

Nina D. Fogelström, Sebastian Barney, Aybüke Aurum, Anders Hederstierna,
“When Product Managers Gamble with Requirements: Attitudes to Value and
Risk,” 15th International Working Conference on Requirements Engineering:
Foundation for Software Quality (RefsQ), 8–9 June 2009, Pages: 1–15

When Product Managers Gamble with Requirements: Attitudes to Value and Risk

Nina D. Fogelström¹, Sebastian Barney¹, Aybüke Aurum², and Anders Hederstierna³

¹ School of Engineering, Blekinge Institute of Technology
nino.dzamashvili.fogelstrom@bth.se, sebastian.barney@bth.se

² School of Information Systems, Technology and Management,
University of New South Wales
aybuke@unsw.edu.au

³ School of Management, Blekinge Institute of Technology
anders.hederstierna@bth.se

Abstract. [**Context and motivation**] Finding a balance between commercial (customer specific, market pull and external quality requirements) and internal quality requirements is a recognized challenge in market driven software product development (MDSPD). In order to address this challenge it is important to understand the preferences and biases influencing decision makers selecting requirements for software releases. [**Question/problem**] Prospect theory has been successfully applied to many disciplines. Applying it to MDSPD suggests decision makers will avoid risk when selecting between commercial requirements, take risk with internal quality requirements, and prefer commercial requirements over internal quality requirements in order to maximize their perceived value. This paper seeks to investigate this claim. [**Principal ideas/results**] This paper presents an experiment investigating whether the biases proposed by prospect theory can be seen operating in MDSPD requirements engineering (RE). The results indicate risk avoidance when dealing commercial requirements, while greater risk is taken when dealing with internal quality requirements. [**Contribution**] As this is the first paper to use prospect theory to explain requirements selection decisions, it presents opportunity to educate people in the biases they bring to the RE process, and facilitate the creation of strategies for balancing the different requirements types.

Key words: requirements selection, prospect theory, value, risk

1 Introduction

Requirements engineering (RE) is a decision rich activity, and RE decision support as a field has received increasing attention as market-driven software product development (MDSPD) has taken off [1,2,3]. Due to the growing popularity and complexity of MDSPD, it is important to understand how inherent biases influence decisions made in this area. This paper explores one key bias presented through prospect theory, and how it impacts value and risk perception.

MDSPD is focused on the task of selecting a set of requirements that should be included in the coming releases of a product. The success of the product is measured in sales and is related to how much the product features are valued by the customers.

The requirements selection process in MDSPD is often perceived as a very complex activity. This complexity is explained by:

- A large number of requirements that need to be considered;
- A variety of different technical aspects that need to be taken into account before a selection of the requirements can be made; and
- The challenge associated with taking decisions based on the decision material of variable quality (such as uncertain cost and value estimations).

Thus, the authors of this paper classify requirements selection as occurring in a high-risk environment.

Requirements selection in MDSPD often involves situations where a decision maker must choose between requirements of different types. Key requirement types include *commercial requirements* – originating from the market and key customers; and *system requirements* (internal quality requirements) – often connected with system maintenance, quality and architecture. The challenge here is to find a balance between commercial and system requirements, such as to allow satisfying the immediate market and customer needs, as well as assuring the healthy product architecture evolution.

Recent studies on MDSPD show that commercial requirements are perceived more important than system requirements [4,5,6,7]. However, these studies also show that system requirements are generally undervalued. Considering the impact of requirements selection decisions on the business of a development company, it would be useful to have an understanding of the heuristics and cognitive biases behind these decisions, which according to the classical decision theory govern the decision outcome [8].

In this paper prospect theory [9,10] is used to explore biases behind requirement selection decisions in a MDSPD context. While this breaks normative assumptions about decision making in risky situations referenced in most engineering literature [11], decisions made by software project managers have found to be better modelled by prospect theory than the more common utility theory.

Prospect theory models decision makers' attitudes towards risk in terms of gains and losses [8]. It predicts that people will be risk adverse when selecting between options described in terms of gains, and risk taking when deciding between options described in terms of losses. Given that software requirements in MDSPD RE can be described in terms of revenues (gains) and/or costs (losses) it seems interesting to apply prospect theory in order to explain the choices of decision makers.

