Decision Theoretic Requirements Prioritization

A Two-Step Approach for Sliding towards Value Realization

Nupul Kukreja
Center for Systems and Software Engineering (CSSE)
University of Southern California
Los Angeles, USA
nkukreja@usc.edu

***Abstract*—Budget and schedule constraints limit the number of requirements that can be worked on for a software system and is thus necessary to select the most valuable requirements for implementation. However, selecting from a large number of requirements is a decision problem that requires negotiating with multiple stakeholders and satisficing their value propositions. In this paper I present a two-step value-based requirements prioritization approach based on TOPSIS, a decision analysis framework that tightly integrates decision theory with the process of requirements prioritization. In this two-step approach the software system is initially decomposed into high-level Minimal Marketable Features (MMFs) which the business stakeholders prioritize against business goals. Each individual MMF is further decomposed into low-level requirements/features that are primarily prioritized by the technical stakeholders. The priorities of the low-level requirements are influenced by the MMFs they belong to. This approach has been integrated into Winbook, a social-networking influenced collaborative requirements management framework and deployed for use by 10 real-client project teams for the Software Engineering project course at the University of Southern California in Fall 2012. This model allowed the clients and project teams to effectively gauge the importance of each MMF and low-level requirement and perform various sensitivity analyses and take value-informed decisions when selecting requirements for implementation.**

***Index Terms*—Requirements, prioritization, decision theory, value-based prioritization.**

I. Introduction

There are usually more requirements than feasible in a given schedule. Simple techniques like 1-10 scoring or MoSCoW (must, should, could and want-to haves) aren't able to capture the true value of requirements. They incorrectly assume that the stakeholders are adept at correctly prioritizing the requirements by business value, and the process may need to be recursive in case of tied requirements. These techniques do not lend themselves well to various types of sensitivity analyses (e.g. ascertaining the impact of change of business goals on requirement priorities) or comparative analysis with existing requirements, in case of new or changed requirements. This makes it difficult to take value-informed decisions when selecting requirements for implementation, compelling one to rely on expert judgment alone.

In [1] Aurum and Wohlin advocated using decision making models for requirements engineering to have more rigor in the requirements prioritization process and selecting high-value requirements when implementing the software system.

In this paper I present a two-step prioritization approach based on a decision analysis framework – TOPSIS (Technique of Ordered Preference by Similarity to Ideal Solution) [2] for prioritizing system and software requirements. The software system to be developed is initially decomposed into high-level Minimal Marketable Features (MMFs) [3] which are further decomposed into low-level requirements. The business stakeholders prioritize the MMFs against business goals and the technical and business stakeholders collaborate to prioritize the requirements with respect to ease of realization (i.e. the extent to which the requirement is economically, socially, politically and technologically feasible). The priorities of the requirements are influenced by the MMFs, which in turn are influenced by the business goals that are themselves prioritized using success sliders [4] (elaborated below)**.**

This hierarchical prioritization approach has been integrated into Winbook [5], a social-networking influenced collaborative requirements management tool and deployed for use by 10 real-client student project teams at the University of Southern California. As a result, the teams were able to perform various types of sensitivity analyses on the requirement priorities and gain further insights into the overall value of the MMFs and individual requirement priorities. This led to better decision making in case of comparing new requirements to existing ones, as well as selecting the most risky (or valuable) requirements for prototyping and subsequent implementation.

II. Background And Related Work

I was involved in a related case-study [6] where we looked at selecting the most appropriate value-based requirements prioritization framework for company-wide use at a premier IT organization in India. We compared 17 prioritization frameworks against 17 distinct criteria as put forth by our stakeholders. TOPSIS was selected as the framework of choice and deployed for company-wide use and was extended to handle the need for hierarchical prioritization [6]**.**Other popular prioritization methods such as AHP [11] and QFD [12] were ranked considerably lower with respect to the criteria by the stakeholders; see [6]

Hierarchical prioritization was implemented using mathematical normalization i.e. prioritizing a 'child-level' requirement and scaling its priority by that of its parent (that

Figure 1 Example success sliders for a set of goals for an existing project (in Winbook)

are themselves prioritized using TOPSIS against the same or additional criteria). The sum of the scores of the children equals to that of their parent. This modification allowed for multiple levels of prioritization but proved to be cumbersome for use beyond three levels.

It must be noted that TOPSIS suffers from rank reversals i.e. if irrelevant MMFs/requirements are added to the prioritization the original prioritization order may change. The concern proved minimal since the teams were result oriented and not intent on gaming the system; but the concern could be significant for negotiation among competitors.

TOPSIS was deployed in the organization across various projects and proved valuable for providing a value-centric basis for prioritization [7]. Based on the positive feedback and usage experiences I decided to integrate TOPSIS into Winbook and used it for performing a two-step hierarchical prioritization of requirements as described below.

III. Approach and Uniqueness

Initially the success critical stakeholders brainstorm the value propositions the software system intends to satisfy. The value propositions (a.k.a. goals or objectives) are prioritized using simple success sliders as advocated in [4].
The stakeholders then prioritize the elicited goals using a simple heuristic – *no two goals must be at the same level,* thus forcing them to make the necessary tradeoffs. Figure 1 shows an example from a live project in Winbook.

MMFs are captured as a list in Winbook (not shown) and are prioritized against the set of goals as shown in Fig 2. The teams score each MMF on a 1-9 scale signifying its contribution in satisfying a particular goal (1: low to no satisfaction, 9: complete satisfaction). Teams can perform sensitivity analyses by changing the goal weights (i.e. vertical sliders in Fig 2) and visually see the impact on changing priorities via the expanding/contracting horizontal sliders displayed above each MMF. The horizontal sliders expand/contract in proportion to the TOPSIS-computed priority scores. The individual requirements/features pertaining to each MMF are captured in a user-story format [8] and are scored similarly against three dimensions of: business value, relative penalty (if not implemented) and ease of realization (scored on a Fibonacci scale using story-points [9, 10]).

