ABSOLUTE SCALES TO EXPRESS STAKEHOLDER VALUE FOR IMPROVING SUPPORT FOR PRIORITIZATION

Lindsey Brodie and Mark Woodman
School of Engineering and Information Sciences, Middlesex University, Hendon, London, U.K.

Keywords: Stakeholder Value, Impact Estimation, Requirements Prioritization, Design Prioritization, Metrics, Value-based Software Engineering.

Abstract: Given the reality of resource constraints, software development always involves prioritization to establish what to implement. Iterative and incremental development methods increase the amount of prioritization required and introduce the need to support dynamic prioritization to identify high stakeholder value. Ideally the needs of all the stakeholders are considered in the priority decision-making and there might be negotiation amongst them. In this paper we argue that the current prioritization methods often lack adequate support for the prioritization process. Specifically that many methods fail to appropriately structure the data for stakeholder value, which results in explicit stakeholder value not being captured. This problem is often compounded by a lack of support for handling multiple stakeholder viewpoints. We propose an extension to an existing prioritization method, impact estimation, to move towards better capture of explicit stakeholder value and catering for multiple stakeholders. A key feature is the use of absolute scale data for stakeholder value. We use a small industry case study to evaluate this new approach. Our findings argue that it provides a better basis for supporting priority decision-making over the implementation choices for requirements and designs.

1 INTRODUCTION

Research into prioritization has increased in recent years with many new prioritization methods and variants being put forward. Much has been achieved in identifying the prioritization factors and the issues of concern when structuring prioritization data. However, existing prioritization methods and the prioritization data they utilize (in content and structure) continues to be insufficient to support the type of prioritization process that ideally needs to be adopted. Specifically, progress in improving the prioritization process seems hampered by inadequate conceptualizations of stakeholder value, in particular by the use of implicit notions of value. This is often compounded by an additional failure to support multiple stakeholder viewpoints. Note the term "stakeholder" is used here to mean any group of people with an interest in the system, and they can be identified by role and/or location.

In this paper, to move towards addressing the problems identified above, we propose capturing stakeholder value by stakeholder role, and using absolute scale data (as opposed to using, for example, ordinal scale data) for stakeholder value. We consider the explicit "real world" data captured by using absolute scales normally provides a better basis for supporting priority decision-making. For example, as we shall discuss later, it supports arithmetic calculations such as return on investment (ROI).

To present the argument for our proposals for stakeholder value, this paper is structured in the following way. Section 2 outlines the need for prioritization explaining why the prioritization process is important. Section 3 provides an overview of the existing research on prioritization and analyses how it relates to the problems we perceive impacting the prioritization of stakeholder value. Section 4 then investigates in detail how stakeholder value is currently expressed within the prioritization data and explains some of the resulting weaknesses. Finally, Section 5 briefly describes initial validation of using explicit absolute scale data for stakeholder value: a case study using value impact estimation (VIE). We have developed VIE as a simple extension to an existing method, impact estimation (IE). IE (Gilb, 2005) uses absolute scale data and

Brodie L. and Woodman M. (2010).
ABSOLUTE SCALES TO EXPRESS STAKEHOLDER VALUE FOR IMPROVING SUPPORT FOR PRIORITIZATION.
In *Proceedings of the Fifth International Conference on Evaluation of Novel Approaches to Software Engineering*, pages 48-57
DOI: 10.5220/0002932600480057

captures the impact of each of the potential designs on each of the requirements. VIE extends this to additionally capture explicit stakeholder value by stakeholder role. Our initial findings are that use of absolute scales is indeed beneficial for capturing stakeholder value, and that capturing stakeholder value by stakeholder role is helpful for decision-making. However, there remains considerable future work to develop adequate theory on stakeholder value and stakeholder viewpoints, and improve understanding of the prioritization process.

2 THE NEED FOR PRIORITIZATION

2.1 Lack of Guidance

Prioritization can be considered something of a "gap" in current software engineering. Certainly within the most commonly used system development methods, it has had far too low a profile in the past. Also industry standards such as the Integrated Capability Maturity Model (CMMI) (Boehm *et al.*, 2002) and SWEBOK (Software Engineering Book of Knowledge) (Bourque and Dupuis, 2004) fail to offer specific guidance on the prioritization process. This lack of attention matters because of the "bigger picture": the main purpose of prioritization is to help ensure projects are implementing the "right thing" at the "right time", while making good use of the always limited human, monetary and time resources. Opportunities to assist project planning, and so improve project delivery, are being lost if prioritization is not intelligently executed.

In addition, the demand to move towards value-based software engineering (VBSE) (Boehm, 2003) raises the need for greater attention to be paid to the delivery of stakeholder value. Indeed, Sullivan (2007) reports on a lack of "formal, testable and tested theories, methods, and tools to support economic-based analysis and decision-making (and value-based analysis more broadly)".

