Available online at www.sciencedirect.com

ScienceDirect

International Journal of Project Management 33 (2015) 380 – 394

International Journal of Project Management

www.elsevier.com/locate/ijproman

Explaining over-requirement in software development projects: An experimental investigation of behavioral effects

Ofira Shmueli, Nava Pliskin, Lior Fink *

Department of Industrial Engineering & Management, Ben-Gurion University of the Negev, Israel

Received 27 February 2014; received in revised form 22 June 2014; accepted 17 July 2014
Available online 1 August 2014

Abstract

One of the major risks associated with software development is related to the phenomenon of *over-requirement*. Also known as over-specification and gold-plating, over-requirement is manifested when a product or a service is specified beyond the actual needs of the customer or the market. In the software development context, we argue in this work that over-requirement is due partially to the emotional involvement of developers with the software features they specify. Similar involvement has been demonstrated for physical items as a result of the *endowment*, *IKEA*, and *I-designed-it-myself* behavioral effects, when people come to overvalue items they possess or self-create. To explore these behavioral effects and the interactions among them in the context of software development, we conducted an experiment in which over 200 participants were asked to specify a nice-to-have software feature. Our results confirm the existence of these behavioral effects in software development and their influence on over-requirement. The findings contribute to theory by explaining the over-requirement phenomenon and by providing insights into behavioral effects in the context of software development. Also practically relevant, the findings can alert managers of software projects to the over-requirement risk evoked by the behavioral effects explored in this study.
© 2014 Elsevier Ltd. APM and IPMA. All rights reserved.

Keywords: Software development; Over-requirement; Over-specification; Gold-plating; IKEA effect; I-designed-it-myself effect; Endowment effect; Demands–abilities fit; Experiment

1. Introduction

The purpose of this work is to show that the *over-requirement* phenomenon manifested in software development is due partially to the emotional involvement of software developers with their creations. Also known as over-specification and gold-plating, it occurs when a product or a service has been specified beyond the actual needs of the customer or the market (Boehm and Papaccio, 1988; Ronen and Pass, 2008). The over-requirement phenomenon is common in software projects and it is seldom reversible since it is very difficult, if not impossible, to eliminate from the overall project scope software features that were introduced during the requirement-engineering phase or later (Dominus, 2006; Wetherbe, 1991).

More than 20 years ago, over-requirement was already included in the top 10 software-development risks and was listed among the "don't do" warnings in that environment (Boehm, 1991; NASA, 1992). Despite the awareness of the dangers of over-requirement and the attention toward its prevention, it has repeatedly been mentioned as one of the top risks in software-development projects both at the beginning of the twenty-first century (Baccarini et al., 2004; Houston et al., 2001; Schmidt et al., 2001) and more recently (Belvedere et al., 2013; Bernstein, 2012; Kaur et al., 2013; Khanfar et al., 2008; Malhotra et al., 2012; Ronen and Pass, 2014; Wheatcraft, 2011). Table 1 lists, according to the literature, some of the major potential problems associated with over-requirement risks, from launch delays and software that is excessively complex to the demise of an entire company. The first two list items are problems that can arise when the allotted resources, such as time or budget, have been exceeded. The next three items are related to the complexity, reliability, and maintainability of the software. The next four list

* Corresponding author at: 1 Ben-Gurion Ave., Beer-Sheva 84105, Israel. Tel.: +972 8 6472224; fax: +972 8 6472958.
E-mail address: finkl@bgu.ac.il (L. Fink).

http://dx.doi.org/10.1016/j.ijproman.2014.07.003
0263-7863/00/© 2014 Elsevier Ltd. APM and IPMA. All rights reserved.

Table 1
Damage resulting from over-requirement, as reported in the literature.

Damage	References
Delayed project launch	Buschmann (2009); Coman and Ronen (2009, 2010)
Project overruns	Buschmann (2009)
Excessive complexity	Battles et al. (1996); Buschmann (2010); Coman and Ronen (2009, 2010)
Increased probability of defects and reliability problems	Coman and Ronen (2010); Westfall (2005)
Difficult to manage and costly to maintain systems	Battles et al. (1996); Buschmann (2010); Elliott (2007)
Devoting human and machine resources developing functionality of no real value	Elliott (2007); Westfall (2005)
Defocusing and distraction from real value requirements	Coman and Ronen (2009, 2010); Elliott (2007)
Cutting-off core features due to project time constraints	Coman and Ronen (2009, 2010)
Reduced user satisfaction	Kautz (2009); Rust et al. (2006)
Hurt reputation	Kautz (2009)
Loss of the entire company	Coman and Ronen (2009, 2010)

entries describe problems elicited by the waste of project resources on functionality with no value instead of on the core-business functionality. The last two items are some of the possible implications of over-requirement for the entire organization.

