International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

Developing software beyond customer needs and plans: an exploratory study of its forms and individual-level drivers

Mattia Bianchi, Giacomo Marzi, Lamberto Zollo & Andrea Patrucco

To cite this article: Mattia Bianchi, Giacomo Marzi, Lamberto Zollo & Andrea Patrucco (2019) Developing software beyond customer needs and plans: an exploratory study of its forms and individual-level drivers, International Journal of Production Research, 57:22, 7189-7208, DOI: 10.1080/00207543.2019.1581953

To link to this article: https://doi.org/10.1080/00207543.2019.1581953

International Journal of Production Research, 2019
Vol. 57, No. 22, 7189–7208, https://doi.org/10.1080/00207543.2019.1581953

Developing software beyond customer needs and plans: an exploratory study of its forms and individual-level drivers

Mattia Bianchi [a], Giacomo Marzi [b*], Lamberto Zollo [c] and Andrea Patrucco [d]

[a]*House of Innovation, Stockholm School of Economics, Stockholm, SE, Sweden;* [b]*Lincoln International Business School, University of Lincoln, Lincoln, UK;* [c]*Department of Economics and Management (DISEI), University of Florence, Florence, Italy;* [d]*Department of Project and Supply Chain Management, Penn State University, Monaca, PA, USA*

(*Received 10 December 2017; accepted 3 February 2019*)

Excessive software development is the tendency to develop new software above and beyond the requirements of the market and/or planned specifications. It is a widespread phenomenon involving both risks and flexibility advantages. As it represents a challenging dilemma for software developers, it is important to study its human origins. Drawing on the tripartite model of individual attitudes, this study investigates the influence of developers's cognitive (intuitive and rational thinking styles), affective (emotional attachment) and behavioural (reliance on past experiences) traits on two forms of excess, beyond needs and beyond plans. Using survey data on 307 software developers, this study shows that different manifestations of excess are associated with distinct traits of software developers. Emotional attachment drives beyond needs excess. A positive (negative) association is found between relying on past experiences and beyond needs excess (beyond plans excess). An intuitive cognitive style fosters the inclusion of extra features in the new product scope, whereas a rational style might lead to developing one-size-fits-all software that targets the needs of a broad user base. These findings contribute to research on the development of digital new products and production technologies by offering a comprehensive yet fine-grained picture of excessive software development's nature and drivers.

Keywords: excessive software development; cognitive style; software; overdesign; scope creep

1. Introduction

Excessive software development is the tendency to develop new software above and beyond what is required by the market and/or by planned specifications (Ronen and Pass 2008). This practice can take different forms, e.g. overloading new software with extra, unnecessary capabilities or adding unanticipated features in the course of the project, and so multiple names, e.g. overdesign, featuritis, scope creep and gold-plating (Bjarnason, Wnuk, and Regnell 2012). Attempting to clarify this terminological confusion, Shmueli and Ronen (2017) propose three distinct yet overlapping categories of excess in software development, depending on the dimension which they interfere with: beyond needs excess (BNE), defined as developing new software beyond the actual needs of the customer; beyond plans excess (BPE), defined as continuously changing the product scope beyond the initially planned specifications; and beyond resources excess, defined as setting a bigger scope than can be implemented with the available resources.

The present study provides an empirical investigation of the individual-level antecedents of the first two excess categories. We focus on BNE and BPE because they represent challenging dilemmas for developers. On one hand, scholars have repeatedly classified BNE and BPE as irrational pathologies plaguing decision making and contributing to project failures (Boehm 1991). Developing nice-to-have features that add little or no value to customers is a major waste of resources and a distraction from core requirements, which can lead to artificial product complexity, lower usability and higher lifecycle costs (Coman and Ronen 2010). Exceeding initial scoping decisions by modifying originally agreed specifications once the project is under way can cause instability and firefighting, i.e. the unplanned allocation of resources to address last-minute issues (Repenning 2001). Accepting a change typically opens the door to an uncontrolled sequence of additional ones, hence the 'creep' denomination of this phenomenon that can cause cost and schedule overruns (Bjarnason, Wnuk, and Regnell 2012).

However, BNE and BPE also represents intentional tactics to retain flexibility to manage the uncertainty inherent in developing something new (Sull 2004).[1] Market and technological conditions can shift unpredictably during and after projects, demanding prompt adaptation. Overdesigning a new product is an attempt to *buffer* uncertainty: adding safety

*Corresponding author. Email: gmarzi@lincoln.ac.uk

margins to software specifications and leaving multiple options open make the product more tolerant of emerging deviations, avoiding costly reworks (Loch, DeMeyer, and Pich 2011). The entrepreneurship literature, especially on lead users (Lettl, Hienerth, and Gemuenden 2008), provides examples of successful innovations that offered vanguard functionalities ahead of the market that eventually became mainstream (Fontela et al. 2006). Expanding the product scope beyond the original plans allows to *adapt* to uncertainty: an evolutionary design that incorporates newly emerged information results in solutions that achieve a better product-market fit (Bianchi, Marzi, and Guerini 2018). Welcoming changing requirements even late in development is a core principle of Agile, a software development approach associated with quality and productivity gains (Lee and Xia 2010).

