Chapter 1
Business Value Disadvantaged

What the customer buys and considers value is never a product. It is always utility, that is, what a product or a service does for the customer.

PETER DRUCKER

Abstract Despite the current emphasis on benefit in stakeholders' minds, there is still a focus on cost management when it comes down to the day-to-day work in modern software development. This works counter to underlying assumptions in modern development methodology. We motivate a more deliberate approach to benefits management during development, but it is the *combination* of cost and benefits management that saves the day.

1.1 A Paradoxical Emphasis on Cost

Modern development ideals focus on business value. In agile management and development, the mantra is 'Value for the customer'. The product owner is involved along the way and backlogs are prioritized, with the best intent to produce benefit. Yet, it seems that, in many information technology development projects, there is still bewilderment regarding how exactly customer value should be expressed in process decisions.

Routines for cost estimation are common, and cost estimates and productivity outlooks are routinely updated and monitored. Earned value measures and burndown charts can tell you when to start cutting back the scope. However, chances are that benefit is treated haphazardly compared to cost [5], which is a paradox, given the focus that business value is supposed to have.

This book promotes the idea that one should treat benefit with at least the same systematic attention as one treats cost. Moreover, benefit and cost estimates should be combined to give estimates of benefit over cost in a manner that enables *benefit/cost-driven development*.

The absence of an explicit treatment of benefit can lead to decisions based only on cost when one actually wishes to make decisions based on business value. It would be good if one could use a burndown chart to cut or promote functionality on the grounds of benefit rather than cost only.

J. E. Hannay, *Benefit/Cost-Driven Agile Software Development*,
Simula SpringerBriefs on Computing 8,
https://doi.org/10.1007/978-3-030-74218-8_1

Further, one is in danger of perceiving an expensive piece of functionality as also representing a lot of benefit. This fallacy piggybacks on the folkloristic *common law of business balance* [13], that is, the principle that one cannot pay a little and receive a lot: one should have to pay more for more of a product (ten bottles of wine) than for less of that product (two bottles of the same wine). The principle applies to software development as well; it is reasonable to expect to pay more for more software than for less software.

However, more software does not necessarily provide functionality that delivers more benefit. The confounding of cost with benefit transforms the reasonable principle above into a fallacy.

Clearly, then, there is another dimension to take heed of, in addition to the *amount of software* or even the *amount of functionality*. Thus, unless one has a sensible measure of benefit for one's backlog, one will not be able to manage construction with respect to benefit and will potentially regress to merely producing amounts of software instead.

1.2 Taking Control ...

Management based explicitly on benefit, in addition to cost, implies steering development activities toward the intended goal. It should also help you to avoid phenomena such as the *escalation of commitment to a failing course of action* in general [15, 16], and in software development in particular [7, 8]. This phenomenon involves people continuing to pursue activities in the face of clear signs that the activities are not achieving the goals, due to an (often emotional) attachment to the effort already invested in the activities. Related to this are the *sunk cost effect* and the *Concorde effect*, wherein a rationale is created to continue an ostensibly failing course of action, with the argument of not wasting what has already been invested [1].

A business strategy with plans that express development metrics explicitly in terms of business value and cost is a valuable tool to counter such effects. Suppose the strategy states that development will cease as soon as potential business value is no longer produced. Suppose, also, that metrics are in place that keep track of not only the amount of software produced (and money spent), but also the amount of benefit that the software is expected to give. Then, it should be impossible to enter into the realm of wilfully producing waste without at least someone in the steering group noticing.

1.3 ... with Agile

Agile promotes the frequent deployment of functionality. When agility is combined with wise architectural design in the form of parts of functionality that provide integral benefit (product elements), stopping development should be much less scary.

One should cease development when the benefit to cost can no longer be defended. If backlogs are ordered so that elements with high benefit over cost are realized first, then, by design, what has been produced and deployed until then already holds benefit. It is not so that everything goes to waste by stopping, so there is much less vulnerability to the sunk cost effect. In fact, what is then happening is not the *premature* cessation of development, but the *cessation of development just in time*.

