BEHAVIOURAL EFFECTS IN SOFTWARE DEVELOPMENT: AN EXPERIMENTAL INVESTIGATION

Research in Progress

Shmueli, Ofira, Ben-Gurion University of the Negev, Beer-Sheva, Israel, ofirash@post.bgu.ac.il
Pliskin, Nava, Ben-Gurion University of the Negev, Beer-Sheva, Israel, pliskinn@bgu.ac.il
Fink, Lior, Ben-Gurion University of the Negev, Beer-Sheva, Israel, finkl@bgu.ac.il

Abstract

Over-Requirement, manifested when a product or a service is specified beyond the actual needs of the customer or the market, is considered as a major risk in software development projects. This research empirically investigates whether Over-Requirement is partially due to emotional involvement of developers with software features they develop. Due to behavioural effects termed Endowment effect, I-Designed-it-Myself effect and IKEA effect, such involvement has been demonstrated when people come to overvalue physical items that they possess, self-design, or self-create. We conducted an experiment to explore these three behavioural effects in the context of a software development project. The 86 participants were randomly assigned to eight different experiment groups, according to whether they were responsible for (or not), specified (or not) and/or constructed (or not) a nice-to-have Over-Required software feature. The study's preliminary findings show that following these manipulations participants over-valued features that they were assigned to be responsible for, to specify or to construct, confirming our proposition that behavioural effects impact software development processes and influence Over-Requirement. The study is of relevance to theory about behavioural effects in software development and to practice via insights to Over-Requirement risk.

Keywords: Software Development, Over-Requirement, Endowment Effect, I-Designed-it-Myself Effect, IKEA Effect.

1 Introduction

The phenomenon of Over-Requirement, which involves product or service specification beyond the actual needs of the customer or the market, presents a real problem since Over-Required features compose at least 30% of project scope (Coman and Ronen, 2009). Known also as over-specification and gold-plating, Over-Requirement is hardly reversible since any feature, whether Over-Required or not, is rarely removed from project scope once introduced (Dominus, 2006).

Over-Requirement has been viewed for years as a major risk in software development projects (Ronen and Pass, 2008; Boehm and Papaccio, 1988). Boehm (1991) included Over-Requirement as one of the top 10 software development risks in the early 1990s which, despite the evolutionary progress in software development methods and techniques, has continued to be mentioned as a top risk ever since (Baccarini et al., 2004; Bernstein, 2012; Houston et al., 2001; Kaur et al., 2013; Khanfar et al., 2008; Malhotra et al., 2012; Ronen and Pass, 2013; Schmidt et al., 2001; Wheatcraft, 2011). The risk associated with Over-Requirement is related to varied damages as project overruns (Buschmann, 2009) and excessive complexity (Coman and Ronen, 2009, 2010). For example, resources wasted to
develop valueless extra functionality (Westfall, 2005) are not available to include valued requirements within project scope (Elliott, 2007), resulting in excessively complex software that suffers from low reliability (Coman and Ronen, 2010; Westfall, 2005) and low maintainability (Battles et al., 1996; Buschmann, 2010). From the customer's perspective, adding extra functions may paradoxically lead to less value (Rust et al., 2006). It also may result in reputation damage for the supplier when the system is not what the user wished for (Kautz, 2009).

Professional interest in advanced technology, developers' pride, and users' ambitious demands are considered to be the primary causes to the occurrence of Over-Requirement (Buschmann, 2009; Cule et al., 2000; Ropponen and Lyytinen, 2000; Schmidt et al., 2001), and are all related to the behaviour of actors in the software development project. Moreover, given the negative impacts of Over-Requirement, it is reasonable to attribute this phenomenon to irrational uneconomic behaviour. Thus, it makes sense to consider behavioural effects that behavioural economists have experimentally investigated mostly in contexts other than software development when looking for the roots of Over-Requirement.

The purpose of this work is to show that Over-Requirement is associated with three effects demonstrated by behavioural economists: (1) Endowment effect, i.e., the tendency of people to overvalue their possessions (Thaler, 1980); (2) I-Designed-it-Myself effect, i.e., the tendency of people to overvalue their self-designed products (Franke et al, 2010); and (3) IKEA effect, i.e., the tendency of people to overvalue their self-constructed products (Ariely and Jones, 2008). Although most research on behavioural effects concentrated on tangible objects, several researchers (e.g., Kahneman et al. (1990) and Hoorens et al. (1999)) demonstrated the Endowment effect with respect to intangible goods like property rights and working time. Moreover, Heyman et al. (2004), in an experiment that emulated Internet auctions, showed that simply imagining possession of an item while being the highest bidder resulted in over-bidding.