Because they present both advantages and disadvantages, the decision to pursue development excess is a complex one.[2] Such decision is rooted in human nature and behaviour (Coman and Ronen 2010). While users might drive BNE and BPE by asking for the best possible solution or changing their priorities (Cule et al. 2000), Shmueli and Ronen (2017) argue that developers have the largest influence. However, evidence on this important influence is scarce and mostly anecdotal (Buschmann 2009). To identify relevant developers' attributes that might drive development excess, we draw on the tripartite model in psychology research according to which individual attitudes have cognitive, affective and behavioural components (Robinson et al. 1991). This study focuses on the influence of intuitive and rational cognitive styles (cognition), of emotional attachment (affection) and of reliance on past experiences (behaviour),[3] as they closely relate to the particular nature of development excess and strongly characterise developers (Shmueli, Pliskin, and Fink 2016). The research question addressed by this study is thus: what is the relation between cognitive, affective and behavioural traits of developers, and different forms of beyond needs and beyond plans excessive software development?

Differently from previous case-based and experimental works, this study employs a survey methodology on a dataset of 307 software developers from ten virtual communities. We adopt a large-N exploratory research design which is appropriate when existing theories provide useful frames for a baseline argument but are not sufficiently robust to formulate precise hypotheses (Bettis et al. 2014). Software is the industry where the development excess phenomenon was first investigated (Boehm 1991). Its prevalence in this context is mainly due to the information-based nature of the product, which promotes the misconception that adding features has low marginal cost, and the high volatility of development projects, which complicates decision making and supports the adoption of flexibility-enhancing practices (MacCormack, Verganti, and Iansiti 2001).[4] This study finds evidence of distinct manifestations of excess (one form of BNE that involves adding extra features to the new product scope; one form of BNE that involves targeting an extremely broad user base; and one form of BPE), which are differently associated to individual developers' attributes. These results contribute to research on the development of new products and of digital production technologies that make up the emerging Industry 4.0 (Liao et al. 2017), whose performance largely depends on the quality of its software components.

The next section reviews relevant literature. Section 3 develops the conceptual framework, Section 4 describes the methodology and Section 5 reports the findings. The final section discusses the contribution of this study, outlining its limitations and future research avenues.

2. Literature review

Since the 1990s there has been an extensive debate on the nature of excess in software development and on its performance effects. The software engineering literature has mainly warned about the risks involved in developing new products that exceed the needs of the market and in letting features creep into the project after the scope has been set (Boehm 1991). Research in innovation management and new product development, where software is one component, has to a larger extent highlighted the flexibility gains that can be obtained by leaving adequate margins in the product specifications and by modifying the plans if changes materialise during the project (Thomke and Reinertsen 1998). Particularly, the growing literature on Agile management is investigating the influence of evolutionary adaptations of product design on innovation speed, cost and quality (Lee and Xia 2010).

This study aims to contribute to research on the antecedents of BNE and BPE. Notwithstanding the importance of understanding their origins, evidence on what drives the adoption of development excess is mainly anecdotal (Coman and Ronen 2010). Conceptual articles and commentaries have speculatively offered a long list of environmental, organisational and individual causes. For BNE, these include incentive systems rewarding engineers for technological brilliance and marketers for the short-run new product sales; limited market knowledge by R&D staff and disregard of business goals; users exhibiting an all-or-nothing attitude and asking for as much as possible in the new product; developers' overconfidence, professional pride, desire to be at the technological frontier, search for peer-appreciation, adoption of market risk diversification and anticipation strategies (Cule et al. 2000; Rust, Thompson, and Hamilton 2006; Ronen and Pass 2008). For BPE, possible drivers are organisational politics and conflicts between stakeholders, lack of formal procedures to evaluate change requests, outsourcing partners that profit from implementing scope changes, early release of prototypes and integration of

the resulting user feedback and developers' intrinsic motivation to continuously improve (MacCormack 2001; DeMarco and Lister 2003; Buschmann 2009).

Empirical studies of BNE and BPE drivers are fewer. A recent, systematic literature analysis by Shmueli and Ronen (2017) identifies seven articles in peer-reviewed journals in the information systems, software engineering and project management areas. We expand this selection including studies in the marketing and innovation fields that investigate excess in the development of new products that include software components, and thus constitute relevant background to construct our conceptual framework. Table 1 provides an overview of these empirical studies. Each work is described in terms of the methodology used, the antecedents and the forms of excess investigated.

Most studies focus on single forms of either BNE or BPE. The studies on BPE investigate organisational and contextual drivers of this excess type. Schmidt, Lyytinen, and Keil (2001), Damian and Chisan (2006) and Chen, Law, and Yang (2009) find that scope creep, i.e. a form of BPE, is a function of poor planning, ineffective change management, the pressure to resolve conflicts between managers and developers, not clearly identifying core customer requirements and not understanding the true work effort and skills needed to complete a project. An in-depth case study of a software company by Bjarnason, Wnuk, and Regnell (2012) show that repeated requirements changes build up from multiple interacting drivers, such as the practice of producing detailed requirements upfront, the existence of communication gaps across teams and the lack of clear strategic goals. Lee-Kelley and Sankey (2008) and Choi and Bae (2009) highlight the role of contextual factors, such as project size, regulatory and technological volatility, in driving BPE. According to our literature review, no study has investigated the individual origins of BPE.