2.2 Changing needs for Prioritization

Moreover, recent developments in software development mean that prioritization can be seen today as having a more central, on-going role to play throughout systems development. In Waterfall methods, prioritization only has to be carried out once, early on in the systems development process, and involves deciding what requirements are to be in the system and what are not. However, prioritization processes now have to support iterative and incremental development (Larman and Basili, 2003). Such development requires on-going communication to capture data from the external environment, accept changing requirements, and receive feedback from each incremental delivery, in order to then establish what the stakeholders agree is of high value and should be in the next increment. This means the prioritization process has to cater for reuse of data while also accommodating changing data. Moreover, dynamic prioritization has to occur with each increment to determine what to implement next. Also that on-going identification of high stakeholder value is essential.

An additional demand comes from the recognition of the need for improved stakeholder understanding, especially the handling of multiple stakeholder viewpoints (Park *et al.* , 1999). There is a need to not only capture and present the different viewpoints, but also to enable stakeholder negotiation and tradeoffs, and to help achieve stakeholder consensus and buy-in (Davis, 2003). The prioritization process has a major part to play in providing better support to the system/product owners, who decide what shall be implemented.

3 EXISTING RESEARCH ON PRIORITIZATION

Research in the area of prioritization has increased in recent years. In 1997, Karlsson and Ryan (1997) wrote an influential paper describing their Cost-Value Approach based on the Analytic Hierarchy Process (AHP) (Saaty, 1990), which acted as a springboard for much subsequent research. In this section, we briefly review the existing literature on prioritization: listing the existing prioritization methods, the identified prioritization factors and some of the identified issues with structuring prioritization data. Concurrently, we analyse how this existing research relates to the problems we perceive in prioritizing stakeholder value.

3.1 Positioning of Prioritization

Aspects of prioritization are discussed in the IT literature under several subject areas including requirements prioritization (Karlsson *et al.* , 1998; Moisiadis, 2002; Berander and Andrews, 2005), release planning (Greer and Ruhe, 2004), architecture selection (Kazman *et al.* , 2001), COTS (Commercial Off-The-Self) selection (Mohamed *et al.* , 2007a), financial management (Favaro, 2002; Sivzattian, 2003), and decision-making and negotiation methods (Park *et al.* , 1999). There appears to be compartmentalization in the literature, which we argue needs questioning. While specialist areas for prioritization exist, it is essential that an overall view be considered because any given system encompasses many of these subject areas: there has to be interaction and integration at the system level. Accordingly, the stance taken by this research is that a holistic view should be taken: any overall prioritization process must include consideration of a wide range of prioritization data, which includes the fundamental software engineering concepts that we have termed here as "objective", "requirement", "design" and "increment". All these four concepts impact on the concept of stakeholder value. For example, carrying out a prioritization process using just the requirements without consideration of, say, the potential designs and the operational impacts, both of which affect the costs, needs to be questioned. See Figure 1, which shows an increment delivery cycle with iteration around these concepts as software development progresses.

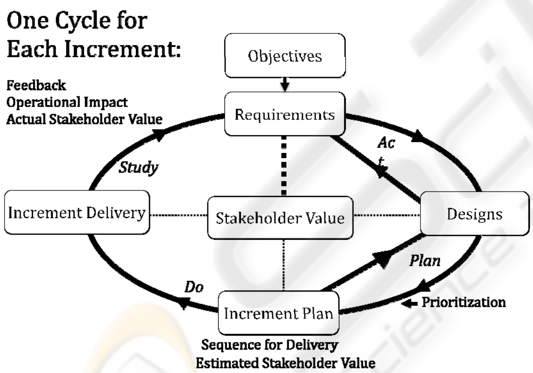

Figure 1: Increment delivery cycle based on Deming's Plan-Do-Study-Act (PDSA) Cycle.

Despite the previous argument, responsibility for the prioritization process and data model probably should reside within requirements engineering because it interfaces with the majority, if not all, of the stakeholders, and because the system requirements form the primary (but not sole) data for prioritization. However, care needs to be taken that there is adequate consideration of the wider aspects of the prioritization process that fall within other viewpoints, such as strategy management and operations management.

3.2 Existing Prioritization Methods

To date, we have identified over 60 different prioritization methods in the literature. For brevity, full discussion of these is not given. A selection of those found categorized by subject area is as follows:

