Three-step Decision Framework for Planning Software Releases

José del Sagrado, Isabel M. del Águila, and Alfonso Bosch
Dpt. Informatics. University of Almería (Spain)

Speaker: Alfonso Bosch – abosch@ual.es
Lecturer at Informatics Dept. in the University of Almería since 1988.
Topics: Database Administration, Intelligent Systems, Business Intelligence, Data Viz
Research Group: Data, Knowledge & Software Engineering
Research Activity: Artificial Intelligence & Information Systems, applied to real problems (agriculture, energetic efficiency, software engineering).
Results: software systems for greenhouse design, decision support systems for intensive cultivations and olive grove. (registered at intellectual property office).
Scientifical Production: Above 60 contributions, including international and national magazines and workshops.
Actual Projects: Renewal of Software Based Software Engineering Spanish Network.
Management Positions: Vice principal of the Politechnical Engineering School, Director of the Teaching Organization Secretariat, Secretary of the Politechnical Engineering School and Coordinator of the Degree in Computer Engineering
Index

• 1. Introduction
• 2. Software Release Planning Framework
• 3. Case Study
• 4. Appraisal of the Framework
• 5. Conclusions and Future Work
I. Introduction

- Systems are large and complex, with interconnections to similar applications
- Worldwide development: software developers, designers, testers, project managers, and users
- Requirement Engineering needs to be dynamic and collaborative (selecting in/out requirements)
  - Decision Support Tasks based on risks, costs and benefits
  - Timelines, Dependencies and Constraints
- Assessment made by implied stakeholders
- ¿What is the best solution?
I. Introduction

- Three questions
  - Who assesses the attributes of the requirements?
  - What is the best set of requirements?
  - Do we have an agreement to build the release?

- Three separate processes framework:
  - stakeholder identification
  - elicitation of candidate requirement sets
  - next release

- Architecture proposal -> Case Study
- Discussion of limitations and scope
Index

• 1. Introduction
• 2. Software Release Planning Framework
• 3. Case Study
• 4. Appraisal of the Framework
• 5. Conclusions and Future Work
2. Software Release Planning Framework

Workflow division into three independent and connected stages

<table>
<thead>
<tr>
<th>Stakeholders identification</th>
<th>Elicitation of candidate requirement sets</th>
<th>Next release agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collect / Estimate stakeholder's salience</td>
<td>Collect / Estimate requirement's attributes</td>
<td>Caculate quality indicators for the solutions</td>
</tr>
<tr>
<td>Rank / Group stakeholder according to salience</td>
<td>Set up optimization problem</td>
<td>Make analysis and pairwise comparison</td>
</tr>
<tr>
<td>Select the representative stakeholders</td>
<td>Obtaining Pareto front as problem solution</td>
<td>Negotiate the next release goal</td>
</tr>
</tbody>
</table>
Stakeholder Identification

Stakeholder Candidate Set \( \text{Stk} = \{sk_1, sk_2, \ldots, sk_q\} \)

Salience: power, legitimacy and urgency (interviews)

\( w_{pij}, w_{lij}, w_{uij} \): values given by interviewee \( i \) to \( sk_j \)

\[
\begin{align*}
p_j &= \sum_{i=1}^{h} w_{pij}, \\
l_j &= \sum_{i=1}^{k} w_{lij}, \\
u_j &= \sum_{i=1}^{q} w_{uij}.
\end{align*}
\]

Different strategies to select the most influential: clustering or weighting

Result: set of \( m \) stakeholders allowed to propose the requirements
2. Software Release Planning Framework

Stakeholder Identification

- Collect / Estimate stakeholder’s salience
- Rank / Group stakeholder according to salience
- Select the representative stakeholders

![Graph showing stakeholder identification with bars for power, legitimacy, and urgency]
2. Software Release Planning Framework

Elicitation of Candidate Requirement Sets

Candidate Requirements \( R = \{ r_1, r_2, \ldots, r_n \} \)

- \( v_i \): subjective value assigned by stakeholder \( s_k \) to \( r_i \)
- \( W = \{ w_1, w_2, \ldots, w_m \} \) stakeholder weight (importance)

For \( r_i \in R \), its satisfaction \( s_j \) is:

\[
    s_j = \sum_{i=1}^{m} w_i \cdot v_{ij}
\]

Effort for developing \( r_j \) \( E = \{ e_1, e_2, \ldots, e_n \} \)

Cost limit (amount of resources): \( B \)