Case 1. In 2013, a public sector welfare administration terminated its information technology modernization programme prematurely after about one and a half years' development. The total budget was about EUR 400 million at the time of writing, to be spent over six years. The sunk cost at termination was about EUR 180 million, of which EUR 36 million was spent on functionality that was never to be used [10, 11]. Generally presented in the press as yet another information technology scandal, the termination of the programme before all was lost was also applauded as a remarkably mature decision [19]. When things began to downhill, programme management took the courageous decision to stop before further losses, thus countering the sunk cost effect.

The ensuing revision pointed to several causes of failure. For example, the programme did not employ the idea of delivering integral functionality in manageable increments. Rather, it defined a total of only three excessively large projects and started with the largest and most complex of them. We also know that programme management did not find it worthwhile to update its skills on benefits management in the inception phase. The decision to halt the programme was made on the grounds of loss of control of costs, functionality, and architecture, rather than on explicit arguments of failure to deliver value for all that money.

Although the programme in the case above was halted before all was lost, this book offers techniques to help management stop even earlier, to save those EUR 36 million and even the EUR 180 million.

Benefits management [18] concerns an information system's entire life cycle. Since benefit is realized by using the system, benefits management concerns not just the system itself, but also how it is adopted and used in organizational and societal life.

Our focus is on techniques of benefits management that are performed during the development phases. We define our techniques in terms of incremental development, which involves stakeholders using – and obtaining benefit from – early deployed functionality. This means that the techniques do concern using the system, beyond developing it. The techniques are, however, for estimating and monitoring the system's expected benefit during these increments, and do not address organizational concerns as such.

Benefits management can be carried out in any software development model, including waterfall-based models. However, empirical results suggest that benefits management works better in a context with a flexible delivery scope, frequent deliveries, and extensive collection and use of feedback (see [5, 6] for pointers).

1.4 Benefit/Cost-Driven Development Methodology

A recent study [5] has found that projects that perceived themselves as successful in delivering the expected benefits differed from less successful ones, in that

- they applied benefits management practices before and during project execution,
- they applied core agile practices of frequent delivery to the client and scope flexibility,
- their clients were deeply involved in these practices.

This corroborates evidence from other empirical studies, suggesting that companies that engage in benefits management perform better in terms of most success criteria, especially those related to better project control and greater success in realized benefits [3, 6, 9, 12, 17]. Better project control, here, relates to updated information on projects' status and productivity.

However, benefits management has not achieved satisfactory uptake. In the words of Brees, Jenner, Serra, and Thorpe [2],

> It is now about 25 years since the emergence of benefits management, but hitherto it has had limited impact on project management and even less on general management practices. This is despite evidence that a focus on benefits improves the success rate of projects and programmes.

Respondents to Jørgensen's study [5] reported a lack of methodological support for benefits management. In particular, they experienced a lack of support in *quantifying the relation between planned returns and product elements*.

This book has been written to help you with that. However, just as management by cost alone is not enough, it would not be sensible to go to the other extreme and manage by benefit alone. The message in this book is, therefore, to combine cost management and benefits management.

In the following chapters, we will present techniques for estimating the benefit of the system under development and how that can be combined with cost estimates. We address a small but vital part of benefits management and provide numerical tools for use in various phases of benefits management.

1.5 Design

The development of our techniques follows the steps of design science [4]: design an artefact according to design principles, deploy the artefact to the field, learn from observations, and redesign. Here, the artefact is the techniques that we develop, and the design principles involve the following.

Concreteness: The techniques should be designed for performing concrete tasks. There can be many reasons why benefit estimation is not common; a lack of concrete techniques will leave project stakeholders and workers in the dark as to what to do, even if they grasp the general idea of benefits management.

Noninvasiveness: The techniques should be designed to be used in the existing process flow. If methods are too complex or too invasive in day-to-day work, they will not be employed. New techniques are often perceived as invasive, regardless.

Satisficing: The techniques should be designed to be *good enough* for the tasks at hand and in line with what Herbert Simon [14] calls *satisficing*, rather than optimizing. This point is essential for the simple and time-efficient use of techniques.

Support for cognitive processes: The techniques should be designed based on research in the field of judgement and decision making, to suit the nature of the cognitive processes involved in assessment.

Recognizability: The techniques should be reminiscent of existing techniques of state of practice to facilitate adoption.