- Requirements Prioritization: MoSCoW (Stapleton, 2003), the Hundred-Dollar Test (Berander, 2007) and Requirements Prioritization Tool (RPT) (Moisiadis, 2002).
- Requirements (and Effort) Prioritization: Cost-Value Approach (Karlsson and Ryan, 1997) and Wiegers' Method (Lehtola and Kauppinen, 2006).
- Requirements (and Design) Prioritization: Analytic Hierarchy Process (AHP) (Saaty, 1990), Quality Function Deployment (QFD) (Cohen, 1995; Akao, 1997) and Impact Estimation (IE) (Gilb, 2005).
- Architecture (Design) Prioritization: Cost Benefit Analysis Method (CBAM) (Kazman *et al.* , 2001) and Reasoning Frameworks (Bass *et al.* , 2005).
- COTS (Design) Prioritization: Procurement-Orientated Requirements Engineering (PORE) (Mohamed *et al.* , 2007a) and Mismatch Handling for COTS Selection (MiHOS) (Mohamed *et al.* , 2007b).
- Release Planning: Planning Game (Beck, 2000), EVOLVE/EVOLVE* (Greer and Ruhe, 2004; Saliu and Ruhe, 2005) and Requirements Triage (Davis, 2003).
- Financial Prioritization: Business Case Analysis/ROI (Favaro, 2003), Incremental Funding Method (IFM) (Denne and Cleland-Huang, 2004) and Real Options Analysis (Favaro, 2002).
- Negotiation Prioritization: Quantitative WinWin (Ruhe *et al.* , 2003) and Distributed Collaborative Prioritization Tool (DCPT) (Park *et al.* , 1999).
- Others: Conjoint Analysis (Green and Wind, 1975).

The prioritization methods given most coverage in the literature include AHP, QFD the Cost-Value Approach, and more recently, the Planning Game.

However, it is not clear to what extent all these methods are used by software development in industry, or indeed how successful they have been. Lehtola (2006) considers, "Even though many authors have evaluated prioritization approaches and

Table 1: Prioritization factors by stakeholder viewpoint and software engineering concept.

STAKEHOLDERS	Strategy management	Systems development		Operations management
CONCEPTS	Organizational objectives (objective)	Systems requirements (requirement)	Design solutions (design)	Delivery plans: increment plans & increment deliveries (increment)
PRIORITIZATION FACTORS				
OPINION	Vision/intuition/gut feeling/preference/bias	Preferences/bias/ importance	Intuition/ preferences/bias	Preferences/bias
STRATEGY	Strategic alignment/ business objectives/ product strategy		Long-term Strategy for systems architecture	
	Competition	Quality		
	Customer demand	Originator of requirement		End user value
	New business potential			
TIME	Urgency/time to market/lead time		Time schedule/ time constraints	
	Long term versus short term		Long term versus short term	
LEGAL	Legal mandate/ regulations	Legal mandate/ regulations		
	Contracts in place			
FINANCIAL BENEFIT	Market value/price			
	Financial benefits			
	Financial penalties			
	Benefit/cost ratio			
	Cost of not implementing			
COST	Development costs/ implementation costs/ support costs		Development costs/ support costs	Implementation costs/ support costs
	Operational costs			Operational costs
FIT	Fit with operational context: . business processes . skills/training . delivery timing		Staff competence Balanced workload	Fit with operational context: . business processes . skills/training . delivery timing
			Resource availability/ effort constraints	Resource availability/ effort constraints
	Fit with other products		Change impact/ base code dependencies	Change impact
			Logical implementation order	
			Reuse potential	
EXTERNAL DEPENDENCY	Intermediary channels	External dependencies	External dependencies	
RISK	Business risk Sales barriers	Volatility of requirements	Technical risk in: . current system . proposed system . implementation process	
			Difficulty of implementation/	Difficulty of implementation/

done comparison studies with them, the suitability of the approaches for solving practical product development challenges in prioritization has not been widely studied. It is even not clear if any of the techniques can solve the existing challenges in the area of requirements prioritization." Indeed, there appear to be some problems with the take-up and continued use of the well-known prioritization methods, such as QFD (Martins and Aspinwall, 2001) and AHP (Lehtola and Kauppinen, 2006).

3.3 Prioritization Factors

There are many prioritization factors (also sometimes called "criteria" (Wohlin and Aurum, 2005; Barney *et al.* , 2008) or "aspects" (Lehtola, 2006; Berander, 2007)) that can be considered in the prioritization process. We have identified a list of over 50 prioritization factors from the literature; the main sources include (Carlshamre *et al.* , 2001; Moisiadis, 2002; Greer and Ruhe, 2004; Firesmith, 2004; Berander and Andrews, 2005; Lehtola and Kauppinen, 2006; Barney *et al.* , 2008). See Table 1, in which we chose to sub-divide the factors into three categories by stakeholder viewpoint and note the similarity to the choices of Lehtola (2006) and Barney*, et al.* (2008).

