Optimizing the Incremental Delivery of Software Features under Uncertainty

Olawole Oni, Emmanuel Letier

Department of Computer Science, University College London, United Kingdom.
`{olawole.oni.14, e.letier}@ucl.ac.uk`

Abstract. [Context] Lean and agile software development processes encourage delivering software in small increments so as to generate early business value, be able to adapt to changes, and reduce risks. Deciding what to build in each iteration is an important requirements engineering activity. The Incremental Funding Method (IFM) partly supports such decisions by identifying sequences of features delivery that optimize Net Present Value (NPV). **[Problem]** The IFM, however, does not deal explicitly with uncertainty and considers the maximization of NPV as the only objective, without explicit consideration for other objectives such as minimizing upfront investment costs and maximizing learning so as to reduce uncertainty and risk for future iterations. **[Ideas]** This short paper presents our ongoing research to address these limitations by extending IFM with Bayesian decision analysis to reason about uncertainty and with Pareto-based optimization to support decisions with respect multiple conflicting objectives. **[Contributions]** The paper presents the current version of our tool-supported extension of the IFM, illustrate it on a small example, and outlines our research agenda.

Keywords: Software Engineering Decision Analysis, Requirements Engineering, Agile Software Development

1 INTRODUCTION

Delivering software in small increments is widely regarded as an appropriate approach to deal with requirements uncertainty, manage software development risks, and generate early business value [1]. An important requirements engineering activity in this context is to decide the sequence in which software features will be developed and delivered [2, 3]. The Incremental Funding Method (IFM) is a financially informed approach to support such decisions by analyzing the cash flows and Net Present Value (NPV) of alternative feature delivery sequences [4, 5]. These financial concerns are critical to requirements engineering decisions; they can turn a project that is not financially viable into one that becomes viable through an appropriate sequencing of feature delivery that brings in early value and funds to the project.

The IFM, however, has limitations. A first limitation is that while clients and software developers have inevitable uncertainty about the value and cost of individual features, the IFM does not represent and analyze such uncertainty explicitly. Extending the method to reason about such uncertainty would help requirements engineers analyze the uncertainty and risks associated with alternative delivery sequences. A second limitation of IFM is that while clients and software developers generally have multiple conflicting goals, the IFM optimization algorithm considers the maximization of NPV as the sole objective. Extending the IFM to deal with multiple objectives would allow requirements engineers to systematically explore tradeoffs between, for example, maximizing NPV, minimizing upfront investment cost, and other non-financial goals.

This paper presents our initial work to address these limitations. Our approach consists in extending IFM with Bayesian decision analysis to reason about uncertainty [6, 7] and with Pareto-based optimization to support decisions with respect multiple conflicting objectives [8]. We give a brief overview of the current version of our tool-supported extension of the IFM, illustrate it on a small example, and present an agenda for future research.

2 BACKGROUND

The IFM considers software systems to be composed of Minimum Marketable Features (MMF) and Architectural Elements (AE). A MMF is a small self-contained unit of functionality that provides value to the client. An AE is an element that does not provide client value in itself but is a prerequisite to the delivery of other AEs and MMFs. MMFs and AEs are collectively referred to as elements. An element X depends on an element Y means that X cannot be delivered before Y, because of constraints in the development process or application domain. To illustrate the IFM and our extension, we use the hypothetical example of the development of a web banking application first introduced in the IFM book [4]. Figure 1 shows the MMFs, AEs, and dependency relations for this application.

Once a system has been broken down into MMFs and AEs, we must analyze the projected cost and revenue of each element over a number of business periods. In our web banking application, the analysis will be over 4 years split into 16 trimesters. Projected costs and revenues are typically elicited from software architects, clients and marketing. The result of such analysis is recorded in a cash flow projection table, such as Table 1, that shows for each MMF and AE, one or more periods of initial investment during which the cash flow is negative followed by periods of revenues during which the cash flow is positive or zero. For example, in Table 1, AE 1 takes one period to deliver at a cost of \$200,000, and MMF B takes two periods to deliver, each period requiring an investment of \$200,000, followed by periods of increasing revenue starting at \$90,000 and rising to \$225,000 6 periods after delivery.