Assuming that in MDSPD commercial requirements are normally associated with revenue, while system requirements are associated with costs, applying prospect theory would suggest decision makers will take risks with system requirements. As these requirements are associated with cost, the preference is to

delay spending in the short-term, despite the potential for increased costs later and thus taking a risk.

Conversely, since commercial requirements are associated with gains, decision makers will be more risk adverse when selecting and prioritizing these requirements. This means preference will be given to requirements with a more reliable revenue stream, even if the alternatives could yield greater ROI.

In the experiment presented in this paper the authors investigate how well prospect theory applies to MDSPD RE. To the best of the author's knowledge this is the first time prospect theory has been applied to MDSPD. The primary objective is to investigate the decision maker's attitude towards risk, as well as perception of requirements value when it comes to choosing between alternatives of both cost and benefit.

The structure of the paper is as follows; Section 2 presents the special characteristics of MDSPD and provides an overview of how prospect theory can be used to explain and model requirements selection decisions in an MDSPD environment; Section 3 details the research question and the experiment design; the experiment results and analysis are found in Section 4 and finally Section 5 presents conclusions and future work.

2 Background

The following section provides a brief summary of the characteristics of requirements selection process in MDSPD. This is followed by a presentation of prospect theory and an experiment that explore attitudes towards losses and gains. The section concludes with a discussion on how prospect theory can be used to explain requirements selection decisions in MDSPD.

2.1 Market Driven Software Product Development

In typical MDSPD the development organization is the main stakeholder and owner of a developed product. The product evolves over time with new functionality added and offered to the general market through product releases. In MDSPD the development organization takes most risks and is responsible for development costs of the product. Delivering the right set of functionality when the market is ready for it is critical for the success of the software product.

Market-driven development is largely product focused and many important activities are performed prior to the initiation of the development projects (i.e. pre-project). The goal of pre-project activities is to catch, analyse, select and plan requirements for future releases of the product. Pre-project activities should result in a prioritized list of requirements that are assigned to a specific release. These requirements are then realized through one or more development projects.

The importance of correct decision making pre-project is acknowledged and highlighted by both industry and academia. However, finding a perfect set of requirements is considered impossible due to a variety of different criteria that need

to be taken into account and conflicting interests of stakeholders. Each stakeholder group is interested to see their requirement in the next release [12,13,14].

Different requirement types that are typically involved in requirements selection process are shown in Table 1 [2,15,16]. *Customer specific features* and *Market pull requirements*, which are often referred to as *commercial requirements* originate from the customers. In the case of *Customer specific features* the main stakeholder is one key customer, whereas for *Market pull requirements* the main stakeholder is often a marketing or sales unit representing a number of customers.

Table 1. Requirement Types in MDSPD

Requirement Type	Explanation
Customer specific features	Requirement originating from one of the key customers
Market pull requirements	Requirements originating from more than one customers that represent a certain market
Innovation	Requirement usually originating from the company's R&D. Examples of this may be innovative technology, patents, innovative features, etc.
External quality aspects	Requirements concerning usability, security, reliability and other quality issues
System requirements (Internal quality aspects)	These are system architecture specific requirements, connected to system architecture health, flexibility, and evolution.

Innovation requirements can originate from *Key customers*, however mostly these requirements are defined by the R&D unit of a company.

The stakeholders of *External quality aspects* can be represented by both external parties, such as customers and internal parties such as R&D. Finally, *System requirements* mainly focus on system architecture and evolution aspects and almost always originate from organisations' R&D units.

Recent studies on MDSPD show that commercial requirements are perceived more important than system requirements [4,5,6,7]. However, these studies also show that system requirements are generally undervalued. The situation where system requirements are usually given a lower priority is troubling system engineers as it will lead to system architecture deterioration and collapse in the longer term, which can be connected to serious losses for the company.

Finding a trade-off and profitable balance between commercial and system requirements is very important for the success of a product. But before this trade-off can be found, the authors believe an understanding should be reached of the reasoning and decision frames that govern decision making process in MDSPD RE. This is where the authors believe prospect theory may be useful.