Since there is a current lack of methodology, or at least a lack of reported use of any methodology, for conducting benefits management at the level and form presented in this book, there is no empirical evidence (systematic observations or analysis) to suggest precisely how effective our ideas are. Therefore, it is essential that projects start to use techniques so that the effects of benefits management can be evaluated. This book contributes such techniques. In due course, then, field studies can be conducted to evaluate the use of these techniques.

References

1. H.R. Arkes and P. Ayton, “The Sunk Cost and Concorde Effects: Are humans less rational than lower animals?” *Psychological Bulletin*, vol. 25, no. 5, pp. 591–600, 1999.
2. R. Breese, S. Jenner, C.E.M. Serra, and J. Thorp, “Benefits management: Lost or found in translation,” *International Journal of Project Management*, vol. 33, no. 7, pp. 1438–1451, 2015.
3. A. Budzier and B. Flyvbjerg, “Making sense of the impact and importance of outliers in project management through the use of power laws,” in *Proc. International Research Network on Organizing by Projects (IRNOP)*, vol. 11, 2013, pp. 228–232.
4. A.R. Hevner, S.T. March, and J. Park, “Design science in information systems research,” *MIS Quarterly*, pp. 76–106, Mar. 2004.
5. M. Jørgensen, “A survey of the characteristics of projects with success in delivering client benefits,” *Information and Software Technology*, vol. 78, pp. 83–94, 2016.
6. M. Jørgensen, P. Mohagheghi, and S. Grimstad, “Direct and indirect connections between type of contract and software project outcome,” *International J. Project Management*, vol. 35, no. 8, pp. 1573–1586, 2017.
7. M. Keil, J. Mann, and A. Rai, “Why software projects escalate: An empirical analysis and test of four theoretical models,” *MIS Quarterly*, vol. 24, no. 4, pp. 631–664, 2000.
8. M. Keil, A. Rai, J. Mann, and G. Zhang, “Why software projects escalate: The importance of project management constructs,” *IEEE Transactions on Engineering Management*, vol. 50, no. 3, pp. 251–261, 2003.

9. C. Lin and Y.C. Liu, “Evaluation issues in managing IS/IT outsourcing contracts: A study of large Australian organizations.” in *Collaborative Decision Making in the Internet Era*, 2005.
10. J. Lystad, “Det er ingen skam å snu – erfaringer fra Mattilsynet og NAV,” Presentation given at Conf. of the Agency for Public Management and eGovernment (Difi), Dec. 6, 2017.
11. S. Olaussen, Ø. Tendal, S. Johansen, V. Sem, S. Bråthen, H. Bremnes, E. Grubbmo, and A.D. Ræder, “KSP-rapport nr. 1 for Modernisering av IKT i NAV – Rapport til Finansdepartementet og Arbeids- og sosialdepartementet, Versjon: 1.0,” 2015.
12. C.E.M. Serra and M. Kunc, “Benefits realisation management and its influence on project success and on the execution of business strategies,” *Interntional J. Project Management*, vol. 33, no. 1, pp. 53–66, 2015.
13. F.R. Shapiro, *The Yale Book of Quotations*. Yale University Press, 2006.
14. H.A. Simon, *The Sciences of the Artificial*, 3rd ed. MIT Press, 1996.
15. B.M. Staw, “Knee-deep in the big muddy: A study of escalating commitment to a chosen course of action,” *Organizational Behavior and Human Performance*, vol. 16, no. 1, pp. 2–44, 1976.
16. B.M. Staw, “The escalation of commitment: An update and appraisal,” in *Organizational Decision Making*, Z. Shapira, Ed. Cambridge University Press, 1997, ch. 9, pp. 191–215.
17. A. ul Musawir, C.E.M. Serra, O. Zwikael, and I. Ali, “Project governance, benefit management, and project success: Towards a framework for supporting organizational strategy implementation,” *International J. Project Management*, vol. 35, no. 8, pp. 1658–1672, 2017.
18. J. Ward and E. Daniel, *Benefits Management: How to Increase the Business Value of Your IT Projects. 2nd Edition*. Wiley, 2012.
19. E. Zachariassen, “Nav stanser IT-prosjekt til 3,3 milliarder – Moderniseringsprogrammet var feil metode. Nav får skryt fra statlig ekspert,” article published Oct. 25, 2013.