Thus, it is reasonable to investigate whether the three behavioural effects impact developers in the processes of developing intangible software products. We propose in this study that emotional involvement, which evolves upon assuming responsibility for a software feature, constructing it or specifying it, biases the software developer's ability to objectively evaluate the importance of that feature, whether Over-Required or not. We conducted an experiment to investigate these propositions, with a 2×2×2 factorial structure created by manipulating, in a fictitious software development project, the responsibility for an Over-Required feature, the construction of its pseudo-code lines and its actual specification.

This empirical study contributes to understanding the impact of behavioural effects on biased perception of feature importance by software developers and on sustained Over-Requirement. This understanding may serve practitioners as well as researchers. Practitioners should be aware that people, when engaged in development of a software feature, come to over-value it and are prone to cognitive biases due to the Endowment, I-Designed-it-Myself and IKEA effects. Researchers are served by this study since enhancing and demonstrating the linkage between concepts from the behavioural economics and software development areas opens the road for theory development in the intersection of both areas.

Section 2 presents next the theoretical background, reviewing the literature on the Endowment, I-Designed-it-Myself and IKEA effects, and hypotheses tested in the experiment we conducted. Then, Sections 3 and 4, respectively, present the details about the method and the preliminary results of the experiment. Finally, Section 5 concludes the paper with a discussion of its contributions, limitations and future research directions.
2 Theoretical Background and Hypotheses

2.1 The Endowment effect

The Endowment effect, derived from Prospect Theory (Kahneman and Tversky, 1979), is about overvaluation by people of objects that they own (Thaler, 1980) and implies that ownership positively influences the perceived value of the owned object (Kahneman et al., 1990, 1991). Ariely and Jones (2008) and Kahneman et al. (1991) have demonstrated in experiments a positive valuation difference between an owner's willingness-to-accept (WTA) for an object, as a mug or a candy bar, and a non-owner’s willingness-to-pay (WTP) for that object, which they attribute to the Endowment effect. While ownership duration was demonstrated to have a positive impact on the product's valuation and perceived attractiveness (Strahilevitz and Loewenstein, 1998), merely touching an object results in perceived ownership and an increment to valuation (Peck and Shu, 2009).

Given the Endowment effect, we hypothesize that software developers too might exhibit ownership feelings while assuming responsibility for a feature, whether that feature is necessary or merely nice-to-have, and attribute this behaviour to higher perceived value than otherwise. Our first hypothesis suggests that upon assuming responsibility for a software feature, including an Over-Required one, the psychological ownership feeling toward this feature positively impacts its perceived value. Thus,

\( H1: \) Assuming responsibility for a feature has a positive impact on one’s perceived value of the feature

2.2 The I-Designed-it-Myself effect

The I-Designed-it-Myself effect is about the psychological benefit gained by self-designing an object (Franke et al, 2010; Ulrich, 2009) and implies that self-designing a product positively influences its perceived value, leading the designer to overvalue that object. Franke et al. (2010) have demonstrated in a series of five experiments positive differences between a designer's WTP for a self-designed object and his/her WTP for an identical object designed by others. They attribute this value gap to the I-Designed-it-Myself effect after controlling for other possible explanations such as preference fit, sunk cost effect and mood effect, and asserting that the feeling of being the originator of the object is the primary explanation. Franke et al. also found that this effect is mediated by a person’s feeling of accomplishment and is moderated by the quality of the outcome and by the perceived contribution to the design process.

Given the I-Designed-it-Myself effect, we hypothesize that software developers too might get emotionally attached to their creation while specifying a feature, whether that feature is necessary or merely nice-to-have, and attribute this behaviour to higher perceived value than otherwise. Our second hypothesis suggests that, upon specifying a feature, including an Over-Required one, there is a positive impact on its perceived value. Thus,

\( H2: \) Specifying a feature has a positive impact on one’s perceived value of the feature

2.3 The IKEA effect

The IKEA effect is about the attachment feelings of people towards objects they self-create and implies that self-constructing a product positively influences its perceived value and leads the creator to overvalue that product (Ariely and Jones, 2008; Norton et al., 2012). Norton et al. (2012) have demonstrated in experiments positive differences in the WTP for a self-assembled object between the constructor and non-constructors and between the constructor's self-assembled object and an identical object assembled by others, which they attribute to the IKEA effect. Their experiments covered a
variety of object types and showed that this effect holds not only for hedonic objects (as origami or Lego models) but also for more practical ones (as IKEA boxes).

