>1.1.2.3 User input is provided via the RC</td>
<td>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)</td>
</tr>
</tbody>
</table>
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
A good requirements specification:

- reflects the real needs of all stakeholders
- describes a feasible product
- answers most critical design questions
- is useful for human product creators

especially customers

sense

simplicity

e.g. more than hard factors!

implicit, latent

sales

manufacturing

logistics

service

non-linear choices

discrete options

(e.g. hard disk vs flash memory)

SMART, but also

understandable

accessible
Multiple Viewpoints to Understand Needs and Feasibility

**top-down**
- key-drivers (customer, business)
- operational drivers (logistics, production, etc.)
- roadmap (positioning and trends in time)
- competition (positioning in the market)

**bottom-up**
- regulations
- "ideal" reference design
- prototyping, simulation (learning vehicle)
- bottom-up (technological opportunities)
- existing systems

---

Needs

Continued Product Creation Process

Feedback

What is a Good Requirement Specification?

9  Gerrit Muller

version: 0.2
March 6, 2013
REQviewpoints
How SMART can requirements be described?

**fuzzy stakeholders**
- User Experience
- Context: social, cultural, mental, etcetera
- Heterogeneous Implementations
- Stakeholder interests
- other stakeholders in the value chain

**smart operation**
- Product Creation
- Sales Service
- Order Realization
- other stakeholders in the value chain

What is a Good Requirement Specification?

version: 0.2
March 6, 2013
VREQfromFuzzyToSmart
Requirements must be SMART and Usable

**fuzzy stakeholders**
- Accessible
- Understandable
- Low threshold

**smart operation**
- Specific
- Unambiguous
- Verifiable
- Quantifiable
- Measurable
- Complete
- Traceable

What is a Good Requirement Specification?
11 Gerrit Muller
When SW engineers demand "requirements",
then they expect *frozen*inputs
to be used for
the design, implementation and validation
of the software
What is a Good Requirement Specification?

March 6, 2013

version: 0.2

Embedded Systems Institute
Why is the Software Requirement Specification so Large?

What is a Good Requirement Specification?

14 Gerrit Muller

version: 0.2
March 6, 2013
VREQsoftwareSubsystem
And why is it never up-to-date?

What is a Good Requirement Specification?

version: 0.2
March 6, 2013
VREQdynamics
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