For brevity here, we have limited discussion of our work to just three stakeholder viewpoints that are representative of the mandatory viewpoints in any systems development prioritization process: strategy management, systems development and operations management (Clearly there are many more stakeholder roles than these in a system.) We added further sub-division under the four software engineering concepts used earlier in Figure 1. We decided that: strategy management has responsibility for the objectives, systems development is primarily responsible for the requirements and designs, and operations management has responsibility for accepting the planned and delivered increments. In other words, the data associated with the selected system concepts would be of prime interest to the stakeholder viewpoint when establishing priorities. Furthermore, we introduced grouping of the prioritization factors by concept area, for example, strategy, cost and risk. Several of these groups are also identified by Berander (2007) as “aspects”. Note that, due to space limitations, any explanations of individual prioritization factors and relevant references have been omitted. Note also that these prioritization factors are not complete; this table only reflects the main prioritization factors found in the literature. The following observations can be made:

- A general set of prioritization factors that could be proposed as “a starter” for a prioritization process emerges from the table.
- The prioritization factors span all the four software engineering concepts. This argues for a prioritization process that offers support for all these concepts. If more narrowly focused, specialized, prioritization methods are to exist then they need to integrate into an overarching prioritization process/method. The table provides support for the existence of different stakeholder viewpoints in the mappings between the stakeholder viewpoints and the prioritization factors: different stakeholder viewpoints are interested in and knowledgeable about different prioritization factors. This means any prioritization process or prioritization method must cater for different stakeholder viewpoints.
- A tentative observation can be made that the prioritization factor groupings (for example, strategy, legal, cost and risk), map across to the dimensions for stakeholder value.

3.4 Known Issues in Structuring Prioritization Data

A list of issues encountered when structuring the prioritization data to support the prioritization process was identified by extrapolating from discussions in the literature, for example from (Karlsson *et al.* , 1998; Carlshamre *et al.* , 2001; Moisiadis, 2002; Davis, 2003; Firesmith, 2004; Lehtola and Kauppinen, 2006; Gorschek and Wohlin, 2006; Mead, 2006). The issues considered relevant to expressing prioritization data include:

Explicit Stakeholder Value. This is the often the expression of stakeholder priority to reflect the stakeholder value as well as the capture of explicit value.

Multiple Stakeholder Viewpoints. There is a need to handle different areas of interest/expertise and capture the different viewpoints together with their associated stakeholder values.

Requirements Abstraction. This is the ability to handle requirements captured at different levels of refinement.

Interdependencies. The ability to express interdependencies among the requirements and also the designs. This becomes increasingly important with iterative and incremental development.

Dynamic Prioritization. The priority data must be captured in order that it can be reused in subsequent prioritizations (future increments) without needing further inputs from stakeholders (unless something significant has changed in the system and/or its environment that they need to provide additional data on).

Scaling-up. This is the ability to scale up to cope with large numbers of items. Some existing prioritization methods become impractical when the number of requirements begins to grow to sizes typical of modern systems. In fact, for most large-scale projects, prioritization can tend to be carried out at a fairly high level of abstraction.

4 ANALYSIS OF EXISTING PRIORITIZATION DATA

4.1 Expressing Prioritization Data

How prioritization data is expressed is a key factor in a prioritization process. We argue in this section that the prioritization data that the prioritization methods currently utilize (in content and structure) is insufficient to support the type of enhanced prioritization process that ideally needs to be adopted. Specifically, the lack of use of quantified data captured on absolute scale types is hindering progress.

The type of scale being used to capture the data is specifically important as it identifies the extent to which arithmetic calculations can validly be carried out. Only the ordinal and ratio scale types are commonly used in existing prioritization methods. The absolute scale type is only occasionally used at present, but we propose it should be much more widely used and in fact, that it should replace much of the use of the ordinal and ratio scale types.

Table 2: Mapping of prioritization technique(s) and scale type(s) to prioritization methods.

Prioritization Method	Prioritization Technique(s)	Scale Type(s)
QFD	Weighting and Grouping	Ratio Ordinal
AHP	Weighting (Pair-wise comparison)	Ratio
IE	Metrics	Absolute
Cost-Value Approach	Weighting (Pair-wise comparison)	Ratio
MoSCoW	Grouping	Ordinal
Planning Game	Grouping	Ordinal
Requirements Triage	Grouping and Weighting	Ordinal Ratio

4.2 Prioritization Techniques

Several different ways (sometimes termed "prioritization techniques" (Berander, 2007)) of expressing prioritization data can be identified (Firesmith (2004); Berander and Andrews (2005)). We have reduced the number of different categories to four main ones as follows:

Grouping. The individual items are each categorized into one of a set of priority groups, for example the MoSCoW prioritization method demands each requirement is categorized as either "must have", "should have", "could have" or "would like, but wouldn't have this time" (Stapleton, 2003). The results are on an ordinal scale.

Ranking. Requirements are ranked in order of preference. Ranking is carried out by bubble sort or by binary search tree (Karlsson *et al.* , 1998). This is an ordinal scale of measure as there is no information about the differentials amongst the ranked items.