Studies on BNE have instead examined the influence of both organisational and individual drivers of this excess type. At the organisational level, Christensen and Bower (1996) show that the tendency to develop products that overshoot the needs of the mainstream market stems from the attempt by industry leaders to seek higher profit margins by offering feature-rich products to the most demanding customers, and is intensified by strong industry competition. Among the studies that focus on more micro-level drivers, Thomson, MacInnis, and Park (2005) emphasise the role of users in driving BNE, as they prefer to purchase feature-loaded offerings even when the extra features will make the product harder to use. Their preference derives from inferring functional benefits from concrete product attributes, even irrelevant ones, and from the social benefits provided by advanced products, e.g. eliciting others' positive impressions of the buyer's skills. Shmueli, Pliskin, and Fink (2015, 2016) focus instead on the individual traits of developers. They find that the more emotionally attached developers are to the new software creation, the greater the value that they will attribute to a feature they specify and the more likely superfluous features will be included in the product scope. Conversely, developers who use reference information about experiences in past projects and who adopt an external observer perspective tend to add less over-specified features. A common characteristic of these micro-level studies on BNE is that they are based on laboratory experiments with university students.

The only work that has simultaneously investigated forms of BNE and BPE, gold-plating and requirements creep respectively, is a survey study by Ropponen and Lyytinen (2000) which shows that development excess is higher in organisations that do not systematically use project and risk management methods and that develop software systems with distributed architectures. However, this study gives an overview of a broad range of software development risks and thus only marginally addresses each form of excess, whose measurement is in fact limited to a single survey item. Also, it does not examine the individual drivers of BNE and BPE.

As indicated in the last row of Table 1, the present study aims to advance understanding of the origins of development excess by offering a comprehensive yet fine-grained analysis of how multiple individual characteristics of software developers simultaneously relate to different forms of excess. Conceptually, we draw on Shmueli and Ronen (2017) who ascribe to developers the highest influence on the emergence of excess and we use the tripartite model of individual attitudes (Robinson et al. 1991) to identify traits that cover the whole spectrum of human responses: cognitive, affective and behavioural. Moreover, we take a balanced perspective to the role of excess in development: not only does excess represent a risk deriving from flawed decisions and human biases but could also stem from well-intentioned attempts to please customers and might offer flexibility advantages. Empirically, we use a large-N survey-based empirical design involving industry practitioners. This study thus offers an original contribution to existing literature which has typically examined single individual antecedents and/or single excess forms separately, mostly emphasised the risks attached to this practice, and primarily offered qualitative and experimental evidence.

3. Conceptual framework

Building on the scholarly works reviewed above, we develop a conceptual framework that can be empirically explored in this study. First, the framework elucidates the concepts of BNE and BPE so to bring clarity to the vague, sometimes inconsistent definitions of, e.g. gold-plating, over-requirement, scope creep and requirements changes in existing literature. Second, the

Table 1. Existing empirical studies on the drivers of BNE and BPE vis-à-vis the present study.

Study	Methodology (empirical setting)	Drivers of excess	Forms of excess
Christensen and Bower (1996)	Multiple case study (Disk drive industry)	Focus on organisational and contextual drivers • Strategy of targeting higher tiers of the market • High level of competition	Focus on a form of BNE • Needs overshooting: excessive capabilities offered by the new product
Thomson, MacInnis, and Park (2005)	Laboratory experiment with students (Digital media players)	Focus on individual, user-related driver • Users' attraction to feature-rich products	Focus on a form of BNE • Feature fatigue: inclusion of extra, unimportant features
Shmueli, Pliskin, and Fink (2015)	Laboratory experiment with students (Software industry)	Focus on individual, developer-related driver, and on contextual factors • Involvement in specifying a feature • Difficulty of the development task	Focus on a form of BNE • Over-requirement: inclusion of extra features of no real value
Shmueli, Pliskin, and Fink (2016)	Laboratory experiment with students (Software industry)	Focus on individual, developer-related drivers • Lack of reference information about similar past projects • Internal view approach, typical of engineers actively developing the software (as opposed to an external view, typical of consultants)	Focus on a form of BNE • Over-requirement: higher perceived valuation of features that leads to the inclusion of over-specified features
Schmidt, Lyytinen, and Keil (2001)	Delphy survey with expert project managers (Software industry)	Focus on organisational drivers • Lack of understanding of the actual development effort (size, complexity, skills) • Ineffective change management • Vague definition of the scope	Focus on a form of BPE • Scope creep: continuing stream of requirements changes
Damian and Chisan (2006)	Single case study (Software industry)	Focus on organisational drivers • Informal requirements management processes • Lack of project tracking	Focus on a form of BPE • Requirements creep: unconstrained late requirements changes
Lee-Kelley and Sankey (2008)	Multiple case study (IT projects in the bank industry)	Focus on organisational and contextual drivers • Regulatory changes • Changes dictated by external stakeholders • Use of emails to communicate changes	Focus on a form of BPE • Requirements creep: numerous requirement changes during development
Chen, Law, and Yang (2009)	Single case study (Automation industry)	Focus on organisational drivers • Tensions between top management and software developers, and resulting pressure • Ill-defined scope planning and definition	Focus on a form of BPE • Scope creep: requirement changes during development)

(*Continued*).