Once the cash flow projections are known, the IFM automatically analyzes possible delivery sequences and suggests a delivery sequence that maximizes NPV —a standard financial metric measuring the difference between revenues and costs (i.e. positive and negative cash flows) taking into account the time value of money at a fixed discount rate. In our example, we use a discount rate of 1% per period.

When a system is composed of only a few AEs and MMFs, it is possible to compute the NPV of all possible delivery sequences and identify one that maximizes

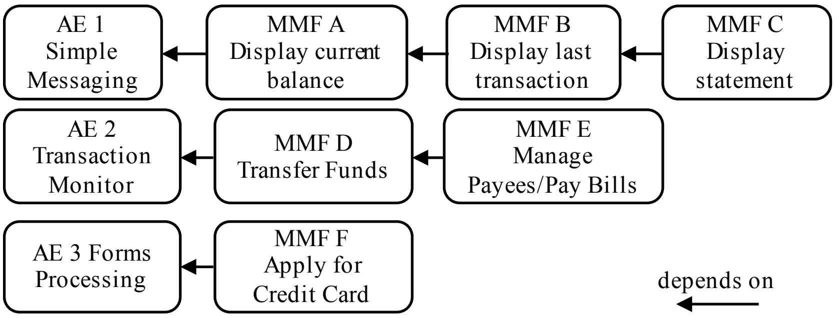
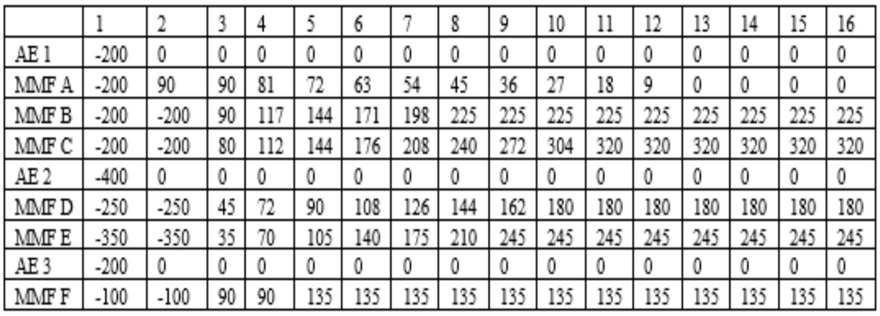


Fig. 1. IFM precedence graph for a hypothetical web banking system [1]



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
AE 1	-200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MMF A	-200	90	90	81	72	63	54	45	36	27	18	9	0	0	0	0
MMF B	-200	-200	90	117	144	171	198	225	225	225	225	225	225	225	225	225
MMF C	-200	-200	80	112	144	176	208	240	272	304	320	320	320	320	320	320
AE 2	-400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MMF D	-250	-250	45	72	90	108	126	144	162	180	180	180	180	180	180	180
MMF E	-350	-350	35	70	105	140	175	210	245	245	245	245	245	245	245	245
AE 3	-200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MMF F	-100	-100	90	90	135	135	135	135	135	135	135	135	135	135	135	135

Table 1 Cash flow projections for the web banking MMFs and AEs [1]

NPV. When such an exhaustive analysis is not possible, IFM uses a heuristic to find a near optimal solution. In our running example, assuming a single MMF or AE can be worked on during each period, IFM computes that the optimal delivery sequence consists in developing AE 1 first, followed by MMFs A, B, and C.

In practice, the suggested optimal or near optimal delivery sequence provides a baseline that decision makers can adapt to take into consideration additional objectives and constraints not represented in the model. The IFM analysis is used to decide what to build in the first period and has to be repeated at the beginning of each new period, possibly with an updated list of MMFs, AEs and revised cash flow projections taking into account business changes and an increased understanding of the business needs and development technologies.

3 RELATED WORK

Other software engineering decision methods take, like the IFM, a financial perspective to inform funding and design decisions [7, 9, 10]. These methods, however, support one-time upfront decisions only without considering how to deliver the system in small increments and optimize the delivery sequence. The IFM is also related to methods supporting release planning by reasoning about the priorities assigned by different group of stakeholders to different requirements [3]. Some release planning methods deal with uncertainty related to development effort [11, 12]. These methods aim to identify release plans that minimize cost and maximize value, where value is defined as a weighted sum of stakeholder's preferences rather than in financial terms.