Weighting. Stakeholders assign their preferences and relative weightings are calculated. The results are on a ratio scale. One means of obtaining the weightings is by using voting (Berander and Andrews, 2005): stakeholders are requested to distribute some fixed number of votes (say 100 or 1000 dollars) amongst the different items being prioritized. Another means is by using pair-wise comparison: priorities are calculated by creating a hierarchy with branches of up to seven comparable items and then the items within each branch are pair-wise compared using a scale of 1 to 9 where 1 equates to "equally important" and 9 equates to "extremely more important" (Moisiadis, 2002). The scales are then converted to normalized weightings, which are then carried up the hierarchy. In AHP, pair-wise comparison is used to first weight the requirements, and then the designs.

Metrics. Absolute scales of measure are used to express certain attributes and these metrics form the basis for selection, for example by enabling calculation of ROI figures (Firesmith, 2004). ROI calculation needs data on the amount of benefit (stakeholder value) that would be achieved by implementing a given design and the implementation cost associated with it. Only absolute scale data enables such ROI estimates to be calculated, as explicit stakeholder value data such as "a cost saving over the next year of 220,000 monetary units" would be captured. This contrasts to the ordinal scale data of say, the MoSCoW method, which simply captures requirements identified as of high stakeholder value into a "must have" priority group. In this paper, we are using Planguage (Gilb, 1988) to express metrics, which captures the performance and resource requirements, as required levels on scales of measure.

See Table 2, which gives some examples of how the scale types and prioritization techniques map to a selection of prioritization methods.

See also Table 3, which shows how the prioritization techniques cope with a selection of prioritization data issues. Some example data has been inserted in the top row. From this row, it can be seen that use of metrics with absolute scale types results in real data that is much easier to understand and say, discuss with another stakeholder. It is less ambiguous than trying to work out what "Medium" should be interpreted to mean. An observation can be made that all the techniques, apart from metrics, are generating additional data that captures some indirect notion of stakeholder value (such as "must have"), but not any explicit value (such as 220,000 monetary units).

5 SOME EXAMPLES FROM A CASE STUDY

5.1 Choice of Prioritization Method

By comparing how prioritization methods handled the prioritization factors and the data structure issues (Brodie and Woodman, 2008), and by considering

Table 3: How prioritization techniques cope with a selection of data structuring issues.

Prioritization Technique:	**Grouping**	**Ranking**	**Weighting**	**Metrics**
Example of prioritization data	ÒHighÓ, ÒMediumÓor ÒLowÓ	1, 2, 3, ... N	30/100	Time to carry out task to be reduced from 1 day to 5 minutes
Data Structuring Issue ↓				
Stakeholder value	Implicit; value is say, ÒMediumÓ	Implicit; value is say, ranked as Ò2Ó	Implicit; value is 30% of whatever 100% equates to	For above metric, an estimate of explicit financial value can be derived if monetary rate of pay and task occurrence are known
Multiple stakeholder viewpoints	N Would be represented as say, ÒHighÓand ÒMediumÓ	N Would be represented as say, Ò1Óand Ò20Ó and Ò3Ó	N Would be represented as say, 30/100 and 2/100	Y Time to carry out task to be reduced from 1 day down to say, 5 minutes and to 2 hours
Interdependencies	(Y) Would have to work by selecting an item and then seeing if there were any prior dependencies that would override	(Y) Ditto	(Y) Ditto	(Y) Ditto
Abstraction	N	N	Y Create hierarchy	Y Create hierarchy
Dynamic prioritization	Y Add any new data to an existing data grouping. No extra effort (unless something has changed)	Y Would need to re-examine existing ranks	(Y) Considerable effort needed by stakeholders	Y Add to existing data and reprocess
Scaling-up	N Too many in a group	N Difficult to keep track	N Considerable effort	(Y) Would use high-level

the usage of software engineering concepts and scale types, we determined that IE offered an initial sound basis for this research: it spans the concepts of requirement, design and increment, and uses absolute scale data (Gilb, 2005). However, the IE method lacks consideration of explicit stakeholder value, so we extended it to cater for stakeholder value by stakeholder role. We chose to link stakeholder value to requirement. See Figure 2 for an example of an extended IE table

5.2 Case Study Description

The case study examples are from a customer business rules “decisioning” system for a bank. The bank’s objectives are customer satisfaction and, more efficient and effective internal processes. The main problems perceived by the bank are the time, effort and accuracy of updating and using the business rules, and the elapse time taken and the accuracy of dealing with customer requests. Of course, having up-to-date business rules in place impacts the accuracy of the handling of the customer requests. As the intention is to demonstrate that absolute scale data helps prioritization reasoning, a detailed discussion of all the requirements is not given here. We also limit our comments here about the use of performance requirements (also known as non-functional requirements) apart from recognizing that this is an additional reason why IE merits attention (given very few prioritization methods handle performance requirements (Firesmith, 2004)).

For brevity, a very restricted, cut down sample of the system specification is presented below. Note the data highlighted in bold in this specification is captured in the extended IE table in Figure 2.

Stakeholders: Regulator, IT Department, Customer, Rules Administration, Business Units, Back Office.

