Prioritizing Software Requirements in an Industrial Setting

Kevin Ryan
University of Limerick
Limerick
Ireland
+353-61-202776
Kevin.Ryan@ul.ie

Joachim Karlsson
Focal Point AB
Teknikringen 1E
583 30 Linköping Sweden
+46-13-213705
joachim@focalpoint.se

ABSTRACT
The planning of additional features and releases is a major concern for commercial software companies. We describe how, in collaboration with Ericsson Radio Systems, we developed and tested an industrially useful approach to software requirements prioritization.

Keywords
Requirements, prioritizing, release planning,

CONTEXT
Ericsson Radio Systems AB is a leading supplier of mobile telephony software. Since 1992 it has collaborated with Linköping University, Sweden, in a concerted effort to improve the early phases of the software engineering process. Starting in January 1994 we have participated in the requirements definition phase of a number of major projects and, through this collaborations, identified requirements prioritization as a key issue for software developers.

FOCUS
From the outset, it was clear that many major projects had far more candidate requirements than the company could ever afford to implement in a single release. A typical method of selecting which ones to implement was to give each requirement a rating from 1 (highest) to 3. Frequently the effect was that only those rated 1 were ever implemented and the others were postponed to possible later releases. It was clear that this approach was inadequate because it was too crude, too variable and did little to clarify the decisions being made. The group therefore decided to focus on developing better means of requirements prioritization.

A COST-VALUE APPROACH
The Analytic Hierarchy Process [1] is an established means of computing the relative weightings of a number of alternatives. It requires a pair-wise comparison of each alternative with every other alternative. Given a full set of comparisons AHP then computes a normalised set of weightings.

ICSE 97 Boston MA USA

Because of the redundancy inherent in so many separate comparisons, AHP can also indicate the consistency of the ratings.

We decided to adopt the AHP as the basis for a systematic method of prioritizing requirements. Our first step was to train the developers in this process. They then used it to estimate both the relative cost of implementing each candidate requirement and the relative value of this requirement to the eventual customer. The resulting distributions gave immediate insight since they showed that the relative cost or value of candidate requirements could vary by orders of magnitude [2].

The second step was to make the design tradeoffs more explicit. We combined the cost and value estimates for a set of requirements and displayed them on a cost-value diagram. This made it graphically clear that, in the project in question, a very high proportion of the total potential value could be achieved by choosing an appropriate subset of requirements for implementation. For example, in one project, choosing 11 of the 14 requirements could deliver approximately 94% of the possible maximum value for about 78% of the possible maximum cost. An example cost-value diagram is given in Figure 1.

This approach was found very useful by all the team members but they pointed out a number of shortcomings. The process was tedious, and was only realistic for small (<20) sets of requirements. Neither did it take account of the interdependencies between requirements that frequently occur in real projects.

TOOL SUPPORT
To address these difficulties we specified, and had developed, a prototype support tool which aided the developer, or other stakeholder, in following the cost-value approach. The tool also incorporates a number of improvements to make the approach more applicable in industry. Firstly, it uses local and global stopping rules to eliminate unnecessary pair-wise comparisons. Secondly, it models the interdependecies between requirements so that, for example, a pair of mutually exclusive candidate requirements can not both be

---

Joachim Karlsson is also at the Department of Computer and Information Science, Linköping University, S-58183 Sweden

chosen in the set for implementation. This tool was then deployed in an on-going project and was found to be both effective and efficient. In using the prototype, developers, and other stakeholders, explored the range of possible alternatives and could decide which features were to be included in each successive release. What was remarkable was that, previously, these choices were always made implicitly, without any quantitative or visual support.

CURRENT STATUS

At present the prototype tool is installed at Ericsson Radio and is being used on a trial basis. If it proves to be generally acceptable it is expected that it will be adopted for corporate use.

LESSONS LEARNED

There is very little guidance available for those making key decisions about what features to include in future systems. A simple, effective prioritization process was found to be very beneficial. A graphical display of costs versus values can act as a focal point for the process. Industrial adoption requires adequate tool support.

ACKNOWLEDGMENTS

This work was supported by Ericsson Radio System AB and the Swedish National Board for Industrial and Technical Development Project No. 9303280-2. The prototype tool was developed by Stefan Olsson.

REFERENCES

1. Saaty T L. The Analytic Hierarchy Process. McGraw Hill Inc., 1980.
2. Karlsson J, Software Requirements Prioritizing in *Proc. 2nd International Conference on Requirements Engineering*, [Colorado, April 1996], IEEE CS Press, 110-116.

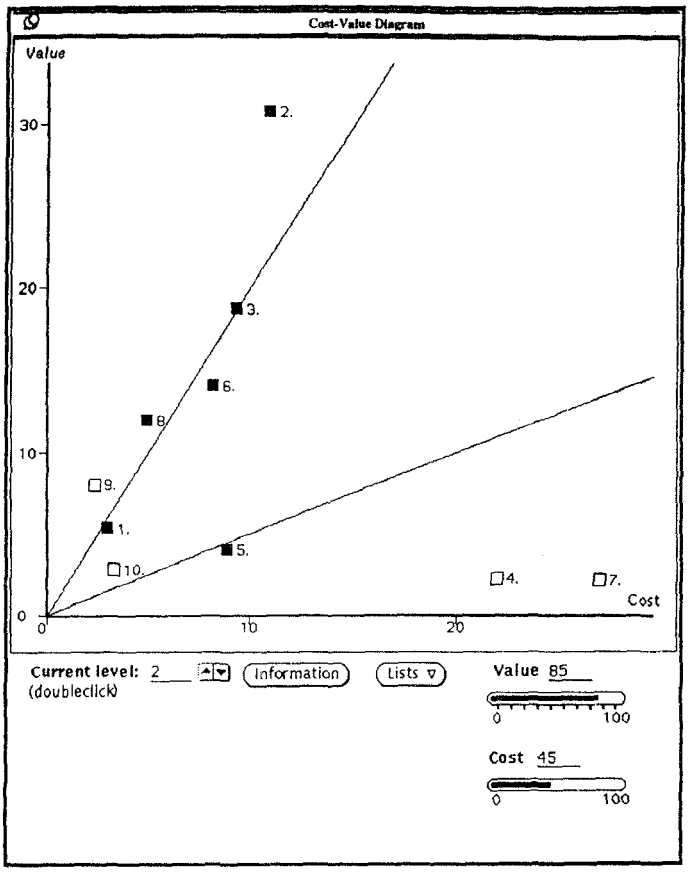

Figure 1. An example of a cost-value diagram.