Several extensions to the IFM have already been proposed: (i) to improve the IFM optimization algorithm [13], (ii) to extend IFM with uncertain cash flows and generate flexible investment policies in the form of decision trees [14], and (iii) to take into account the behavior of competitors using game theory [15]. In our approach, we model and analyze cash flow using probability distribution functions similarly to previous work [14] but differ from previous work by considering multiple optimization objectives and by aspiring to introduce concepts from Decision Analysis, such as the expected value of information [7], to guide decisions about which uncertainty to reduce in order to reduce risks and increase NPV.

4 MULTI-OBJECTIVE IFM UNDER UNCERTAINTY

In order to test the feasibility of extending IFM to deal with uncertainty and multiple objectives, we have developed a prototype tool (in R) and have applied it to a couple of small examples. Our tool has the following capabilities:

1. Uncertainty about MMF and AE cash flows are represented as triangular distributions. A triangular distribution is characterized by three parameters specifying the lowest, most likely, and highest value for a variable. We have chosen this distribution because it is easily understood and used in IT portfolio management tools [10]. We envision, however, extending our tool to additional probability distributions [16].
2. Our tool uses Monte-Carlo simulation to compute the impact of MMF and AE cash flow uncertainty on the NPV of alternative delivery sequences. For each delivery sequence, our tool then computes a series of statistics including its expected NPV (the mean NPV over all simulations), expected investment cost (the mean of the total cost to be invested in the project before it has a positive cash flow), and its investment risk (the ratio between its NPV standard deviation and its expected NPV [10]).
3. The statistics about the NPV simulations are then used to select the Pareto-optimal set of delivery sequences that maximize expected NPV, minimize expected investment cost, and minimize investment risks. We have chosen these objectives because they are used in IT project portfolio management tools [10]. Decision makers can, however, select alternative set of optimization objectives that suits their context.

Returning to our running example, we have extended the cash flow table of Table 1 with uncertainty by assuming that cost items were underestimated with an uncertain cost overruns having a triangular distributions with parameters (0, 0.2*c*, 0.45*c*) where *c* is the initial cost prediction, and by assuming that revenue items tended to be overestimated and have a triangular distributions with parameters (0, *r*, 1.2 *r*) where *r* is the initial revenue prediction. Given these uncertainties, Fig. 2 shows a plot of the expected NPV and investment cost for all possible delivery sequences with the sequences identified as Pareto-optimal marked with a cross. Our tool also allows visualizing the cash flow uncertainty of any delivery sequence, as shown in Figure 3 for one

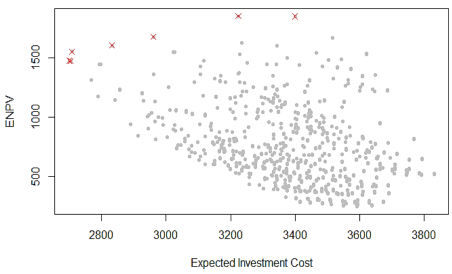


Fig. 2. Expected NPV and investment cost for the web banking application. Pareto-optimal sequences are marked with a cross.

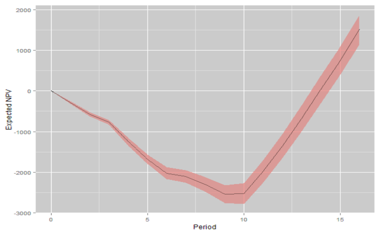


Fig. 3. Uncertain cash flow for one of the Pareto-optimal delivery sequence

of the Pareto-optimal delivery sequences. The main black line represents the mean cash flow and the shaded area its standard deviation.

Our current implementation has a couple of limitations: (i) like the standard IFM algorithm, it assumes a single MMF or AE can be worked on during each period, and (ii) it uses an exhaustive search to identify Pareto-optimal delivery sequences which limits its scalability to problems involving no more than a dozen MMF and AEs. We intend to address these limitations by removing the assumption from our model and by using search-based evolutionary algorithms instead to improve scalability.