2.2 Prospect Theory and Related Works

Prospect theory was developed by Daniel Kahneman and Amos Tversky [9], for which Kahneman won a Nobel Prize in economics in 2002. It examines possible outcomes in terms of gains and losses to determine the value of these outcomes [8]. When asked about the preferred choice between the “sure thing” and a risky option, it is assumed that the decision maker’s attitude to risk is revealed. If X is the monetary outcome, 0.5 is the probability of “winning” X and V is the value of a choice option, the preferences in Table 2 reveal the risk attitude of an individual.

Table 2. Risk Preferences

Scenario	Risk Attitude	Risk Preference
$V(X) = V(0.5 * 2X)$	Risk neutral	Indifferent between the options
$V(X) > V(0.5 * 2X)$	Risk adverse	Prefers the “sure thing”
$V(X) < V(0.5 * 2X)$	Risk taker	Prefers the risky option

While prospect theory is built on utility theory, prospect theory differs in that it considers issues from within a reference frame. Utility theory looks at net wealth, where prospect theory considers gains and losses. Thus prospect theory can explain why a loss of \$500 is felt more strongly than a gain of \$500.

Prospect theory has many practical applications, even within software engineering. It has been used to understand the reasons behind software project escalation [17], explain why early involvement in software project bidding leads to higher quotes [18], and support with the pricing of software bundling [19]. What is interesting about prospect theory is the relationship between the outcome and value is not linear, as seen in Figure 1. The function is steepest around zero and flattens out as the losses and gains become greater. It is also steeper for losses than gains.

The properties of the function mean that \$500 loss is felt more strongly than a \$500 gain as the relative value for losses is greater than the value for a similar sized gain. This can be seen in Figure 1 by using the guide markings.

Given the choice between (a) a 50% chance of gaining \$1000 or (b) a sure gain of \$500, most people select the safer option, B [9]. Figure 1 shows that a doubling in gains is not equivalent to a doubling in value, so the value delivered by option B on average is actually less than option A (as the value derived from the average of no gain and a \$1000 gain is less than the value of a \$500 gain).

Conversely when presented with options to (a) lose \$1000 with 50% probability or (b) a sure loss of \$500, most people chose the riskier option A [9]. As more value is lost from \$0 to \$500 than from \$500 to \$1000, the expected value of the outcome is less than the guaranteed loss of \$500.

Another implication of prospect theory is that preferences will depend on how a problem is framed [8]. If the decision maker perceives the outcome as a gain

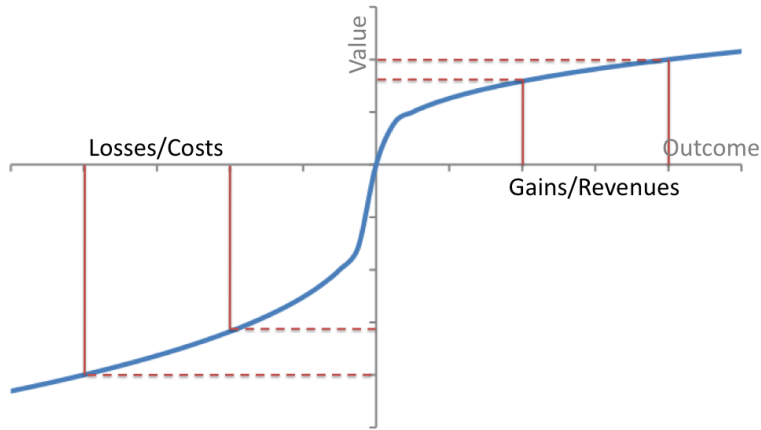


Fig. 1. Value function from prospect theory

they will accordingly act in a risk-averse manner, while acting in a risk-taking manner where the perceived outcome is a loss.

Framing has been found to play an important role in medicine. How the benefits of a vaccine are framed will yield very different take-up rates [11]. Patients who are told a vaccine is guaranteed to stop a virus are more likely to take it, than patients who are told a vaccine will stop two equally likely viruses with 50% probability.

