# Controlling Lost Opportunity Costs in Agile Development – the Basic Lost Opportunity Estimation Model for Requirements Scoping

Krzysztof Wnuk[1], David Callele [2], Even-Andre Karlsson[3], Björn Regnell[1]

[1] Department of Computer Science, Lund University, Sweden,
{Krzysztof.Wnuk,Bjorn.Regnell}@cs.lth.se
[2] Department of Computer Science University of Saskatchewan,
Saskatchewan, Canada,
callele@cs.usask.ca
[3] AddALot, Lund, Sweden
even-andre.karlsson@addalot.se

**Abstract.** In this paper, we present a model for estimating the final decision point for committing to the development of features that are under analysis for inclusion in the scope of a future release. The Basic Lost Opportunity Estimation Model (BLOEM) is based on studies at a company that uses an agile-inspired software development model. The main objective of BLOEM is to support feature selection in a context where the business value estimates change as the requirements analysis progresses and can be represented as a function of time. With BLOEM, a set of possible management strategies can be assessed for individual features in order to determine a final decision point when either an implementation commitment decision or a rejection decision has to be made. Our initial validation, conducted on a set of 166 features, suggests that the model can be applied in a real-world context to control lost opportunity costs due to feature cancellation and BLOEM can therefore provide valuable input to the selection process. Limitations of BLOEM are discussed and issues for further research are presented.

**Keywords:** Requirements management, scope management, agile development, software business

## 1 Introduction

Market-driven software development attempts to deliver the right product at the right time to the target market [1]; time-to-market and release scheduling may strongly affect market success [2]. Threats include introducing new requirements in response to competitive pressures thereby creating a risk of feature creep negatively impacting timely (reliable) market introduction. Prior work [3] identified a pattern where features were pruned from a release only after significant (wasted) investment. These wasted efforts may result in pressures that reduce the effectiveness of requirements engineering and management activities, activities considered key components of business success for software companies [1].

Numerous prioritization techniques have been investigated and utilized to identify and select the most valuable features for the next release of a project. However, most techniques rely on accurate market value and effort estimates that appeared to be difficult to generate early in the development process. The agile software development movement [4] attempts to increase the flexibility of requirements processes and to improve process response to unexpected changes in scope – using one-dimensional (relative) methods for cost estimation and as a substitute for real market values [5]. These challenges influenced the case company in this work to adopt a similar one-dimensional prioritization methodology. Unfortunately, this methodology does not address unexpected market forces nor does it target the release date as a critical success factor for software product delivery in a market-driven context [1].

We present here the Basic Lost Opportunity Estimation Model (BLOEM), a simplified version of the previously published LOEM model [6] for controlling software project lost opportunity costs. BLOEM targets processes that use a one-dimensional requirements prioritization technique such as agile software development [5, 7]. The simplified model can use relative or absolute values for candidate features as well as their planned release date for estimating final decision points for inclusion or rejection within a release. Pragmatically, BLOEM attempts to control wasted effort by identifying candidate features for cancellation at an earlier stage.

The paper is structured as follows: Section 2 introduces both related work and the motivation for the work. Section 3 presents the model and Section 4 presents the initial validation of the model using empirical data from a large company, including discussion, while Section 5 concludes the paper and lists further work.

## 2 Background and Motivation

Agile software development focuses on continuously delivering business and customer value to increase the probability of early return on Investment (ROI) [5]. Cao *et al.* note that agile requirement engineering practitioners uniformly reported that their prioritization is principally based on business value [8]. While prioritizing based on business value is considered a key requirements prioritization criterion, it doesn't decrease the temporal uncertainty as a consequence of rapidly changing markets. The BLOEM model presented here helps to control this uncertainty by reducing costs associated with canceled features, facilitating constructive use of resources by controlling lost opportunity costs.

To illustrate the motivation behind BLOEM we present the results of our analysis of an agile prioritization process applied to a set of features for product it developed at a large multinational company. The company has introduced a new development process inspired by agile development processes where the old stage-gate model was replaced by a continuous development model. The requirements legacy database contains over 20,000 requirements, represented at various abstraction levels, and the total number of features under consideration exceeds 10,000. Figure 1 depicts the normalized value of the (widely varying) business priority for each of a set of features plotted against the total time that the features were in the software development process (including the requirements phase). Of 166 candidate features that were considered for this release, 83 were withdrawn. As can be seen in Figure 1, withdrawn features had both high and low priorities (value) and were withdrawn at all times throughout the release cycle.

BLOEM's intent is to quantify the effort spent on the features in the triangle labeled “area of interest” in Figure 1 to support efficient process management. Ultimately, we want to decide to make the feature keep/cancel decision as quickly as possible – if a feature is withdrawn, it is best to withdraw it as early as possible to minimize wasted effort.