5 FUTURE WORK

Our end goal is to develop a sound and practical method to reason about uncertainty and take into account multiple goals during incremental software development projects. This paper presents our first steps towards that goal. Future work needed to achieve our goal include: (i) facilitating the elicitation of accurate cash flow uncertainty, notably by relying on methods used in other domains [16]; (ii) enriching the IFM decision model by integrating it with other requirements and architecture models, for example with quantitative goal models [17] and software value maps [18]; (iii) enriching the IFM so as to take into account learning objectives aimed at reducing uncertainty about the cost and value of future development activities, for example through using information value analysis [7]; (iv) helping decision makers interpret and act on the method's output, i.e. the Pareto-optimal solutions and uncertain cash flows, notably through clustering of Pareto-optimal delivery sequences [19]; and (v) evaluating the method scientifically through simulations and real case studies.

6 CONCLUSION

Requirements engineering decisions are inherently multi-objective and confronted with uncertainty. Developing and delivering software features in small increments helps managing uncertainty but raises the question of what to develop in each iteration. Today, such decisions are largely guided by intuition. We believe that a more scientific and evidence-based decision method could lead to better decisions and result in significantly reducing the cost and increasing the value of software projects. We have outlined our initial work and roadmap to develop such a method by extending the IFM with Bayesian decision analysis and Pareto-based optimization methods.

Acknowledgment. Olawole Oni's research is funded by the Tertiary Education Trust Fund (TETFund) in Nigeria.

7 REFERENCES

1. G. Larman and V. R. Basili. "Iterative and incremental development: A brief history", *IEEE Computer* 6 (2003): 47-56.
2. D. Leffingwell. *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. Addison-Wesley Professional, 2010.
3. D. Greer and G. Ruhe, "Software release planning: An evolutionary and iterative approach," *Inf. Softw. Technol.*, vol. 46, no. 4, pp. 243–253, 2004.
4. M. Denne and J. Cleland-Huang, *Software by Numbers*, Sun Microsystems Press, 2004.
5. M. Denne and J. Cleland-Huang, "The incremental funding method: Data-driven software development," *IEEE Softw.*, vol. 21, no. 3, pp. 39–47, 2004.
6. R. L. Winkler. *An introduction to Bayesian inference and decision*. 2nd Edition., Probabilistic Publishing, 2003.
7. E. Letier, D. Stefan, and E. T. Barr, "Uncertainty, Risk, and Information Value in Software Requirements and Architecture," *Proc. 36th Int. Conf. Softw. Eng.*, 2014.
8. M. Harman and B. F. Jones. "Search-based software engineering." *Information and Software Technology* 43.14 (2001): 833-839.
9. M. Moore, R. Kaman, M. Klein, and J. Asundi, "Quantifying the value of architecture design decisions: lessons from the field," *Proc. 25th Int. Conf. Softw. Eng.,* 2003
10. M. Cantor, "Calculating and improving ROI in software and system programs," *Commun. ACM*, vol. 54, no. 9, 2011.G. Ruhe, Günther and D. Greer. "Quantitative studies in software release planning under risk and resource constraints." Proc. *ISESE 2003.*
11. K. Logue and K. McDaid. "Agile release planning: Dealing with uncertainty in development time and business value." *Proc. ECBS 2008.*
12. A. J. Alencar, et al., "A statistical approach for the maximization of the financial benefits yielded by a large set of MMFs and AEs," *Comput. Informatics*, vol. 32, no. 6, 2013.
13. B. P. Barbosa, E. A. Schmitz, and A. J. Alencar, "Generating software-project investment policies in an uncertain environment," *Proc. SIEDS 2008*.
14. E. M. Da Cunha Mattos, et al., "Applying Game Theory to the Incremental Funding Method in Software Projects," *J. Softw.*, vol. 9, no. 6, , 2014.
15. A. O'Hagan, C. Buck, A. Daneshkhah, J. Eiser, P. Garthwaite, D. Jenkinson, J. Oakley, and T. Rakow, *Uncertain Judgments: Eliciting Experts' Probabilities*. Wiley, 2006.
16. W. Heaven and E. Letier. "Simulating and optimising design decisions in quantitative goal models." *Proc. 19th Intl. Conf.* on *Requirements, 2011.*
17. K. Mahvish, T. Gorschek, and M. Wilson. "The software value map—an exhaustive collection of value aspects for the development of software intensive products." *Journal of Software: Evolution and Process*25.7 (2013): 711-741.
18. V. Veerappa and E. Letier. "Understanding clusters of optimal solutions in multi-objective decision problems." *Proc. 19th Intl. Conf.* on *Requirements, 2011.*