Table 1. Continued.

Study	Methodology (empirical setting)	Drivers of excess	Forms of excess
Choi and Bae (2009)	Simulation (Software in military industry)	Focus on contextual drivers • Volatile technical conditions • Volatile organisational conditions • Project size and long-term orientation	Focus on a form of BPE • Creeping requirements: uncontrolled changes to requirements
Bjarnason, Wnuk, and Regnell (2012)	Single case study (Software industry)	Focus on organisational and contextual drivers • Early detailing of requirements • Long project lead times • Communication gaps between technical areas • Unclear strategic vision and goals	Focus on a form of BPE • Scope creep: frequent late scope changes
Ropponen and Lyytinen (2000)	Survey (Software industry)	Focus on organisational and contextual drivers • No application project management, development and risk management methods • Inattention to ill-defined parts of the project • Distributed product architecture • Less interactive batch oriented software	Simultaneous investigation of BNE and BPE, although measured in an aggregate way, with a single survey item for each type of excess • Gold-plating: inclusion of unnecessary features (BNE) • Requirement changes: continuous, uncontrolled changes in specifications (BPE)
The present study	Survey (Software industry)	Focus on multiple developer-related drivers, according to the tripartite model of individual attitudes • Intuitive and rational cognitive styles • Emotional attachment • Reliance on past experiences	Simultaneous investigation of different forms of BNE and BPE, measured through multiple survey items

framework identifies relevant developers' characteristics that might relate to BNE and BPE. For a wide spectrum of personal traits, we draw on the tripartite model of individual attitudes in social psychology, which posits that an attitude, defined as the predisposition towards an object (here, software development and related decisions), consists of three classes of components (Shaver 1987; Robinson et al. 1991). The cognitive component refers to the beliefs and thoughts the developer has about the attitude object. The affective component concerns the feelings perceived by the developer towards the object. The behavioural component refers to his/her tendency to behave in a given way in relation to the object. Within each class, we use prior literature to identify traits that strongly characterise software developers and are closely linked to the nature of the decisions and activities involved in the emergence of excess (Baron 2007). For cognition, we focus on the fundamental thinking styles, intuitive and rational, that a laboratory experiment by Eliëns et al. (2018) has investigated as drivers of escalation of commitment, a BPE-related phenomenon. The also experimental works by Shmueli, Pliskin, and Fink (2015, 2016) inspired us to examine the role of emotional attachment, for affection, and of reliance on past project experiences, as behaviour.

In this study, we offer arguments to support the existence of associations between these developers' traits and excess forms. Although the arguments might differ, we expect the influence of individual drivers to be the same independently from whether the excess is relative to customer needs or plans. This follows the early-stage exploratory nature of this study and is consistent with Shmueli and Ronen (2017) who identify overlapping causes for BNE and BPE.

3.1. Beyond needs excess

BNE can manifest in multiple, overlapping ways. It can relate to the decisions of which features to include in the scope of the software system, of how many and how capable the included features are (Shmueli, Pliskin, and Fink 2015). Some features are core because they are currently required by the market and thus their inclusion in the scope is necessary to add value to users and so for the product's commercial success. Other features might be instead optional and not necessary toward meeting customer needs. Coman and Ronen (2010) argue that at least 30% of developed features in software projects are superfluous and report surprising examples, such as a hidden pinball application in Microsoft Word. Features, whether must-haves or nice-to-haves, could be over-specified, offering a higher performance level than customers require and are willing to pay for. Christensen and Raynor (2003) described the race to increase a processor's clock rate between Intel and AMD, which ended up providing more speed than their mainstream customers could use.

BNE might also manifest when the new product is specified very comprehensively so that it caters for the current needs of a broad user base (Rust, Thompson, and Hamilton 2006). Instead of offering an assortment of products tailored to each customer segment, market heterogeneity is addressed through a multi-purpose, one-size-fits-all system targeting any potential customer and in so doing, likely overshooting the needs of some segments. Thomson, MacInnis, and Park (2005) mention the case of a firm that when developing a calculator with financial analysis functions, added a set of features useful to biochemists in order to hit two birds with one stone. The excess could also materialise when developers attempt to accommodate expected, but not yet real, future market trends. The product scope is thus specified inclusively to meet demand whatever dynamics will affect the market during and after the development project. YouTube and Facebook are multi-functional platforms that run experiments trying to anticipate emerging communication and socialisation user needs (Nadkarni and Hofmann 2012).