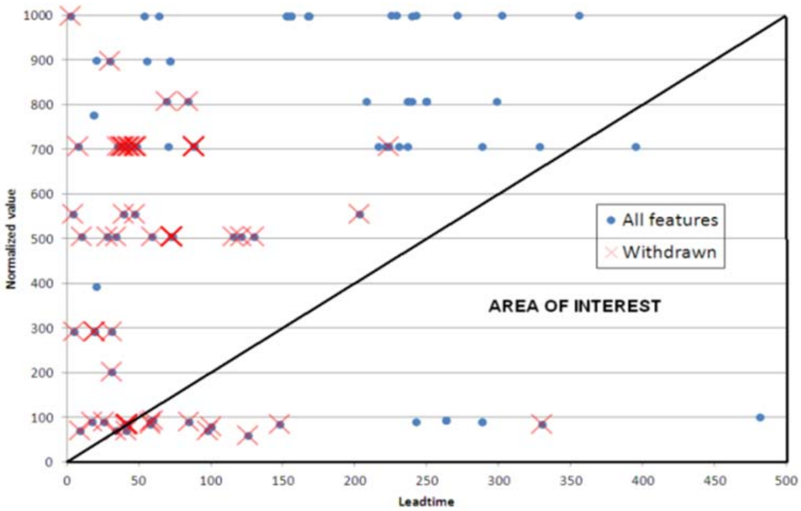


**Fig. 1** Dots represent implemented features while crossed dots represent withdrawn features.

## 3 The Basic Lost Opportunity Estimation Model

In this section, we define BLOEM and describe how it can be used for analyzing the efficiency of the requirements process and temporal parameters for estimating lost opportunity costs. The main assumption of the model is that the decision-making criteria are temporal functions, not fixed values, which facilitate their use even in dynamic situations with uncertain scoping decisions. The model is expressed in such a way that different management techniques, such as introducing an excess number of features into the process and allowing the requirements process to find the strongest features, can be addressed. The model is based on the market-driven requirements engineering premise that the value of a requirement is a temporal function that is sensitive to market forces and opportunities – often a feature will only have market value for a limited time. Even features that offer unique capabilities see a significant reduction in their market value when competitors catch up and offer the feature in their own products.

*Total value* ***V(t)*** for a feature is defined in equation (1) where t=a is feature inception (when the feature begins to have non-zero value) and t=b is when the feature ceases to have any market value. A feature is cancelled at t=c. Maintenance, as both a cost and as a revenue source, is not considered in this simplified model.

$$\int_a^b V(t)dt \quad \textbf{(1)} \qquad \int_a^c V(t)dt \leq \delta \quad \textbf{(2)}$$

The value function will depend on the characteristics of the target market.

From a management perspective we assume that features under investigation should be kept within the project scope until a defined value threshold ($\boldsymbol{\delta}$), known as the Final Decision Point (FDP), is reached. The threshold value can be unique to each feature. High-value features (*e.g.* priority in the top 25%) could have the threshold set higher than less valuable features (*e.g.* priority in the bottom 25%). Final decisions as to whether to keep (and realize the investment) or cancel (and minimize losses) are delayed for the most valuable features while the least valuable features are canceled relatively early. The Final Decision Point can be used as to enforce a budget-like approach to the scoping management process.

We consider all canceled features to be wasted effort. However, investments in features that are canceled before the threshold are considered *controlled waste*: there is waste but it is under management control and the risk of inter-feature dependencies is held to an acceptable level. Features that are canceled after the final decision point are *uncontrolled waste* – something unexpected has happened and time or resource constraints cannot be met for this release cycle.

The flexibility required in the development process can be adjusted by changing the value of $\boldsymbol{\delta}$**.** The *overall impact* of a set ***K*** of withdrawn or cancelled features within a development cycle is calculated using equation 3:

$$\sum_{k=1}^{K} \frac{\int_a^b V_k(t)dt - \delta_k}{K} \quad \textbf{(3)}$$

## 4. Initial model evaluation

BLOEM was initially validated using a set of 166 features analyzed by the case company (depicted as dots in Figure 1). The feature status in the data set ranged from the definition phase, through implementation, to completion. During this period 87 features were canceled (dots with X's in Figure 1, several Xs overlap). The value function is defined relative to the lifespan for the feature, the period from feature inception until the feature ceases to have any market value. Two value functions were considered to observe their effects upon the results. The first function assumes a constant value across the lifespan of the feature. The second function assumes that value is normally distributed with the mean positioned at 50% of the lifespan and the standard deviation set to 1/6 of the lifespan.

