Less requirements, more agreement: delta tree platform governance at Sonova AG

Daniel Lucas-Hirtz, Christophe Benz
System Design team, Sonova AG
Less requirements, more agreement

• Our context
  • Problem statement
  • Why agreement ?
  • How ?
  • Survey
  • Learnings
  • Discussion
Sonova Group

Broadest offering: hearing instruments, cochlear implants …

HI Hearing Instruments
- Behind-The-Ear hearing instruments (ITE)
- FM systems
- Custom In-The-Ear hearing instruments (ITE)
- Hearing protection
- Wireless communication systems
- Earphones
- Invisible extended-wear hearing instruments

CI Cochlear Implants
- Cochlear implants

… and professional retail services
In charge of platform projects.

Reusable engineering assets are packaged into **system releases**:

<table>
<thead>
<tr>
<th>Releases</th>
<th>Product families</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rio16.1</td>
<td>X1, U1</td>
</tr>
<tr>
<td>Rio16.2</td>
<td>X2, U2</td>
</tr>
<tr>
<td>Tokyo20.1</td>
<td>X3</td>
</tr>
<tr>
<td>Tokyo20.2</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Each system release enable **product families**: 

- Rio16.1: X1, U1
- Rio16.2: X2, U2
- Tokyo20.1: X3
- Tokyo20.2: ...
- ...
Example system context
Less requirements, more agreement

• Our context
• Problem statement
• Why agreement ?
• How ?
• Survey
• Learnings
• Discussion
Why system requirements?

- Product managers
  -> drive the product's **vision**
- Project managers
  -> **plan** costs, dates and risks
- Developers
  -> understand (**new**) features
- V&V team
  -> test the (**complete**) system
- QA
  -> check **compliance**
- All
  -> **reuse** features
Over-documenting
“Requirements management is a means to an end; it is not a goal. The goal is to build (or acquire) a system that solves some real problem or leverages some real business opportunity. There is no shortage of ways that we can spend enormous amounts of our time performing requirements management. We need to develop the discipline of avoiding such quagmires”.

Alan Davis, “Just Enough Requirements”, 2004
Complexity

Minitutorial by Björn Regnell at www.re11.org
Sonova AG
Less requirements, more agreement

• Our context
• Problem statement
• Why agreement?
• How?
• Survey
• Learnings
• Discussion
Why system requirements?

- Product manager -> drive the product's vision
- Project manager -> plan costs, dates and risks
- Developers -> understand (new) features
- V&V team -> test the (complete) system
- QA -> check compliance
- All -> reuse features

Essence -> agreement

- Agree on high level scope
- Agree on a plan (release scope)
- Agree on work package scope
- Agree on externally observable characteristics
- Agree on risks & their mitigation
- Agree on planed variability
RE team in System Design has made agreement the core of its purpose, with a mission statement close to:

“Purpose of RE team is to **facilitate and document agreement on** what the platform project commits to deliver, and to maintain **this documentation up to date**.”
Less requirements, more agreement

• Our context
• Problem statement
• Why agreement?
• How?
• Survey
• Learnings
• Discussion
“Facilitate & document agreement”

-> agreement is **incremental**

---

Figure 1.1: Overview and classification of RE decisions in MDSPD

*In: 2010-Fogelstrom-Understanding_and_supporting_requirements_engineering_decisions_in_market-driven_software_product_development*
“Facilitate & document agreement”

-> agreement is **incremental**

**Scope of Requirements Engineering for platform projects**
1. Release planning: the “delta tree”

- Portfolio mgmt.
- Roadmapping
- Release planning

incrementally scope each release (i.e. list of deltas)
1. Release planning: the “delta tree”
1. Release planning: the “delta tree”

Release i = 
Base release +
Sum (delta i) +
Sum (change requests j) +
[variant definition]
2. Delta scope definition

- Portfolio mngt.
- Roadmapping
- Release planning
- Delta scoping

Incrementally scope each delta
What is a delta

A Delta is a unit of commitment to deliver value in a release (similar to “Product Backlog Item”).

A delta is
- a change in the solution to be built
- that has an externally observable benefit
- that is to be committed in a release

More information in: “The Mythical Adequate Level of Details - Bundling Requirements for Large Scale Market Driven Engineering”:
Incremental agreement

"a delta (unit of commitment) has an incremental maturity throughout the development process".
Incremental agreement

Change control:
Delta scope change is controlled by CCB (change control board) once delta maturity level has reached “10-Assigned”.

Delta shall not move to “13_Specified” unless System Specifications document is approved.

Change control:
System Specifications change is controlled by CCB once approved.
Less requirements, more agreement

- Our context
- Problem statement
- Why agreement?
- How?
- Survey
- Learnings
- Discussion
Survey
Interviewees

- 4 Project leaders products
- 2 Project leaders platforms
- 3 Architects
- 3 Verification engineers
- 2 Product managers
- 2 Developers
- 2 Experts

Total 18 interviewees
Survey Question

As a product manager, Project leader, Developer, Verification engineer, Quality engineer, Sonova employee have my goals been easier to reach with Project “Rio16” in comparison with previous project “Peking08”?

Rating: harder = 0, easier = 5
Survey Results

5 = easier with Rio16      ------ = Neutral      Total 18 interviews
## Key feedback

<table>
<thead>
<tr>
<th>+/−</th>
<th>Feedback</th>
<th>Roles</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Easy to get an overview of project scope</td>
<td>All</td>
<td>Delta tree and incremental scoping helps</td>
</tr>
<tr>
<td>+</td>
<td>Early involvement of different stakeholders</td>
<td>AR, EX, DE</td>
<td>High level scoping forces early commitment</td>
</tr>
<tr>
<td>+</td>
<td>Increased quality of requirements</td>
<td>VE, PL, DE</td>
<td>Less requirements - kept up to date</td>
</tr>
<tr>
<td>+</td>
<td>More shared information</td>
<td>VE, DE</td>
<td>Delta tree as a living entry point in the project</td>
</tr>
<tr>
<td>-</td>
<td>Unclear relation between product and system</td>
<td>All</td>
<td>Organizational issues not solved</td>
</tr>
<tr>
<td>-</td>
<td>Decision-making power unclear</td>
<td>All</td>
<td>Authority &amp; organization</td>
</tr>
<tr>
<td>-</td>
<td>Process and concept not to everyone clear</td>
<td>PL, DE</td>
<td>Process transition</td>
</tr>
</tbody>
</table>
Less requirements, more agreement

• Our context
• Problem statement
• Why agreement?
• How?
• Survey
• Learnings
• Discussion
Learning 1 – scope as late as possible

Scoping a delta too early is wasted effort:
• scope with the team, only when the team is about to start
• which calls for a higher level unit of front end planning

A possible model (ideas for the future):
• “Themes” as unit of innovation roadmap (X years)
• Features (or “Delta”) as unit of release planning (1 year)
• finer units of system integration (“e.g. “epics” – 2 months), of sprint planning (“User story” – 3 weeks), and of daily planning (“Tasks”).
Learning2: relative versus absolute requirements
Less requirements, more agreement

• Our context
• Problem statement
• Why agreement?
• How?
• Survey
• Learnings
• Discussion
Thank you
References:

7. S. Bühne, K. Lauenroth, K. Pohl, Why is it not Sufficient to Model Requirements Variability with Feature Models? AURE04, 2004
11. D. Lucas-Hirtz, "Practical requirements reuse - How to initiate improvement in software requirements reuse practices ?" REFSQ 2011