Given the IKEA effect, we hypothesize that software developers too might get emotionally attached to their creation while constructing a feature, whether that feature is necessary or merely nice-to-have, and attribute this behaviour to higher perceived value than otherwise. Our third hypothesis suggests that, upon constructing a feature, including an Over-Required one, there is a positive impact on its perceived value. Thus,

\( H3: Constructing a feature has a positive impact on one’s perceived value of the feature \)

3 Method

Figure 1 depicts our research model, reflecting the three above hypotheses. In the experiment conducted in this study to empirically test the hypotheses, using a 2×2×2 factorial design, the dependent variable \( \Delta \text{Rank} \) was measured, and three dichotomous independent variables were manipulated: Endowment, I-Designed-it-Myself, and IKEA. Advanced Engineering students in the Information Systems track at an Israeli university were asked to participate in the experiment.

\[
\begin{align*}
\text{Independent Variables} & \\
\text{Endowment} & \quad \text{Dependent Variable} \\
\text{I-Designed-it-Myself} & \quad \Delta \text{Rank} \\
\text{IKEA} & \\
\end{align*}
\]

\textbf{Figure 1. Research model}

The one-hour-long experiment involved five steps, during which participants were randomly assigned to eight groups, representing eight combinations of the three effects manipulation, each of which being manipulated (or not). At the beginning of each step, the first author informed participants how much time is dedicated to that step and instructed them to carefully read a written mission and then, as described next, either complete a paper-based questionnaire or perform some other task on paper. At the end of each step and prior to the next step, the completed papers were collected. The five steps included: background story and baseline ranking (Step 1), three manipulations (Step 2 through Step 4, one for each independent variable) and summary (Step 5).

Seven minutes were dedicated to Step 1 of the experiment, which began with a written presentation of a fictitious case about developing a software system for remote-banking clients. A list of nine functional features, deliberately composed to be diverse in terms of importance, was presented as well within the background story. To set their perceived value ranking baselines, participants were then asked, given the system's goal, to complete a questionnaire and rank the nine features in descending
importance toward meeting the system’s goal (from the most important in the first place to the least important in the last place).

Five minutes were dedicated to Step 2 of the experiment, which manipulated the Endowment effect. Participants were told in a written statement that they assume responsibility to further develop one of the nine features of the remote-banking system in future development phases of the project. Half of the participants, who were grouped under the Endowment condition (Endowment manipulation=Yes), were assigned the same feature: "Presenting my itemized expense report for a specific period" (hereinafter, "THE Feature"), deliberately chosen to be Over-Required given the system’s goal, while the others (Endowment manipulation=No) were assigned a different feature: "Presenting my account data - balance, savings, etc" (hereinafter, "PLACEBO Feature"). All participants were asked to think about their assigned feature and fill on paper a given space of two lines with their thoughts and comments about their feature.

Eighteen minutes were dedicated to Step 3 of the experiment, which manipulated the I-Designed-it-Myself effect and required participants to specify in detail the logic and screen layout of one of the nine features, given general instructions. Half of the participants, who were grouped under the I-Designed-it-Myself condition (I-Designed-it-Myself manipulation=Yes), were asked to specify THE Feature, while the others (I-Designed-it-Myself manipulation=No) were asked to specify the PLACEBO Feature.

Eighteen minutes were also dedicated to Step 4 of the experiment, which manipulated the IKEA effect. In this step, the first task of participants was to express, on a 1 to 7 Likert scale, about feelings (e.g., achievement and challenge) toward their previous task performed in Step 3. Then, they were given a list of scrambled pseudo-code lines for the algorithm behind the feature assigned to them, and were asked to construct the pseudo-code lines in the right order according to guidelines. Participants, who were grouped under the IKEA condition (IKEA manipulation=Yes), were required to construct THE Feature, while the others (IKEA manipulation=No) were all required to construct the PLACEBO Feature.