However, these results would suggest that most people would not buy insurance. Replicas of the original experiments into prospect theory have found that most people act differently depending on whether the cost is framed as a gamble or an insurance [20,21]. When a cost scenario is framed as insurance most people are risk averse preferring to pay the smaller fixed amount, while when it is presented as a gamble people prefer to take on the risk.

Kahneman and Tversky [9] also found that people overweighed low probabilities and underweighted higher probabilities. This explains why people are prepared to buy lottery tickets.

2.3 Using Prospect Theory to Explain Requirements Selection Decisions

Given prospect theory can be applied to decision-making processes in disciplines from finance to medicine [22], this section proposes how prospect theory can be used to explain the requirements selection decisions in MDSPD – in this context losses are described in terms of costs and gains in terms of revenues.

The requirements types presented in Section 2.1 are represented in Table 3 with information on whether each is perceived as generating costs or revenues. Commercial requirement types are marked with an asterisk, while system requirement types are not.

Table 3. Requirement associations and related risk

Requirement Type	Cost/Revenue Driver
Customer specific features	* Revenue
Market pull requirements	* Revenue
Innovation	* Revenue
External Quality aspects	* Revenue
System Requirements (internal quality)	Cost

Commercial and system requirements are obviously very different in nature. Commercial requirements are clearly associated with customers – and therefore with revenue, while system requirements are associated with costs. This result can be seen through the language used to describe the different requirement types. Commercial requirements are referred to as *business opportunities*, and are discussed in terms revenue generation and expanding markets. In contrast system requirements are discussed as costs and must be justified with either, “If we do not do this now it will lead to higher costs in the future,” or, “If we make this change now we will be able to lower our production costs.”

As commercial and system requirements can be described in terms of gains and losses, prospect theory has an obvious connection in how these different requirement types are valued. This is seen most clearly by replacing gains and losses with revenue and costs, and placing requirements of different types on the graph in Figure 1.

Looking at a commercial requirement with an expected gain, one can see that a doubling of the gain does not double the value of the requirement. Similarly halving the gain does not halve the value of the requirement. The results from previous studies of prospect theory would therefore suggest that decision makers would be risk adverse when selecting commercial requirements, preferring guaranteed returns over larger riskier returns. Conversely decision makers would prefer to take more risk with system requirements, trying to minimize costs, even if these savings come at the risk of spending more money.

Different levels of risk are associated with the various requirement types. A customer specific requirement often has a customer willing to foot the entire cost, and one may reasonably assume that an individual customer is indicative of wider market demands. The return on investment of an innovation, however, does not have the same guaranteed revenue stream, but there is a potential for large revenues if the requirement proves popular. Internal system quality requirements are likewise risky as they involve investments in the longer-term sustainability of a software product. This could mean, for example, a requirement is developed allowing different database systems to be used with the product, but if this is never used the cost was unnecessarily incurred. It should also be noted that the customer is only aware of commercial requirements, with only the development organisation having clear visibility into internal system quality.

Given the clear link between prospect theory and MDSPD RE, the aim of the experiment presented in this paper is to investigate whether prospect theory holds in an MDSPD RE setting. While the authors acknowledge MDSPD RE is a complex decision making setting, the focus of this paper is on understanding the bias towards the perception of value as proposed by prospect theory, and how it may impact the decision making process.

3 Experiment Planning and Operation

This section presents the research question and major steps in the experiment planning and operation.

3.1 Research Question

The goal of the experiment presented in this paper is to investigate if the results of the experiments conducted by Tversky and Kahneman [9,10] can be seen in the MDSPD RE context - that is people are more risk taking when it comes to losses (costs) and risk adverse when it comes to gains (revenue).

In order to investigate this idea, the experiment follows the arrangement of the experiments conducted by Tversky and Kahneman [9,10], but places the participants in a MDSPD RE context. Thus the research question for the experiment is formulated as follows:

- When selecting requirements, do product managers show more risk-adverse behaviour with requirements formulated in terms of revenue, than with requirements formulated in terms of cost?