The causes of over-requirement seem to be rooted in human nature. The professional interest or pride of the software developers, on the one hand, and excessive user demands, on the other, appear to be the main forces driving the tendency to introduce over-required features into project scope (Cule et al., 2000; Ropponen and Lyytinen, 2000). Indeed, developers and users alike often seem to ignore business requirements, instead favoring advanced technology (Buschmann, 2009; Schmidt et al., 2001), in the process adding costly “bells and whistles” to system requirements (Markus and Keil, 1994). In some cases, they adopt the “optimizer approach” (Ronen and Pass, 2008) and wish to achieve the best possible solution (Rust et al., 2006; Westfall, 2005) or to add functionality aimed to fulfill potential future needs (Buschmann, 2010; Coman and Ronen, 2010). DeMarco and Lister (2003) describe a case in which politically adversarial stakeholders in a software project overloaded it with excessive functionality for political reasons. Regardless of the causes behind over-requirement in the software industry, once development has begun, it is almost impossible to eliminate software features incorporated during the project planning stages (Dominus, 2006; Wetherbe, 1991).

Arguing that behavioral effects may bias one’s ability to objectively evaluate software feature importance and necessity, this work aims to show that over-requirement is due partially to the emotional involvement of the developers with the features they specified. This emotional involvement seems to be associated with three effects that have been demonstrated by behavioral economists: (1) *Endowment effect*, i.e., the tendency of people to overvalue their possessions (Thaler, 1980), (2) *IKEA effect*, i.e., the tendency of people to overvalue their self-constructed products (Ariely and Jones, 2008), (3) *I-designed-it-myself effect*, i.e., the tendency of people to overvalue their self-designed products (Franke et al., 2010). A secondary goal of this work is to demonstrate these three behavioral effects in the context of software development. To realize these goals, we conducted an experiment in which the participants were asked to specify a nice-to-have software feature. This setup enabled us to operationalize the three behavioral effects by manipulating and measuring task characteristics.

This work makes three main contributions. First, it improves the understanding of over-requirement, which, despite being a common but highly undesirable and risky phenomenon, has barely been explored. This research empirically demonstrates behavioral explanations for the over-requirement phenomenon. Second, this work demonstrates the existence and the influence of three behavioral effects in the context of the development of intangible software products, a perspective not taken in previous research on the subject. Indeed, most studies of the above three behavioral effects (Franke et al., 2010; Kahneman et al., 1991; Knetsch and Sinden, 1984; Norton et al., 2012) focused on tangible products that were either self-owned (e.g., mugs, candy-bars, lottery-tickets), self-assembled (e.g., Lego, Origami), or self-designed (e.g., T-shirts, watches), while very few studies of the endowment effect have dealt with the perceived value ascribed to intangible products (Hoorens et al., 1999; Kahneman et al., 1990). Third, this work is innovative, in that the three behavioral effects are tested together, and the interactions among them are explored.

The theoretical background of our study and our research hypotheses are outlined in the next section, after which we describe the research methodology of our experiment in the third section. After presenting the results in the fourth section, the paper concludes with a discussion about the implications of this work for theory and practice.

2. Theoretical background and hypotheses

Similar to previous experiments conducted in non-software contexts (Franke et al., 2010; Kahneman et al., 1991; Norton et al., 2012), we investigate whether the endowment, IKEA, and I-designed-it-myself behavioral effects also occur upon specifying a software feature during software development. As such, we propose that like the owners in endowment experiments, self-designers in I-designed-it-myself experiments, and self-assemblers in IKEA experiments, software developers become emotionally attached to their software creations. Furthermore, that attachment probably applies whether the features they specify are necessary or merely nice-to-have. Once their own ideas and cognitive efforts have been invested in feature specifications, software developers probably develop feelings of attachment toward those features, thereby increasing the features’ perceived values in their minds. We argue that these changes in their feelings and value perceptions reduce the willingness of developers to exclude from the project scope features that they specified, a scenario with a strong potential to create over-requirement in the case of nice-to-have but unnecessary features.

The first hypothesis in this study refers to the overall influence of feature specification without attributing this influence to a

Download English Version:

https://daneshyari.com/en/article/275744

Download Persian Version:

https://daneshyari.com/article/275744

Daneshyari.com