3.2. Beyond plans excess

While BNE can occur in both upstream and downstream phases of the development process, i.e. when defining the specifications of the new system as well as during their implementation and testing, BPE occurs in the latter phases only, as it concerns the modification of initial scoping decisions through the addition and/or alteration of product features (Bjarnason, Wnuk, and Regnell 2012). Late changes might occur unexpectedly and take the form of reworks, where developers must iterate back to earlier phases and correct or redo activities (Karlström and Runeson 2006). Such changes might be uncontrolled as a flood of new or modified features creep into scope expanding the size of the project. This phenomenon occurred to the FBI's virtual case file project which aimed to replace the bureau's paper files with a networked system for tracking criminal cases (Keil and Mähring 2010). Alternatively, downstream changes might be the intended result of adopting flexible approaches to software development that advocate dynamic scope management and iterative replanning of specifications so to accommodate unforeseen contingencies (Iansiti 1995). By keeping the product scope fluid for a longer proportion of the development process, final decisions can be taken as close to market introduction as possible. This was a critical success factor in the development of Microsoft Internet Explorer, whose feature freeze occurred only a week before product release (MacCormack 2001).

3.3. Cognitive styles and software development excess

Dual processing theory posits that human cognition, i.e. the mental processes by which information is organised and processed, consists of two distinct elements that shape decision making: rationality and intuition (Kahneman 2003). Persons with a strong inclination towards a rational cognitive style tend to use a structured, analytical and logical approach to problem solving. Persons with a pronounced intuitive cognitive style solve problems in a holistic, experiential and automatic way and make decisions based on gut-feeling. Rationality and intuition are not opposing but co-exist in human beings: any person can be placed along a continuum ranging from highly rational to highly intuitive (Kirton 2003).

An intuitive cognitive style could be argued to drive excess. Development excess has been classified by different scholars as an erroneous, risk-laden form of decision making that may derive from human biases (Shmueli and Ronen 2017). Compared to analytical thinkers, intuitive decision-makers are particularly susceptible to cognitive biases and adopt heuristics (Bazerman and Moore 2013), that in complex and uncertain situations such as when developing new software lead them to take inaccurate decisions (Tversky and Kahneman 1974), e.g. overvaluing the importance of a feature and/or of changing requirements. In their analysis of escalation of commitment in innovation projects, Eliëns et al. (2018) show that intuitively inclined developers are more likely to fall victim of this pathology related to scope creep.

However, BNE and BPE have also been presented as tactics to retain flexibility. Adding just-in-case features, creating performance reserves or offering generic systems that are suitable for many customer segments might allow to readily respond to market and technological shifts without costly redesigns (Thomke and Reinertsen 1998). Changing planned

specifications might ensure a better product-market fit in volatile environments. To the extent that excess can provide performance gains, rational decision makers who analytically and logically process each information piece should be more prone to deliberately pursue BNE and BPE.

3.4. Emotional attachment and software development excess

As software development requires an extensive cognitive effort and time investment from developers, they tend to become emotionally involved with their new creations. They might describe the products they are working on as 'their babies' (Gino and Pisano 2008) and see them as part of their extended selves (Belk and Coon 1993). Shmueli, Pliskin, and Fink (2015) show that emotional attachment is stronger when developers have specified and designed the new product themselves. Also, the more difficult the development tasks are, the stronger the psychological-ownership feelings towards the software being developed. When this happens, they tend to perceive both critical and nice-to-have features as being more valuable than they actually are. This distorted evaluation impairs their ability to exclude superfluous features from the scope and thus increases the likelihood of BNE. Moreover, a strong affection for the software creation might be a key force driving the developer's desire to provide the best possible solution to customers, which is reported to cause both BNE and BPE, and to continuously improve the developed software through frequent changes that eventually lead to BPE (Shmueli and Ronen 2017).

3.5. Reliance on past experiences and software development excess

Using experience gained in past projects, both successful or unsuccessful, as reference information to make decisions about current initiatives is a relevant behaviour for software developers. This practice can help mitigate the planning fallacy according to which decision makers make overly optimistic evaluations of the value of the product under development (Lovallo and Kahneman 2003). Through this mechanism, harnessing knowledge about previous cases when scoping the current one reduces the inclusion of over-required features (Shmueli, Pliskin, and Fink 2016). Conversely, ignoring past lessons and focus on the unique characteristics of the current case, treating it as one-of-a-kind, might lead to BNE and BPE. In line with this finding, Coman and Ronen (2010) argue that excess more likely occurs when developers are inexperienced, as they strive to satisfy too many customer needs with their new products. Expert individuals instead develop effective 'rules of thumb' over time to better assess the value of new products, preventing the emergence of excess. These arguments would suggest a negative association of reliance on past experiences with BNE and BPE.

However, this behaviour *per se* might not be useful or could actually be detrimental. This can occur when developers reference the case at hand to previous projects that are poor approximations of it. Basing current decisions on information about these dissimilar occurrences can lead to negative knowledge transfer, which occurs when experience accumulated in a given activity is inappropriately transferred to a new activity that appears to be similar, but is, in fact, very different (Hoang and Rothaermel 2010). Related to this phenomenon is the use by software developers of the anchoring and adjustment heuristic (Aranda and Easterbrook 2005), wherein an early and possibly ill-advised estimate skews subsequent estimates (Tversky and Kahneman 1974). When developers make evaluations about, e.g. the value of certain features or of late scope changes, these evaluations might be significantly driven by initial anchors, such as past project outcomes. If these events are poorly comparable to the current one, inaccurate initial estimates lead to inaccurate final estimates, even after considerable adjustments (Gino and Pisano 2008). On the basis of this reasoning, one might expect reliance on past experiences to be positively associated to BNE and BPE.