3.2 Experiment Instrumentation

A questionnaire was developed to investigate whether the participants behave in a manner that can be predicted by prospect theory. A range of risk levels and monetary values was used. The questions included are presented in Table 4. In order to control the variable for value gains and losses are only described in monetary terms. While providing a richer context would better simulate reality, it is not easy to assess what value participants place on the different aspects that make up this context.

Each set of questions is designed in the similar style used in the original experiments by Tversky and Kahneman [9,10]. Examples of the original questions can be found in Table 5.

From the questions presented in Table 4 and Table 5 it is easy to see the similarities between formulation of the questions from the authors' experiment and the original experiment. However, the question formulation differed between the original and authors' experiment with the original experiment using zero gain and zero loss as an option in each case. The effect of this difference in the design of the questions is discussed in Section 4.

Table 4. Question Sets for Revenue and Cost

Set	Revenue	Cost
1	Select one requirement: A has a guaranteed revenue of €10,000. B could raise €2,400 revenue with 75% probability, and €32,800 revenue with 25% probability.	Select one requirement: A has a cost of €10,000. B could cost €200 with 30% probability and €14,200 with 70% probability.
2	Select one requirement: A has a guaranteed revenue of €10,000. B could raise €2,000 or €18,000 revenue with equal probability.	Select one requirement: A has a cost of €10,000. B could cost €2,000 or €18,000 with equal probability.
3	Select one requirement: A has a guaranteed revenue of €10,000. B could raise €200 revenue with 30% probability, and €14,200 revenue with 70% probability.	Select one requirement: A has a cost of €10,000. B could cost €2,400 with 75% probability and €32,800 with 25% probability.

Table 5. Questions from Related Studies

Ref	Gain	Loss
[10]	Select one: <ul style="list-style-type: none"> – A sure gain of \$250. – A 25% chance to gain \$1000, and a 75% chance to gain nothing. 	Select one: <ul style="list-style-type: none"> – A sure loss of \$750. – A 75% chance to lose \$1000, and a 25% chance to lose nothing.
[9]	Scenario: In addition to whatever you own, you have been given \$1000. You are now asked to choose between the alternatives: <ul style="list-style-type: none"> – A 50% chance of gaining \$1000. – A sure gain of \$500. 	Scenario: In addition to whatever you own, you have been given \$2000. You are now asked to choose between alternatives: <ul style="list-style-type: none"> – A 50% chance of losing \$1000. – A sure loss of \$500.

A number of measures were undertaken to reduce systematic biases in the results. The order effect was addressed by systematically varying the order of the questions in the study. Additionally questions were placed between questions on revenue and cost to distract the participant from their previous answers.

A pilot of the questionnaire was completed prior to conducting the experiment. Two lecturers in software engineering completed the questionnaire, with both feeling they had a clear understanding of what was required, and were able to make informed decisions.

3.3 Experiment Subjects

The experiment involved 71 student participants completing either their final year of undergraduate studies or a masters in Software Engineering at Blekinge Institute of Technology, Sweden. The group consisted mostly of international masters students, who came from Pakistan, India, China, Iran, Nepal, Bangladesh, Nigeria, Germany and Jordan; with four undergraduate students, all Swedish.

Forty-nine of the participants had prior experience of working in the software industry. The Swedish students had experience from software engineering projects in terms of project courses which are run in tight cooperation with industry and very much resemble real development in industry. At the time of the experiment most of the students had completed the courses in Software Requirements Engineering and Software Verification and Validation, thus the students were assumed to be familiar with requirements engineering concepts.

3.4 Experiment Operation

The experiment was conducted at Blekinge Institute of Technology in the spring of 2008 during a single two-hour session.

Prior to starting the experiment the participants were introduced to the role of a software product manager and presented with the key performance indicators (KPIs) used to assess a person in this position. The list of KPIs included responsibility over the selection of requirements in a release to maximize the profit (return on investment) generated by the sales of a product. The KPIs remained visible for the duration of the experiment and the participants were encouraged to act in accordance to them.

4 Results and Analysis

This section presents the results, with an analysis and discussion.

4.1 Presentation of Results

The results of the experiment are presented in Table 6. The table shows the percentage of participants that chose the safe and risky options for the cost and revenue related questions in each of the question sets presented in Section 3.2.