Five minutes were dedicated to Step 5 of the experiment, which involved filling up a questionnaire with three parts. First, participants were asked to express feelings, on a 1 to 7 Likert scale, toward the previous task performed in Step 4. Second, participants were informed about management intention to reduce project scope due to resource constraints and were asked to help management by re-ranking the nine features in descending importance order, as performed in the baseline task in Step 1. Third, they were asked to respond to demographic and background questions as gender and age.

The three independent variables (Figure 1) were manipulated in Steps 2, 3, and 4. In each of these steps, half of the participants experienced the Endowment, I-Designed-it-Myself, or IKEA effect towards THE Feature by respectively being responsible for, specifying, or constructing THE Feature, while the other half performed the same tasks for the PLACEBO Feature. Therefore, relating to each effect, differences across groups were only in the assigned feature. The Δrank dependent variable measured the change in perceived ranking of THE Feature before and after manipulation.

To avoid a sequence effect, half of the participants were exposed to the I-Designed-it-Myself manipulation in Step 3 and to the IKEA manipulation in Step 4, while the other half were exposed to the IKEA manipulation in Step 3 and to the I-Designed-it-Myself manipulation in Step 4. With no significant differences found between the two sequences, we ignore this dimension in data analysis.

4 Preliminary Results

Data collection yielded 86 participants (60% male; 40% female), most of whom were in their late 20s (37% aged 20 to 25; 63% aged 26 to 30). Table 1 depicts the number of participants in each of the
eight experiment conditions. Due to missing data in the case of two participants, 84 usable responses were analysed.

<table>
<thead>
<tr>
<th></th>
<th>IKEA=Yes</th>
<th>IKEA=No</th>
<th>IKEA=Yes</th>
<th>IKEA=No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endowment=Yes</td>
<td>12</td>
<td>10</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>Endowment=No</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>I-Designed-it-Myself = Yes</td>
<td>42</td>
<td>I-Designed-it-Myself = No</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>IKEA=Yes</td>
<td>44</td>
<td>IKEA=No</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Participants' distribution among the eight experiment conditions

Prior to manipulations, in Step 1 of the experiment, participants positioned THE Feature out of all the nine features in the seventh place of importance by the measure of the dependent variable (i.e., ranking mean of 5.79). Overall, participants who were exposed to at least one of the effects changed their ranking of THE Feature from the seventh (M=5.79) to the fourth place (i.e., ranking mean of 5.23), with a significant difference between before and after ranking (i.e., t(73)=2.855, p=0.006).

Table 2 shows the ranking means, before and after manipulating each effect, for participants who were exposed to the effect (effect=Yes; worked on THE Feature) and for those who were not (effect=No; worked on the PLACEBO), and the t-tests results for the differences between the before-after ranking means. Where effect=Yes, all three effects generated statistically significant ranking improvements for THE Feature, supporting hypotheses H1 through H3. Where effect=No, in contrast, none of the differences between the before-after measures in all groups were statistically significant.

<table>
<thead>
<tr>
<th>Effect</th>
<th>effect = Yes; THE Feature</th>
<th>effect = No; PLACEBO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Before</td>
<td>After</td>
</tr>
<tr>
<td>Endowment</td>
<td>6.23</td>
<td>5.48</td>
</tr>
<tr>
<td>I-Designed-it-Myself</td>
<td>5.86</td>
<td>4.98</td>
</tr>
<tr>
<td>IKEA</td>
<td>6.32</td>
<td>5.66</td>
</tr>
</tbody>
</table>

Table 2. Statistics for the means of before and after ranking of THE Feature

Given that a negative measure of \( \Delta \text{rank} \) represents an improvement in value assessment, the arithmetic sign of \( \Delta \text{rank} \) and its t-statistics were converted so that increased value assessments are positive.

Table 3 presents a comparison between mean \( \Delta \text{rank} \) for THE Feature and for the average of the means for the other eight features obtained by the same participants for each effect. These comparisons, which were all statistically significant, eliminate possible alternative explanations for the before-after improvements, such as a confounding effect of repeated measurement or a mood effect, in which value assessments increase as a result of a general enjoyment caused by performing the experiment tasks.