The conceptual framework in Figure 1 graphically depicts the relations articulated in this Section, between the cognitive, affective and behavioural traits of developers, and different forms of excess.

4. Methods

4.1. Sample and data collection

Software development, where excessive practices are particularly diffused, is the empirical setting of this exploratory study. Delimiting the study to a single product type helps isolate the association between individual-level drivers and excess. As a consequence of digitisation and technology convergence trends, software is a key element in many new products also in non-IT industries (Karlsson, Larsson, and Öhrwall Rönnbäck 2018), and in the emerging suite of advanced production technologies that are part of Industry 4.0 (Liao et al. 2017). This study's unit of analysis is the development process that is employed to execute multiple projects. Because we do not measure project-level factors, we cannot account for heterogeneity between different projects, which could influence the relationships investigated in this study (Jayaram, Oke, and Prajogo 2014). Although this represents a limitation, other scholarly works on new product development (Ettlie and Elsenbach 2007) and software engineering (Tripp and Armstrong 2016) adopt this approach.

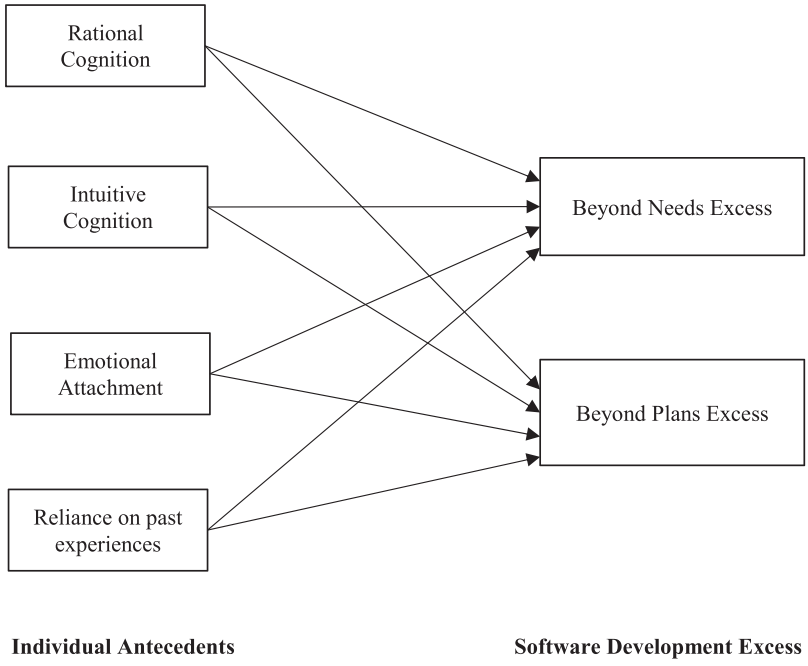

Figure 1. Conceptual Framework.

This study surveyed developers of internet software products who are members of five Italian virtual communities (Italia JavaScript, ASP.NET Italia, Google Development Group Slack Milan, Google Development Group Florence, WordPress Italia) and five international communities (Gnome, Grav Developers, WordPress International Slack, Gimp Developers, Joomla Core Code Developers). A web-based questionnaire was used to collect data in Spring 2017. Following the suggestions by existing literature to reduce non-response, we carefully designed the survey instrument, established the research significance, and balanced survey length (Rogelberg and Stanton 2007). In the first page of questionnaire, we informed participants that the questions refer to their actual, general software development work. They were thus asked to answer with respect of the process they employed to execute development projects instead of confining their responses to a specific project (Misra, Kumar, and Kumar 2010). Community administrators encouraged their members to participate (Steenkamp and Geyskens 2006) posting the link to the questionnaire on different channels, e.g. the community's forum, chat and Facebook group.

With this support, 479 questionnaires were collected corresponding to a raw response rate of 7.5%.[5] Online virtual communities have many inactive members which explains the typically low rates in the studies that use them (Chow and Cao 2008; Petrovčič, Petrič, and Manfreda 2016). After removing responses with missing values, we obtained a final nested sample of 307 software developers.[6] Table 2 reports relevant descriptive statistics of the sample. Non-response bias was checked through independent sample t-test on questionnaire variables (Dalecki, Whitehead, and Blomquist 1993), e.g. age, organisation size, cognitive styles, which indicated no statistically significant differences between early and late respondents.

4.2. Scale development

Multiple-item, 7-point Likert scales measures, ranging from 'strongly disagree' to 'strongly agree', are used for the main constructs of this study (Jarvis, MacKenzie, and Podsakoff 2003). The items assessing rational and intuitive cognitive styles are based on the short version of the Rational Experiential Inventory, a validated scale measuring both rationality and intuition (Hodgkinson and Sadler-Smith 2003). The scale measuring reliance on past experiences leans on the work by Lovallo and Sibony (2010), later adapted to a new product development environment by Belvedere, Grando, and Ronen (2013).

Scale development was necessary for the emotional attachment, BNE and BPE constructs due to a lack of existing scales. The measures in prior experimental research (Shmueli, Pliskin, and Fink 2015) were not applicable to a survey-based empirical design. Due to this study's aggregate level of analysis, we could not use objective indicators of excess,

Table 2. Sample descriptives.