Table 6. Participant responses (percentages)

Set	Revenue		Cost	
	Safe	Risk	Safe	Risk
1	0.69	0.31	0.49	0.51
2	0.68	0.32	0.68	0.32
3	0.56	0.44	0.52	0.48

As shown in Table 6, most participants were risk adverse when answering questions related to revenue, with most selecting the safer option. For the revenue questions presented in *Question Set 1*, 69% chose a safe option, in *Question Set 2* a similar 68% chose the safe option, while in *Question Set 3* 56% chose the safe option.

The results for the cost related questions show only a small difference in preference between the safe and risky options in *Question Set 1* and *Question Set 3*. However, in *Question Set 2* the participants were more risk adverse, which was the opposite of what was expected (68% safe).

Comparing the answers between the revenue and cost related questions, the results show increased risk taking attitude in cost related questions for *Question Set 1* (from 31% to 51%) and *Question Set 3* (from 44% to 48%). The attitude towards risk is unchanged for *Question Set 2*.

4.2 Analysis

The results for revenue related questions show a preference for avoiding risk in each and every case, demonstrating alignment with aligned with the original experiment conducted by Tversky and Kahneman [9]. While the results showed more risk-taking behavior in the questions related to cost, the results were not as strong as the original prospect theory experiments.

The level of risk avoidance is higher for the revenue questions in *Question Set 1* and *Question Set 3* than for the questions regarding cost in the same question sets. When it comes to cost related questions, two of the three tested cases (*Set 1* & *Set 3*) do not show a strong preference to take or avoid risk. These combined results, when considered relative to one another, indicate a more risk-adverse behaviour with requirements formulated in terms of revenue, than with requirements formulated in terms of cost.

The results for *Question Set 1* and *Question Set 3* show a clear change in attitudes towards risk between requirements termed as costs and requirements termed as revenue. While the result for *Question Set 2* did not show a strong difference, the majority of the cases support the application of prospect theory to software requirements selection.

4.3 Discussion

The results of the experiment indicate that decision makers will be more risk adverse when choosing between requirements formulated in terms of revenue, compared to when choosing between requirements formulated in terms of cost. This provides ground for concluding that prospect theory can be used to understand and explain decision making in MDSPD requirements selection situation.

The observed attitude towards risk taking for cost related questions was not as strong as expected, however, aspects of this may be explained by the experiment design. The participants of the experiment are students and do not have the same sense of ownership and responsibility towards the money of a real product manager. This could mean that they were not as sensitive to losing money. For example, looking at the cost related questions in *Question Set 2*, it is reasonable to assume that the students did not see the value of taking the risky option because it was not their own budget, own reputation or job position that was at stake. Another difference is that in original experiments in prospect theory, the subjects were offered to decrease the loss to zero, which may have motivated more to risk as other psychological factors are involved with zero [23].

The authors expect an experiment involving professional software product managers would be more aligned with the original experiments as people in this role have a greater sense of ownership in the product for which they are responsible and will face a greater loss in reputation for failing to meet budget than the experiment participants. This assumption is supported by findings of the study on application of prospect theory on software project decisions, where project managers' decisions were found to be aligned with prospect theory [11].

4.4 Validity Threats

Internal and external validity threats are most important to address for experiments in software engineering field [24] and social sciences [25].

Internal validity is concerned with identifying whether the observed result of the experiment is influenced by other factors than the experiment treatment. For the experiment presented in this paper an ordering effect (learning) and experiment instrumentation (the way questions are formulated) are the most significant threats.

Treatment order effect would mean the order the questions are presented will affect the participants' answers. To minimize this threat, the order of cost and revenue questions were systematically varied in the study.

To minimize the risk associated with the question formulations, a pilot of the experiment was conducted involving three participants. The intention of the pilot was to discover ambiguities and improve the formulation of the questions.

External validity is associated with the possibility to generalize the outcome of the experiment. While using students as experiment subjects usually puts a threat for generalizing the results [26], most of the students participating in the experiment have prior industrial experience and are trained in the requirements engineering field at a masters level. However, the participants may not have felt

the responsibility for the success of the product in the same way that a product manager in industry would. As discussed earlier in Section 4.3 this provides ground for assuming that an experiment involving professionals will be more aligned with the original experiments in prospect theory.