Requirements:
Function: Submit request.
Performance requirement: **Reduce time for customer to submit request.**
Scale: Average time taken for defined [request type: Default = Loan].
Past: **30 minutes.**
Goal: **10 minutes.**

Function: Enter customer request details.
Performance requirement: **Reduce time for Back Office to enter request.**
Past: **30 minutes.**
Goal: **10 minutes.**

Function: **Process a customer request.**
Performance requirement: **Reduce time to process customer request.**
Past: **5 days.**
Goal: **20 seconds.**
Performance requirement: Reduce number of complaints.
Scale: Average number of complaints in defined [Time] from defined [Stakeholder].
Past **[Back Office]: 10 per week.**
Goal: **0 per week.**
Past **[Customer]: 25 per week.**
Goal: **5 per week.**

Function: **Update the business rules.**
Performance requirement: Reduce time to

Stakeholder Value

Bank System
By End Date: dd/mm/yyyy

Key:
s = seconds
m = minutes
d = days
w = week

Designs by expected Increment with design dependencies

Regulator	IT Dept.	Customer	Rule Admin.	Business Units	Back Office	Total Value / Benefit	Requirements	1: D1: Automate Rules + Manual Testing	2: D2: Back Office Loan Decisioning	3: D3: Web Self-Service	4: D4: Automate Rules + Automate Testing
		4				4	R1 Time for customer to submit request 30 min <-> 10 min	-	-	10 m 100%	-
					3	3	R2 Time for Back Office to enter request 30 min <-> 10 min	-	-	0 m 150%	-
		9		9		18	R3 Time to respond to customer request 5 days <-> 20 seconds	-	1 d 80%	20 s 100%	-
					1	1	R4. No of Back Office complaints 10 per week <-> 0	5 50%	<1 90%	0 100%	(2) (80%)
		1			5	6	R5 No of customer complaints 25 per week <-> 5	-	15 50%	5 100%	-
1			5	4	8	18	R6 Time to update business rules 1 month <-> 1 day	2 w 50%	-	-	1 d 100%
1			3	4	6	14	R7. Time to distribute business rules 2 weeks <-> 1 day	1 d 100%	-	20 s 103%	-
2		14	8	17	23	64	Cumulative Total for Performance Requirements	200%	170%	280%	50%
							Design Cost (M)	0.2	0.3	1.0	0.5
							Development Budget 2.5M <-> 300K	2.3	2.0	1.0	0.5
							Cumulative Perf to Devt Cost Ratio	1000	567	280	100
							Cumulative Stakeholder Value to Development Cost Ratio	23.5/0.2 =117.5	17.8/0.3 =59.3	13.7/1.0 =13.7	9/0.5 =18

Figure 2: VIE table for bank case study. The shaded area represents the extension to IE.

update rules.
Scale: Average time taken for defined [request type].
Past: **1 month**.
Goal: **1 day.**

Function: **Distribute business rules.**
Performance requirement: Reduce time taken. Scale: Average time taken.
Past: **2 weeks**.
Goal: **1 day**.

Designs:
APTM: **Automate the rules & test manually.**
Rationale: Speed up the distribution to Back Office staff.

BD: **Back Office loan decisioning system.**
Rationale: Automating applying the rules will save time.
Dependency: **APTM.**

WSS: **Web self-service.**
Rationale: Customers can get a rapid response.
Dependency: **BD, APTM.**

APAT: **Automate the rules & test automatically.**
Rationale: Speed up the distribution to Back Office staff.
Dependency: **APTM.**

5.3 Analysis of the VIE Table

Figure 2 also captures the estimated impacts of the various designs on the requirements and the estimated value for each stakeholder of achieving 100% of the requirement. When feedback is obtained after implementing an increment, the actual figures can be added into the table, compared to the estimates and any deviations from the plan can be addressed.

Stakeholder value is determined as figures of merit on the basis of the monetary value of additional customer sales and the cost savings due to reduction in effort. An initial assumption is made that the utility curves (Daniels *et al.* , 2001) for value against the requirement levels are all linear.

The extended IE table also shows an initial proposed ordering of the designs into increments. The bold arrows show the design dependencies. Note the impact of a design can be overridden by the impact of earlier designs: the impact of D4 Automate Rules and Automate testing on the requirement R4 No. of Back Office complaints is such a case and the estimated figures are therefore shown in brackets. The development costs (but not the operational implementation costs) are also shown.

Using these figures the cumulative stakeholder-to-development cost ratio for each of the designs was calculated. For example, for the design D2 Back office loan decisioning, the ratio was calculated as follows: (80% (impact) of 18 (total value) for requirement R3 Time to respond to customer request

+ 40% (impact = 90% – 50%) of 1 (total value) for requirement R4 No. of Back Office complaints + 50% (impact) of 6 (total value) for requirement R5 No. of customer complaints) divided by 0.3 (the development cost) = (14.4 + 0.4 + 3) / 0.3 = 17.8/0.3 = 59.3, shown at the bottom of the column for design D2.

