Delta tree visualization at Phonak AG

Problem statement: how to extract a one page view of relevant release planning information out of a work item based requirements management tool.

Daniel Lucas-Hirtz, Phonak AG, Stäfa, Switzerland
daniel.lucas-hirtz@phonak.com
26/08/2014, IWSPM Workshop,
22nd IEEE International Requirements Engineering Conference,
Karlskona, Sweden
Daniel.Lucas-Hirtz@phonak.com
Sonova Group

Broadest offering: hearing instruments, cochlear implants ...

HI Hearing Instruments

- Behind-The-Ear hearing instruments (BTE)
- 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
Abstract—At Phonak we are documenting the governance of an hearing device product line thanks to “deltas” (i.e. “units of decisions”) which are committed to “releases”. These release planning items are documented within a requirements management tool.

We are maintaining a manual overview of this release plan on a single A3 page. We think this manual export is not scalable, and we are looking for an automatic export.

[..]
Governance artefacts

Idealized. Only partly matches Phonak R&D.
Governance processes

Roadmapping (Portfolio)

Release planning

Change control board

Idealized. Only partly matches Phonak R&D.
## Governance boards

### Roadmapping (Portfolio)

<table>
<thead>
<tr>
<th>Focus</th>
<th>Main artefacts</th>
<th>In charge of:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategic decisions: Go/ no go for release Ri</td>
<td>Releases</td>
<td>- New release requests (ex. Rio, Elvis2.0, Crostino)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Major release changes/drops</td>
</tr>
</tbody>
</table>

### Release planning

- Tactical decisions: $R_i = \sum_i \delta + R_{base}$

<table>
<thead>
<tr>
<th>Focus</th>
<th>Main artefacts</th>
<th>In charge of:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delta requests</td>
<td>Deltas (or “Product Backlog Items”)</td>
<td>- New delta requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Delta re-scope requests (change/split/merge) (within same release)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Delta de-scope (move another release / drop)</td>
</tr>
</tbody>
</table>

### Change control board

- Operational decisions: $R_i \in \sum_i C_r$

<table>
<thead>
<tr>
<th>Focus</th>
<th>Main artefacts</th>
<th>In charge of:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Requests and Problem reports</td>
<td></td>
<td>- Non major change requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Problem reports on the product (i.e. “product bug”)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Problem reports on the artefacts (ex. spec error)</td>
</tr>
</tbody>
</table>
Reusable (“Platform”) versus product releases
(i.e. “Domain Engineering versus application engineering” according to the SW Product Line Eng. literature).

<table>
<thead>
<tr>
<th>Release</th>
<th>Defined by</th>
<th>Which means:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reusable Release “i”</td>
<td>$RR_i = RR_{base} + \sum_i \delta + \sum_i Cr$</td>
<td>A reusable release “i” (“$RR_i$”, ex. “Elvis 2.0”) is defined by a base reusable release (ex. “Elvis 1.0”), a set of deltas (those committed to $RR_i$), and a set of change requests (those committed to $RR_i$). Not that a reusable release has a set of variation points (the “variability model” of the RR).</td>
</tr>
<tr>
<td>Product Release “j”</td>
<td>$PR_j = RR_{base} + \sum_j \text{Variant on base} + \sum_j \delta + \sum_j Cr$</td>
<td>A product release “j” (“$PR_j$”, ex. “Rio”) is defined by a base reusable release (ex. “Elvis 1.0”), the variant model of the base release (i.e. the chosen variant for each variation points of the variability model or the base release) a set of deltas (those committed to $RR_j$), and a set of change requests (those committed to $RR_j$).</td>
</tr>
</tbody>
</table>
Engineering governance - big picture
Delta tree implemented with Polarion tool – each icon is a delta:

Benefits:
- traceability to specs and tests
- history / ownership
- each delta opens a one page deeper level of details

Drawback:
- no “big picture”
Delta tree represented with Visio – each box is a delta:

**Benefits:**
- “Big picture”
- Helps to think at a higher level – e.g. do release planning (ex. what if I drop this feature?)

**Drawback:** Manual documentation out of the RM database:
- Error prone,
- Time consuming,
- Risk of obsolescence
“Manual extraction” hurts. How about automatic extraction out of the Requirements Management database?
Attempt to automatically extract a graphical representation with a graph visualization tool – Gephi.
Any idea to do better?

Abstract—At Phonak we are documenting the governance of an hearing device product line thanks to “deltas” (i.e. “units of decisions”) which are committed to “releases”. These release planning items are documented within a requirements management tool.

We are maintaining a manual overview of this release plan on a single A3 page. We think this manual export is not scalable, and we are looking for an automatic export. Our test to automate this visual representation thanks to a graph visualization tool has failed.

Our question therefore can be summarized as “how can we automate the visualization of the relevant release planning information out of a RM tool so that we can facilitate release planning”.

We would be very glad to benefit from the feedback from the Software Product Management community on this question, and at the same time would be glad to exchange and feed the “release planning” domain of this community with our field experiences and issues.
REFERENCES

10. D. Lucas-Hirtz, "Practical requirements reuse - How to initiate improvement in software requirements reuse practices?" REFSQ 2011