MDSPD RE operates in a much more complex setting than that modelled in this experiment, potentially impacting the generalisability of the result. For example, while it can be argued that software requirement selection decisions are more commonly group decisions, and not up to individuals as modelled in this experiment. However, the application of the results in a group situation should still be possible as research has shown that individuals' attitudes to risk are translated into a groups attitudes to risk [27]. Similarly, while requirements are more complex than questions of cost and revenue, in order to control the participants perceived gains and losses and how these impact value, the problems were reduced to monetary terms.

The results presented in this paper are inline with other findings from industry, recognising the importance commercial requirements with a fixed return over more variable market opportunities [6]. Similarly, literature recognises the risk-taking attitude towards system requirements [4,5,6,7].

5 Conclusions and Future Work

The intention of the research presented in this paper is to investigate if the ideas behind prospect theory, one of the prominent theories in decision making, can explain requirements selection decisions in market-driven software product development (MDSPD). The experiment presented in this paper replicates the design of the original experiments into prospect theory, but places it in the context of requirements selection situations in MDSPD.

The experiment results show a clear potential for applying prospect theory in MDSPD setting. The participants consistently displayed risk adverse behaviour when selecting between requirements described in terms of revenue. In two of the three cases investigated the participants were more risk taking when selecting between requirements described in terms of cost when compared to the revenue situation. The increase in risk taking attitude was not as distinct as anticipated. However, the authors' expect that an experiment involving professionals would show larger difference between risk taking approach between the revenue and cost related situations.

To the best of the authors' knowledge the work presented in this paper is the very first application of prospect theory in MDSPD requirements selection context. The insights found in this paper provide contributions both to researchers and practitioners in this field, as they open up possibilities for explaining the behaviour of decision makers in different requirements selection situations. This provides an answer to why recent studies observe internal quality requirements being consistently undervalued compared to commercial requirements, as well as help in finding ways for managing the balance between different types of commercial requirements originating from market pull and technology push.

Looking more closely at commercial requirements, prospect theory suggests preference favour for the guaranteed revenue stream over more uncertain options. Looking at the different requirements types, this would translate to a preference for customer specific features over innovation related requirements, as predicting market demands brings with it more risk.

The situation for system requirements is more complex. As these requirements are viewed as cost-drivers, they cannot be directly compared with the other types of requirements that are perceived as delivering revenue. This highlights the need to describe system requirements in terms of the gains that they deliver, so that a comparison between requirements of different types can be made. However, it should be noted that the risk associated with system requirements is still higher – so even when described in terms of gains, natural preference will be given to less riskier options. Putting this in MDSPD context implies that unless there is a clear strategy for balancing commercial and internal quality requirements, the later will be consistently down-prioritized until the point when the architecture issues will pose an impediment for the production of commercial features.

A number of actions should be undertaken as future work. A follow-up study is planned involving software product managers from companies working in a market-driven software development context, helping address issues faced in the experiment presented in this paper. Additionally this work should be used to educate software project managers to the biases they bring to the development process, and will be used as an input to a model to help software product managers balance requirement types when selecting requirements for a release.

References

1. Aurum, A., Wohlin, C.: The fundamental nature of requirements engineering activities as a decision-making process. *Information and Software Technology* **45**(14) (2003) 945–954 Eighth International Workshop on Requirements Engineering: Foundation for Software Quality.
2. Regnell, B., Brinkkemper, S.: Market-driven requirements engineering for software products. *Engineering and Managing Software Requirements* (2005) 287–308
3. Carlshamre, P.: Release planning in market-driven software product development: Provoking an understanding. *Requirements Engineering* **7**(3) (2002) 139–151
4. Wohlin, C., Aurum, A.: What is important when deciding to include a software requirement in a project or release? *International Symposium on Empirical Software Engineering* (November 2005) 246–255
5. Wohlin, C., Aurum, A.: Criteria for selecting software requirements to create product value: An industrial empirical study. *Value-Based Software Engineering* (2006) 179–200
6. Barney, S., Aurum, A., Wohlin, C.: A product management challenge: Creating software product value through requirements selection. *Journal of Systems Architecture* **54**(6) (2008) 576–593 Selection of best papers from the 32nd EUROMICRO Conference on ‘Software Engineering and Advanced Applications’ (SEAA 2006).
7. Barney, S., Hu, G., Aurum, A., Wohlin, C.: Creating software product value in china. *IEEE Software* **26**(4) (July–August 2009)

