How to assign cost to “Avoidable Requirements Creep”
A step towards the waterfall’s agilization

Ralf Fahney & Dr. Stefan Puehl

Motivation - The Paper in a Nutshell

• The situation (sucks)
  − Complex fixed price projects deal inevitably with undetailed requirements.
  − Experience and theory gap: rework is a real financial challenge triggered by the SRS.
  − Risk ... planned for unavoidable creep only

• Our suggested approach
  1. Construct each iteration with articulated change
  2. Structure the plan of payments accordingly
  3. Adjust the flow of payments
  4. Plan the iteration scope with creep incorporated

• Your take-aways
  − Structure right from the start
  − Address stability and flexibility at the same time
  − Enable with the explicit requirement structure a robust customer dialogue on costs

• Our benefits
  − Agile elements are defined as managed change for transparency.
  − Waterfall – as fixed-price approach in global delivery – follows closer to customer learning and agreements.
A sample/standard fixed price project setting

Customer → 1) elaborated → RFP
3) commissioned → General contractor
4) commissioned → Subcontractor

RFP → 2) bidded jointly for

General contractor → 5) elaborated jointly → SRS

SRS → 6) accepted as RFP
req. spec. replacement

Usually late during project

How SRS elaboration is intended to work

Requirements specification

RFP → SRS iteration 1
SRS iteration 2
SRS iteration 3

Software design & implementation

Proof of concept iteration scope
Construction iteration 2 scope
Construction iteration 3 scope
Construction iteration 4 scope = Go Live
What really happens - Rework necessary but not planned

The Problem – Our suggestion

How to organize a project to avoid “avoidable requirements creep” and thereby to reduce subsequent implementation rework?

Stability
SRS acceptance step after each and every SRS iteration

Flexibility
Change request management at SRS level for each iteration
#1 Construction iteration <n> scope

Specify SRS I1 changes and assign cost to them

Result: SRS I1

Specify additional functionality as per initial plan

Accept SRS iteration 1

Construct SRS I1 scope

Specify additional functionality as per initial plan

#2 The actual flow of payments (ex. CRs)
Event-driven, spread mostly equally over time

Accept SRS iteration 1

Construct SRS I1 scope

Specify additional functionality as per initial plan

Specify SRS I1 changes and assign cost to them

Result: SRS I1

The actual flow of payments (ex. CRs)
Event-driven, spread mostly equally over time
#3 The initial plan of payments

<table>
<thead>
<tr>
<th>Payment trigger</th>
<th>Amount</th>
<th>Requirements (1/3)</th>
<th></th>
<th>Construction (2/3)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope planned to be increased by x%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total of scope achieved at the iteration end</td>
<td>20% 30% 30% 20%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Effort spent for</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements (1/3)</td>
<td>33%</td>
<td>SRS I1 SRS I2 SRS I3 SRS I4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contract signment</td>
<td>20%</td>
<td>1.33% 2.00% 2.00% 1.33%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRS delivery</td>
<td>50%</td>
<td>3.00% 4.00% 4.00% 3.00%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RFP req. section replacement acceptance</td>
<td>30%</td>
<td>2.00% 3.00% 3.00% 2.00%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Construction (2/3)</td>
<td>67%</td>
<td>CI1 CI2 CI3 CI4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contract signment</td>
<td>20%</td>
<td>2.67% 4.00% 4.00% 2.67%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation delivery</td>
<td>25%</td>
<td>3.25% 5.00% 5.00% 3.25%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User acceptance test passed</td>
<td>25%</td>
<td>3.25% 5.00% 5.00% 3.25%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IT system productive deployment</td>
<td>30%</td>
<td>4.00% 6.00% 6.00% 4.00%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change requests might apply to a maximum of x% of scope</td>
<td>0% 20% 30% 80%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Payments are updated according to CRs from rework.

Benefits come with challenges

**Customer Benefits**
- Joint proceeding on safe ground
- Negotiations take place as early as possible
- Flow of payments supports to initiate negotiations.
- Flow of payments motivates supplier to work coordinated
- Mutual confidence to build the right system
- Any type of requirements creep processed uniformly

**Challenges**
- As supplier you’re called nitpicker, killjoy, spoilsport
- Customer might not like to take responsibility
- Plan of payments might be difficult to explain to customer
- Procedure maybe not perceived as fixed price (even though still NOT "time & material")
- Challenge to identify core and self-contained functionality is not addressed by cost only
- Maybe difficult to convince that creep will occur anyway
Conclusion

• Apply Change Request Management to SRS artifacts in the fixed price frame of reference and work for incremental acceptance

• Watch out, plan and raise discussion as early as possible
  – Seems to be counter-intuitive because everything seems to be defined.
  – But: Customers appreciate to leverage iteration scope transparency.

• Our progress:
  – CR are never discussed in the beginning of a project; RE can change that.
  – Step to converge waterfall and agile in a global service delivery setting
  – But: further domain questions remain

Thank you

stefan_puehl@dell.com
rf@fahney.com