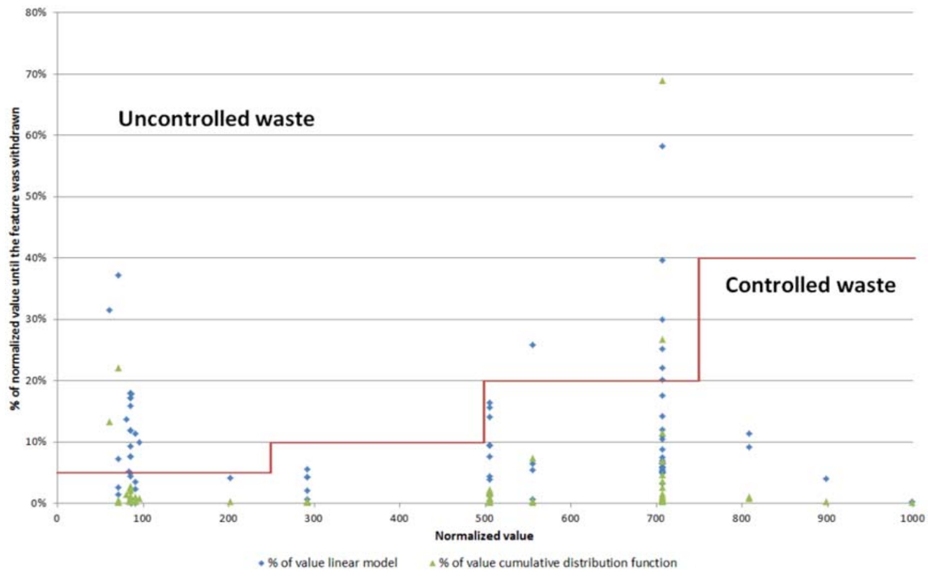


**Fig. 2** Results from model validation efforts.

**Results.** Figure 2 depicts the results of using BLOEM with these two value functions (several data points overlap). The constant value function is represented by dots and the normal value function is represented by triangles. Because the value function is defined, in this case, to cover the entire product lifecycle, the final decision points should be only a small portion of the lifespan of the feature. The final decision points, represented by the red line in Figure 2, are set to 5% of the value function for low priority features (below 250), 10% for low-medium priority features (between 251 and 500), 20% for medium-high priority features (between 501 and 750) and 40% for high priority features (over 750). These final decision point values represent, in effect, the budget for feature scoping activities at each priority level.

**Discussion.** Under the assumptions of the constant value function, the average *uncontrolled waste* was 10.2% of the normalized value for the 25 features that were withdrawn after their final decision point. These features remained in the process for a cumulative 1021 days after the FDP and the resources expended upon these features during this period represent both direct costs and lost opportunity costs. Under the assumptions of the normal value function, the average *uncontrolled waste* was 20.3% of the normalized value for the 4 features that were withdrawn after their final decision point. These features remained in the process for a cumulative 75 days after the FDP and the resources expended upon these features during this period represent both direct costs and lost opportunity costs. Under the normal value function, each feature has a very low value at the beginning of the lifespan. However, the value follows the cumulative distribution function therefore those features that exceed the FDP have much higher associated value on a per-feature basis that is realized as a loss (wasted effort) when the features are cancelled.

In both cases, the overall impact of the entire feature set (kept and cancelled) was under the budget line (linear: -5.1%, normal: -37.6%) indicating that there was capacity to investigate more features within the given budget. Alternatively, the budget could have been tightened (final decision points set earlier) or fewer resources could have been allocated to the release as a whole (more features per human resource).

## 5. Conclusions and Further Work

The Basic Lost Opportunity Estimation Model (BLOEM) for controlling lost opportunity costs related to cancelled and withdrawn features was presented. Targeted at processes that employ one-dimensional requirements prioritization [5] (such as agile methodologies), its performance was investigated with an initial data set. The analysis clearly identified opportunities to improve process efficiency within the examined data set. The costs associated with delayed feature cancellation were quantified and a budget-driven final decision point mechanism was presented as well as management guidance for interpreting the results.

Preliminary discussions of the initial validation results were held with two practitioners at the case company. They responded positively to the model concept and its potential for controlling lost opportunity costs. The investigation showed that the results are sensitive to the cost function and further investigation into other cost functions is suggested to determine their utility for management decision support. For example, agile methodologies suggest a constant feature priority evaluation process – how well does BLOEM perform in such an environment? Further validation with other data sets is needed.

## References


1. Regnell, B., Brinkkemper, S.: Market-Driven Requirements Engineering for Software Products. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 287-308, Springer, 2005.
2. Chen, J., Reilly, R. R., Lynn, G. S.: The impacts of speed-to-market on new product success: the moderating effect of uncertainty. Trans. Soft. Eng. 52, 199-212 (2005)
3. Wnuk, K., Regnell B., Karlsson, L.: What Happened to Out Features? Visualization and Understanding of Scope Change Dynamics in a Large-Scale Industrial Setting. In: 17th IEEE Int. Requirements Engineering Conference, pp. 41-50. IEEE Press, New York (2009)
4. The Agile manifesto, http://agilemanifesto.org/ accessed January 2012
5. Racheva, Z., Daneva, M., Sikkel, K., Herrmann, A., Wieringa, R.: Do We Know Enough about Requirements Prioritization in Agile Projects: Insights from a Case Study. In: 18th IEEE International Requirements Engineering Conference, pp. 147-156. IEEE Computer Society, Washington, DC, (2010)
6. Wnuk, K., Callele, D., Regnell, B.: Guiding requirements scoping using ROI: towards agility, openness and waste reduction. In: 18th IEEE Int. Requirements Engineering Conference, pp. 409-410. IEEE Press, New York (2010)
7. Beck, K.: eXtreme Programming Explained: Embrace Change. Addison Wesley, 2000.
8. Cao, L., Ramesh, B.: Agile Requirements Engineering Practices: An Empirical Study. IEEE Software. 25, 60-67 (2007)