Optimization problem: Find \( U \), candidate requirement sets to be included in next release using Pareto dominance

\[
    \max \sum_{j \in U} s_j, \quad \min \sum_{j \in U} e_j, \quad \text{subject to} \quad \sum_{j \in U} e_j \leq B
\]
2. Software Release Planning Framework

Elicitation of Candidate Requirement Sets

Alternative formulation: Constraints on implementation order

- Implication, \((r_i \text{ implies } r_j)\)
- Combination interaction, \((r_i \text{ combined with } r_j)\)
- Exclusion interaction, \((r_i \text{ excludes } r_j)\)

Downsizing result set

Optimization algorithm: Obtain Pareto optimal solutions
2. Software Release Planning Framework

Next Release Agreement

Choose the set of requirements to be implemented

Quality indicators: Visual aids to guide decision makers

Let $U \subseteq \mathbb{R}$: solutions under analysis

Productivity: $\text{prod}(U) = \frac{\text{sat}(U)}{\text{eff}(U)}$

Coverage: $\text{sk} \in \text{Stk}$

$$\text{stcov}_i(U) = \frac{\sum_{j \in U} v_{ij}}{\sum_{j \in \mathbb{R}} v_{ij}}.$$

Diagram showing solutions and their properties.
Index

1. Introduction
2. Software Release Planning Framework
3. Case Study
4. Appraisal of the Framework
5. Conclusions and Future Work
3. Case Study

Dataset: Replacement Access, Library and ID Card project (RALIC)

Combine Access control systems at University College London (UCL)

RALIC Stakeholders Identification Recommendations Network → 144 stakeholders

138 requirements: 10 objectives, 48 requirements, 104 specific requirements

Effort: 4 to 7000 persons-hour

75 RALIC stakeholders use the 100-point method → 83 requirements left

Relevant Stakeholders Identification: Clustering → No interactions

12 stakeholders left

![Bar chart showing power, legitimacy, and urgency for various individuals]
3. Case Study

RALIC Elicitation of Candidate Requirement Sets

From Stk, R and E and Satisfaction $\rightarrow$ Optimisation Problem

\[
\begin{align*}
\text{max} & \sum_{j \in U} s_j, \\
\text{min} & \sum_{j \in U} e_j, \\
\text{subject to} & \quad B_1 \leq \sum_{j \in U} e_j \leq B_2
\end{align*}
\]

$B_1: 20\%, \ B_2: 25\%$

From total effort

Resource limit interval: No upper resource limit

Discard solutions with lower effort

For RALIC, $B_1=12473.3, \ B_2=13304.8$

Pareto front: Greedy Algorithm

For each effort value in the range $[B_1, B_2]$

Find solution with max number of requirements

Iterate replacing requirement with valid effort limit

Check dominance and replace if it’s higher

Simple and produces a Pareto front (not exhaustive)
3. Case Study

RALIC Elicitation of Candidate Requirement Sets

Pareto Front

Satisfaction

Effort (hours/person)
3. Case Study

RALIC Next Release Agreement

Visual Indicators Coverage

Best candidate solutions seems 13 or 14  
(Not for Productivity)

Human process: Pairwise comparisons (ex. 12 and 7)  
Other factors (non quantitative): Risk -> Better 7
3. Case Study

RALIC Next Release Agreement

Better to use quality indicators: summarise the information
Index

- 1. Introduction
- 2. Software Release Planning Framework
- 3. Case Study
- 4. Appraisal of the Framework
- 5. Conclusions and Future Work
4. Appraisal of the Framework

Previous strategies: three separate tasks
   Best requirements set: optimization problem
   Do not work in identification and prioritization

Stakeholders identification
   Manually (experience & intuition)
   Systematic (consistent, precise and complete result)
   Do not have a requirement selection stage

Solution Selection in Pareto front: Complex techniques

The three stages have not been in a unique framework
Index

- 1. Introduction
- 2. Software Release Planning Framework
- 3. Case Study
- 4. Appraisal of the Framework
- 5. Conclusion and Future Work
5. Conclusion and Future Work

Linking three complex software engineering problems
Global view of defining the next release goal

Framework provides a pragmatic approach

- Stages
  - Stakeholder Identification
  - Elicitation of Candidate Requirement Sets
  - Next Release Agreement

- Manage and improve tools/algorithms for each one
- Improve the whole process

Validity: Application to a real problem (RALIC)

Future work: Application to other software projects with data
Investigate the impact on the solutions in NRP
Thank you very much

abosch@ual.es