8. Plous, S.: *The Psychology of Judgement and Decision Making*. McGraw Hill (1993)
9. Kahneman, D., Tversky, A.: Prospect theory: An analysis of decision under risk. *Econometrica* **47**(2) (1979) 263–291
10. Tversky, A., Kahneman, D.: The framing of decisions and the psychology of choice. *Science* **211**(4481) (1981) 453–458
11. Lauer, T.W.: Software project managers' risk preferences. *Journal of Information Technology* **11**(4) (1996) 287–295
12. Karlsson, L., Regnell, B., Karlsson, J., Olsson, S.: Post-release analysis of requirements selection quality post-release analysis of requirements selection quality — an industrial case study. In: *9th International Workshop on Requirements Engineering — Foundation for Software Quality (RefsQ)*. (June 2003)
13. Regnell, B., Karlsson, L., Host, M.: An analytical model for requirements selection quality evaluation in product software development. *Proceedings of the 11th IEEE International Requirements Engineering Conference* (September 2003) 254–263
14. Ngo-The, A., Ruhe, G.: A systematic approach for solving the wicked problem of software release planning. *Soft Computing — A Fusion of Foundations, Methodologies and Applications* **12**(1) (2008) 95–108
15. Karlsson, L., Dahlstedt, A.G., Natt och Dag, J., Regnell, B., Person, A.: Challenges in market-driven requirements engineering — an industrial interview study. In: *8th International Workshop on Requirements Engineering — Foundation for Software Quality (RefsQ)*. (September 2002)
16. Gorschek, T., Wohlin, C.: Requirements abstraction model. *Requirements Engineering* **11**(1) (2006) 79–101
17. Keil, M., Mixon, R.: Understanding runaway it projects: Preliminary results from a program of research based on escalation theory. *Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences* **3** (January 1994) 469–478
18. Jorgensen, M., Carelius, G.J.: An empirical study of software project bidding. *IEEE Transactions on Software Engineering* **30**(12) (December 2004) 953–969
19. Chang, W.L., Yuan, S.T.: A markov-based collaborative pricing system for information goods bundling. *Expert Systems with Applications* **36**(2) (2009) 1660–1674
20. Hershey, J.C., Schoemaker, P.J.H.: Risk taking and problem context in the domain of losses: An expected utility analysis. *Journal of Risk and Insurance* **47**(1) (1980) 111–132
21. Slovic, P., Fischhoff, B., Lichtenstein, S.: Response mode, framing and information processing efforts in risk assessment. *New directions for methodology of social and behavioral science*. In: *Question framing and response consistency*. Jossey-Bass, San Francisco (1982)
22. Wu, G., Markle, A.B.: An empirical test of gain-loss separability in prospect theory. *Management Science* **54**(7) (2008) 1322–1335
23. Shampanier, K., Mazar, N., Ariely, D.: Zero as a special price: The true value of free products. *Marketing Science* **26**(6) (2007) 742–757
24. Wohlin, C., Höst, M., Runeson, P., Ohlsson, M.C., Regnell, B., Wesslén, A.: *Experimentation in Software Engineering: An Introduction*. Springer (2000)
25. Robson, C.: *Real World Research: A Resource for Social Scientists and Practitioner-researchers*. Blackwell Publishing (2002)
26. Berander, P.: Using students as subjects in requirements prioritization. *International Symposium on Empirical Software Engineering (ISESE)* (August 2004) 167–176
27. Farber, H.S., Katz, H.C.: Interest arbitration, outcomes, and the incentive to bargain. *Industrial and Labor Relations Review* **33**(1) (October 1979) 55–63