	Number of respondents	Percentage of respondents
Age		
18–30	113	36.8
31–45	149	48.5
> 45	45	14.7
Gender		
Woman	13	4.2
Man	294	95.8
Main market		
B2C	92	30
B2B	215	70
Level of responsibility		
Managerial role	132	43
Non-managerial role	175	57
Number of members in a typical project team		
< 5	196	63.8
5–10	89	29
> 10	22	7.2
Number of employees in the organisation		
< 10	141	45.9
10–100	92	29.9
101–1000	37	12.1
> 1000	37	12.1
Location		
Italy	191	62.2
Europe (not Italy)	71	23.1
North America	27	8.8
Asia	15	4.9
Other	3	1

such as the number of superflous features in a product. The aim of measuring multiple dimensions of excess required to go beyond early aggregate attempts (Ropponen and Lyytinen 2000).

Jaworski and Kohli (1993) iterative procedure was used to develop the scales. A comprehensive literature analysis and multiple discussions with software professionals served to preliminarily form a large pool of items for the constructs. By applying the criteria of uniqueness and ability to convey different meanings (Churchill 1979), we selected a subset of items which were tested for clarity and appropriateness with the previously interviewed practitioners. Their feedback allowed to exclude, alter and add items. We reverse-score some items to reduce response set bias.

We then asked three academic experts to critically assess the scales in terms of domain representativeness, item specificity, and construction clarity. This pre-test returned meticulous suggestions for item revision. Finally, a pilot test was run with Master students in computer- and management-related disciplines who answered the questionnaire. After addressing the minor concerns raised by students, we had enough confidence that the instrument was reliable and valid for full-scale survey administration.

Table 3 reports the items included in the questionnaire that are relevant to this study. Items measuring BNE capture its multiple aspects, such as including superfluous features in the new software scope, providing performance levels that are better than required by the market, targeting all possible needs of the entire user base and accommodating expected, but not yet real, market trends, and the time in which they can occur, during the early specification phase and during later execution phases (Coman and Ronen 2010; Bianchi, Marzi, and Guerini 2018). Items measuring BPE assess the extent of late changes to planned specifications, their creeping nature often associated to the adoption in volatile environments of linear Stage-Gate models that advocate early scope freezing (MacCormack 2001; Bianchi, Marzi, and Guerini 2018) as well the use of dynamic scope management and iterative replanning techniques (Bjarnason, Wnuk, and Regnell 2012). The emotional attachment construct consists of five items assessing the strength of the developer's feelings towards his/her creations, which also depends on his/her task autonomy. As for the constructs derived from earlier works, the items for rational and intuitive cognition are four and five, respectively, and those for reliance on past experiences are three.

Table 3. Items included in the questionnaire.

First order constructs	Description	Code
Beyond Needs Excess	Including nice-to-have, non-core features in the scope of the new software product is a good way to differentiate from competing products and attract new users.	BNE1
	During the development of new software, innovative features can be added even if not strictly necessary to users.	BNE2
	To fully satisfy a user, it is important to offer something 'extra' beyond their expectations or needs.	BNE3
	When specifying the requirements for the new software, it is a good practice to include features that anticipate possible long-term future user needs, technical evolutions, and product extensions.	BNE4
	When the following release is too distant in the future, it is good practice to incorporate new features in the current development project, even if unanticipated.	BNE5
	When specifying the requirements for the new software, it is important to cater to the needs of the most advanced users ('power-users').	BNE6
	To be on the safe side and leave all options open, it is preferable to specify new software requirements as broadly and inclusively as possible.	BNE7
	One key goal in creating new software is to please as many users as possible.	BNE8
Beyond Plans Excess	Changing software requirements after the project specifications have been already 'frozen' should be absolutely avoided. (reverse)	BPE1
	The software's scope should be dynamic and change during the course of the project to accommodate new events and unforeseen contingencies.	BPE2
	Whenever a feature is added later in the project, it creates a distraction from the development of the core features of the new software. (reverse)	BPE3
	Features added later in the project during software development have typically lower value to users than those included in the initial specifications. (reverse)	BPE4
Rational Cognition	I double-check my information sources to be sure I have the right facts before making decisions.	RAT1
	I make decisions in a logical and systematic way.	RAT2
	My decision making requires careful thought.	RAT3
	When making a decision, I consider various options in terms of a specific goal.	RAT4
Intuitive Cognition	When I make decisions, I tend to rely on my intuition.	INT1
	When I make a decision, it is more important for me to feel the decision is right than to have a rational reason for it.	INT2
	I generally make decisions that feel right to me.	INT3
	When making decisions, I rely upon my instincts.	INT4
	When I make a decision, I trust my inner feelings and reactions.	INT5
Emotional Attachment	I perceive coding as a personal challenge.	EMO1
	Developing software is similar to creating art.	EMO2
	I often feel a strong link with my software creation, as if it is 'my baby.'	EMO3
	When developing new software, I enjoy great autonomy.	EMO4
	I feel delighted when I see my code not only working well, but also aesthetically pleasing.	EMO5
Reliance on past experiences	When I am planning and/or executing a new software development project, I recall examples of previous development projects.	PAS1
	When I am planning and/or executing a new software development project, I tend to replicate practices and decision criteria already utilised in the past.	PAS2
	I tend to ignore past experiences because each software development project is unique (reverse)	PAS3

4.3. Exploratory factor analysis

Consistently with the early stage of extant research on development excess and the exploratory nature of this study, we run an Exploratory Factor Analysis (EFA) to identify parsimonious, mutually exclusive and unitary constructs. We use principal component extraction with varimax rotation (Hair et al. 2006), with indictors on factors loading and reliability tests reported in Table 4. Only the factors with eigenvalue higher than 1 were retained. Items were retained if the corresponding factor loading exceeded the value of 0.5 only for a single factor and was below this threshold for the others (Hu and Bentler 1999).