<table>
<thead>
<tr>
<th>Effect</th>
<th>THE Feature</th>
<th>Other eight features</th>
<th>t-test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endowment =Yes</td>
<td>0.75</td>
<td>-0.09</td>
<td>t(43)=2.661, p=0.011</td>
</tr>
<tr>
<td>I-Designed-it-Myself=Yes</td>
<td>0.88</td>
<td>-0.17</td>
<td>t(41)=2.484, p=0.017</td>
</tr>
<tr>
<td>IKEA =Yes</td>
<td>0.66</td>
<td>-0.12</td>
<td>t(43)=2.126, p=0.039</td>
</tr>
</tbody>
</table>

Table 3. Statistics for \( \Delta \text{rank} \) means
5 Discussion and Conclusion

This research in progress provides preliminary support for the role that the Endowment, I-designed-it-myself, and IKEA behavioural effects play in software development projects in general and, in particular, their impact on the Over-Requirement phenomenon. Participants who assumed responsibility for, specified, or constructed an Over-Required software feature came to overvalue it, advancing it from seventh to fourth place (out of nine). This increase in ranking, reported while participants were made aware of the need to reduce project scope, points to overall decreased willingness to exclude that feature from scope due to the behavioural effects. This research is thus about the linkage between behavioural effects and software development processes, which has not been studied enough yet.

The preliminary findings contribute to both practice and research. For practitioners, it is necessary to acknowledge that, similar to ownership feelings associated with overvaluation due to merely touching an object (Peck and Shu, 2009), a software feature might be overvalued not only upon specifying or constructing that feature but even by merely assuming responsibility for it. Thus, it is worthwhile for practitioners to be aware that software developers’ objective judgement regarding feature value might be impeded due to behavioural effects. Assigning developers to be responsible for, construct, or specify a software feature ties them emotionally to that feature and leads them to overvaluation of that feature at the expense of other features. Taking into consideration their recommendations while deciding about project scope, or while deciding about scope reduction, might thus result with Over-Requirement. For researchers, this study opens avenues for further holistic exploration of the linkage between behavioural effects and software development processes in which, obviously, humans are involved. Moreover, it is worthwhile for researchers to harness awareness of this linkage toward improvement of methodologies for software development.

The main limitation of this study, i.e., the absence of subjects who are experienced professionals, is mitigated by the fact that mature (mostly in their late 20s) undergraduate students (advanced enough to hold part-time jobs as professionals) participated in the study. Since they were a few months away from joining the target population of professional software developers, this may be considered as one of the circumstances in which student-based research can produce generalizable statements (Compeau et al., 2012). Yet, future research by us and by others should address this issue and include practitioners to enhance external validity. It is also noteworthy that our study pertains to the waterfall software development approach. Hence, whether its findings hold for agile approaches as well needs to be explored in further research. Future research may also explore the impact of the three behavioural effects explored here as well as others in relation to anomalies in software development projects, other than Over-Requirement, such as schedule underestimation.

The preliminary results presented above are the first step toward further exploration of the basic hypotheses addressed here via more tests. In addition, we plan to pursue further proposition and testing of other hypotheses about the interaction among the behavioural effects and the impact of several factors on the magnitude of behavioural effects. For example, Norton et al. (2012) observed that perceived difficulty of the construction task is an important factor and that the task at hand should be difficult enough but not too difficult for the IKEA effect to emerge. When the construction task is too difficult, attachment can be thwarted (Lowry et al., 2011) since an increase in perceived value emerges due to feelings of pride and competence that arise by the construction task (Mochon et al., 2012). Regarding the I-Designed-it-Myself effect, Franke et al. (2010) similarly found that feeling of perceived contribution, elevated by availability of more design freedom, positively impacts the perceived value of the self-designed product. Likewise, availability of more alternatives and more features during product design by a computerized toolkit were shown to lead to higher product value (Dellaert and Stremersch, 2005). Also, the level of perceived contribution was found to be one of the motivational factors for the voluntarily participation of software developers in open-source Linux
development processes (Hertel et al., 2003). In addition, the effort invested in the process seems to play a role regarding the magnitude of both the IKEA effect and I-Designed-it-Myself effect. While ease of a self-designed process may harm the feeling of being the originator and lead to a decrease in perceived value (Franke et al., 2010), much effort may actually increase perceived value as demonstrated when designing a gift (Moreau et al., 2011) or when results match one's preferences better (Franke and Schreier, 2010). We intend to explore the impact of these factors (subjective difficulty, subjective creative freedom feeling, enjoyment, and achievement feelings) on the perceived value of a feature by addressing the different feelings reported by participants during the experiment.

References