In the calculations, all impacts were capped at 100% as that represented what the customer had requested (Of course, the customer might be interested in the additional impact especially if there is no additional cost associated with achieving it.)

By reviewing the cumulative stakeholder value to development cost ratio, it looks as if the design "Automate rules and automate testing" would be of more value than implementing "Web self-service". However, it could be that the utility curve for value for the requirement "Time to respond to customer request" is not linear. Further discussion is need with the stakeholders. Such negotiation is exactly the purpose of using an extended IE table. Note how where the stakeholder value resides can be seen and how it is possible to detect which are the "valuable" requirements.

6 CONCLUSIONS

Despite much research in the last decade on prioritization in software engineering projects, progress is being hampered by inadequate representation of stakeholder value. The issue is becoming more urgent because dynamic prioritization of stakeholder value is increasingly needed as iterative and incremental development methods become more widely used. We have argued in this paper that the use of absolute scale data is essential to address the problems with the current prioritization processes: specifically, to provide unambiguous prioritization data that stakeholders can understand and relate to, and to support arithmetic calculations.

This paper has briefly reviewed existing research on prioritization. Our findings include:

- By categorizing and analysing the existing prioritization methods, that many (not all of) the existing prioritization methods are restricted in their scope (for example, some methods are just considering the requirements).
- By investigation of the prioritization factors discussed in the literature, we have shown that the scope of the prioritization process spans system-wide data from organizational objectives to increment delivery. An additional finding from this data is that different stakeholders have different viewpoints on the prioritization factors, and that therefore, multiple stakeholder viewpoints need to be supported.
- By identifying the known issues with structuring prioritization data and analyzing how the prioritization techniques and scale types used in prioritization methods tackle these issues, we determine that the techniques of grouping, ranking and weighting are weaker than metrics in addressing the issues. Specifically, expression of stakeholder value is implicit in the prioritization data, and that arithmetic calculations are often impossible or problematic, apart from when metrics are used.

Further, we demonstrate the validity of the use of absolute scale data in prioritization by using VIE, an extended version of the IE prioritization method with some examples from a case study. We specifically extended IE to cater for stakeholder value for multiple stakeholders. We show the ability to carry out calculations to investigate requirement and design priorities.

Work is underway to investigate further extending the IE method to represent additional aspects of stakeholder value. Future work plans to make the detailed decision-making of a rational prioritization process explicit.

REFERENCES

Akao, Y., 1997. "QFD: Past, Present, and Future", *Procs. of the International Symposium on QFD '97*, Linkoping.

Barney, S., Aurum, A., Wohlin, C., 2008. "A product management challenge: Creating software product value through requirements selection", *Journal of Systems Architecture*, Elsevier.

Bass, L., Ivers, J., Klein, M., Merson, P., 2005. "Reasoning Frameworks" (CMU/SEI-2005-TR-007), *Software Engineering Institute, CMU.*

Beck, K., 2000. *Extreme Programming Explained: Embrace Change*, Addison Wesley.

Berander, P., 2007. *Evolving Prioritization for Software Product Management*, Blekinge Institute of Technology. Doctoral Dissertation Series.

Berander, P., Andrews, A., 2005. "Requirements Prioritization" (Chapter 4), *Engineering and Managing Software Requirements*, Aurum and Wohlin (Eds.), Springer-Verlag. ISBN 3540250433.

Boehm, B., 2003. "Value-Based Software Engineering", *Software Engineering Notes*, Vol. 28, No. 2, ACM.

Boehm, B., Port, D., Basili, V.R., 2002. "Realizing the Benefits of the CMMI with the CeBASE Method", *Systems Engineering*, J. Wiley and Sons, Inc.

Bourque, P., Dupuis, R. (eds.), 2004. *SWEBOK: Guide to the Software Engineering Body of Knowledge 2004 Version*, IEEE Computer Society.

Brodie, L., Woodman, M., 2008. "Towards a Rational Prioritization Process for Incremental and Iterative Systems Engineering", *Procs. of the 1st International Workshop on Requirements Analysis*, Pearson.

Carlshamre, P., Sandahl, K., Lindvall, M., Regnell, B., Natt och Dag, J., 2001. "An Industrial Survey of Requirements Interdependencies in Software Product Release Planning." *Procs. of the 5th IEEE International Symposium on RE.*

Cohen, L., 1995. *Quality Function Deployment: How to Make QFD Work for You*, Addison Wesley.

Daniels, J., Werner P.W., Bahill, A.T., 2001. "Quantitative Methods for Tradeoff Analyses", *Systems Engineering*, John Wiley & Sons, Inc., 4, 3.