IV. Results and Conclusions

Preliminary results have been more cost-effective than our previous use of MoSCoW and 1-10 scoring. Teams perform various types of sensitivity analyses to help ascertain the riskiest and most valuable requirements for prototyping and subsequent implementation. The business stakeholders were better able to gauge the impact of business goals on the MMFs and gain insight into the true priorities of things if they differed from intuition.

The two-step approach for prioritization has little overhead with a high-value output that helps the teams prioritize the requirements in a more rigorous and value-centric manner. The business stakeholders can independently prioritize the MMFs as per the business goals and have it impact individual requirements and can thus take informed decisions about which requirements to implement and channelize the negotiation sessions around the most valuable requirements.

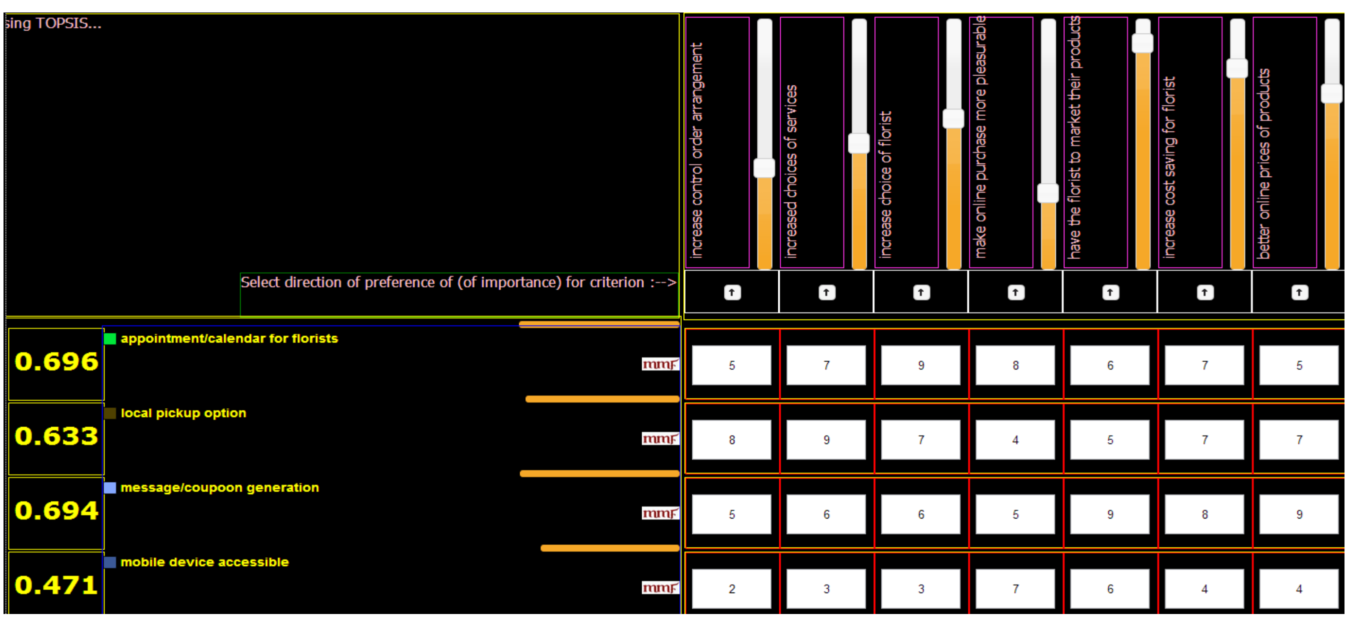

Figure 2 TOPSIS Prioritization of the MMFs. The priority scores are seen on extreme left and the goals along the top row, top-right.

REFERENCES

[1] A. Aurum and C. Wohlin, "Applying Decision-Making Models in Requirements Engineering," in Proceedings of Requirements Engineering for Software Quality, Germany, September 2002.

[2] R. Wang, “Performance evaluation method - Technique for order preference by similarity to ideal solution (TOPSIS)”, researcher.nsc.gov.tw/public/caroljoe/Data/02182133671.ppt.

[3] M. Denne and J. Cleland-Huang, "The Incremental Funding Method: Data Driven Software Development," IEEE Software, vol. 21, no. 3, 2004, pp. 39-47

[4] R. Thomsett, “Radical project management”, Upper Saddle River, NJ: Prentice Hall PTR, 2002. Print.

[5] N. Kukreja, B. Boehm, “Process Implications of Social Networking-Based Requirements Engineering Tools”, Proceedings of the International Conference on System and Software Proceses, June 2012.

[6] N. Kukreja, S. S. Payyavula, B. Boehm, S. Padmanabhuni, “Selecting an Appropriate Framework for Value-Based Requirements Prioritization: A Case Study”, Proceeding of the IEEE Requirements Engineering Conference, Sept. 2012.

[7] N. Kukreja, S. S. Payyavula, B. Boehm, S. Padmanabhuni, “Value-based requirements prioritization: usage experiences”, 2013 Conference on Systems Engineering Research. In press.

[8] K. Beck, “Extreme Programming Explained”, Addison Wesley, 1999.

[9] J. Grenning, “Planning Poker,” 2002, http://renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf

[10] M. Cohn, “Agile estimation and planning”, 1st ed. Prentice Hall, 2005.Print.

[11] T.L. Saaty, “The analytic hierarchy process”, McGraw-Hill, Inc. (1980).

[12] L. Cohen, “Quality function deployment: How to make QFD work for you”, Prentice Hall, 1995. Print.