Davis, A., 2003. "The Art of Requirements Triage", *IEEE Computer,* March 2003, 36, 3, pp. 42-49

Denne, M., Cleland-Huang, J., 2004. *Software by Numbers: Low-Risk, High-Return Development*, Prentice-Hall. ISBN 0131407287. 190 pages.

Favaro, J., 2003. "Value-Based Management and Agile Methods", *M.Marchesi and G. Succi (Eds.): Procs. of the 4th International Conference on XP and Agile Methods*, May 2003, Springer-Verlag.

Favaro, J., 2002. "Managing Requirements for Business Value", *IEEE Software*, March/April 2002.

Firesmith, D., 2004. "Prioritizing Requirements", *Journal of Object Technology*.

Gilb, T., 2005. *Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage*, L. Brodie (Ed.), Butterworth-Heinemann. ISBN 0750665076.

Gilb, T., 1988. *Principles of Software Engineering Management*, Addison Wesley. ISBN 0201192462.

Gorschek, T., Wohlin, C., 2006. "Requirements Abstraction Model", *Requirements Engineering*, Springer Verlag, 11, pp. 79-101.

Green, P.E., Wind, Y., 1975. "New Way to Measure Consumers' Judgments", *Harvard Business Review*.

Greer, D., Ruhe, G., 2004. "Software release planning: an evolutionary and iterative approach", *Information and Software Technology*, 46, 4, pp. 243-253.

Karlsson, J., Ryan, K., 1997. "A Cost-Value Approach for Prioritizing Requirements", *IEEE Software*.

Karlsson, J., Wohlin, C., Regnell, B., 1998. "An Evaluation of Methods for Prioritizing Software Requirements", *Information and Software Technology*, Elsevier Science B.V.

Kazman, R., Asundi, J., Klein, M., 2001. "Quantifying the Costs and Benefits of Architectural Decisions", *Procs. of the 23rd International Conference on Software Engineering (ICSE 2001)*, IEEE.

Larman, C., Basili, V.R., 2003. Iterative and Incremental Development: a Brief History, *IEEE Computer*, 36, 6.

Lehtola, L., 2006. *Providing value by prioritizing requirements throughout software product development: State of practice and suitability of prioritization methods*, Licentiate thesis, Helsinki University of Technology.

Lehtola, L., Kauppinen, M., 2006. "Suitability of Requirements Prioritization Methods for Market-driven Software Product Development", *Software Process Improvement and Practice*, Wiley.

Martins, A., Aspinwall, E., 2001. "Quality Function Deployment: an empirical study in the UK", Total Quality Management, August 2001.

Mead, N., 2006. "Requirements Prioritization Introduction", *Software Engineering Institute (SEI), Carnegie Mellon University (CMU)*.

Mohamed, A., Ruhe, G., Eberlein, A., 2007a. "COTS Selection: Past, Present, and Future", *Procs. of the IEEE Intl. Conf. and Workshops on the Engineering of Computer-Based Systems (ECBS'07), IEEE.*

Mohamed, A., Ruhe, G., Eberlein, A., 2007b. "Decision Support for Handling Mismatches between COTS Products and System Requirements", *Procs. of the COTS-Based Software Systems (ICCBSS'07) Conf.*

Moisiadis, F., 2002. "The Fundamentals of Prioritising Requirements", *Procs. of the Systems Engineering, Test and Evaluation Conference*.

Park, J., Port, D., Boehm, B., 1999. "Supporting Distributed Collaborative Prioritization for Win-Win Requirements Capture and Negotiation", *Procs. of the International Third World Multi-conference on Systemics, Cybernetics and Informatics (SCI'99)*, International Institute of Informatics and Systemics.

Ruhe, G., Eberlein, A., Pfahl, D., 2003. "Trade-off Analysis for Requirements Selection", *International Journal of Software Engineering and Knowledge Engineering*, 13, 4, pp.345-366.

Saaty, T.L., 1990. "How to Make a Decision: The Analytic Hierarchy Process", *European Journal of Operational Research*, 48, 1990, pp. 9-26.

Saliu, O., Ruhe, G., 2005. "Supporting Software Release Planning Decisions for Evolving Systems", *Procs. of the 2005 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05),* IEEE.

Sivzattian, S.V., 2003. *Requirements as Economic Artifacts: A Portfolio-Based Approach*, Ph.D. Thesis, Department of Computing, Imperial College of Science, Technology and Medicine, London.

Stapleton, J. (Editor), 2003. *DSDM: Business Focused Development* (2nd Edition), Addison Wesley.

Sullivan, K., 2007. Introduction to the First Workshop on the Economics of Software and Computation, *In Companion to the Procs. of the 29th International Conf. on Software Engineering*, IEEE.

Wohlin, A., Aurum, A., 2005. "Criteria for Selecting Software Requirements to Create Product Value: An Industrial Empirical Study", S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, P. Grunbacher (Eds.), *Value-Based Software Engineering*, Springer.