The distribution of items and their factor loadings provide important insights. The predicted factor variables for Rational cognition, Intuitive cognition, Emotional attachment and Reliance on past experiences emerge from the scale items. Due to low correlation with the other items on the scale, two items (INT3 and EMO4) were removed from the corresponding constructs. As regards excessive software development constructs, the factor analysis supports the unidimensionality of the

Table 4. Exploratory factor analysis.

	Feature-Driven BNE	One-Size-Fits-All BNE	BPE	Intuitive cognition	Rational cognition	Emotional attachment	Reliance on past experiences
BNE4	0.684						
BNE1	0.661						
BNE2	0.647						
BNE6	0.576						
BNE5	0.569						
BNE8		0.7831					
BNE7		0.707					
BPE1			0.767				
BPE3			0.742				
BPE4			0.613				
INT4				0.816			
INT1				0.776			
INT5				0.754			
INT2				0.547			
RAZ2					0.747		
RAZ3					0.719		
RAZ4					0.717		
RAZ1					0.694		
EMO2						0.729	
EMO3						0.718	
EMO1						0.665	
EMO5						0.620	
PAS1							0.766
PAS2							0.713
PAS3							0.612
Cronbach	*0.602*	*0.671*	*0.695*	*0.666*	*0.704*	*0.675*	*0.731*
Item – total correlation	*0.522–0.559*	*0.591–0.631*	*0.601–0.666*	*0.588–0.637*	*0.622–0.698*	*0.597–0.652*	*0.662–0.843*

three-item BPE factor (the item BPE2 was dropped due to low correlation with the other items), while it results in a two-factor solution for BNE. The five-item scale *Feature-Driven BNE (FD-BNE)* captures manifestations of excess that relate to the inclusion of extra features that are not necessary towards meeting current core customer needs. The two-item scale *One-Size-Fits-All BNE (OSFA-BNE)* assesses the forms of excess that relate to developing a generic new product so that is suitable for any potential customer segment. This factor structure is coherent with the conceptual development of BNE in Section 3. One item (BNE3) was dropped due to the low correlation with other items on the scale.

A reliability test based on Cronbach alpha statistic and corrected item-total correlation coefficients was used to test the validity of the factors. In exploratory research, value of Cronbach alpha > 0.60 are considered acceptable (whereas in more advanced stages of research, values between 0.70 and 0.90 can be regarded as satisfactory; Nunnally and Bernstein 1994), while the threshold for item – total correlation is 0.5. Therefore, our results confirm a satisfactory level of reliability of the factors.

4.4. Model testing

The relations in the conceptual framework were statistically explored using Covariance-based Structural Equation modelling (CB-SEM), which is a common method used also for survey-based exploratory research (Perols, Zimmermann, and Kortmann 2013). The model was tested using the maximum likelihood (ML) estimation method (Hair, Ringle, and Sarstedt 2011), as ML compared to other methods – like Generalised Least Squares and Weighted Least Squares – is able to provide more realistic indexes of overall fit and less biased parameter values for paths that overlap with the true model (Olsson et al. 2000). ML estimation assumes that the variables in the model are (conditionally) multivariate normal, which is true for our dataset according to the Doornik – Hansen test ($\chi^2 = 948.708; p > \chi^2 = 0.000$) and the Henze – Zirkler test ($\chi^2 = 9945.008; p > \chi^2 = 0.000$).

The research model was analysed and interpreted sequentially in two stages: first, the assessment of the reliability and validity of the measurement model and second, the assessment of the structural model (Anderson and Gerbing 1988). Stata

14.0 was used to estimate both the measurement model and the structural model. The ML algorithm was used to obtain the paths, the loadings, the weights and the quality criteria.

The model fit was evaluated using the chi-square goodness-of-fit statistic and the use of other absolute or relative fit indices (Bagozzi and Yi 1988). For the former, some authors suggest checking for the ratio between the chi-square value and the degrees of freedom in the model, where the cut-off values range from < 3 (for testing purposes) to < 5 (for exploratory research). For the latter, to evaluate the fit of a model, several fit indices were used to supplement the chi-square. The fit indices range from 0 to 1, with values closer to 1, indicating a good fit. Hu and Bentler (1999) recommend MLE-based fit indices, and they suggest a three-index presentation strategy that includes the comparative fit index (CFI), The Tucker–Lewis index (TLI), and the root mean square error of approximation (RMSEA). A satisfactory threshold for CFI and TLI is > 0.90 (with a value > 0.95 showing excellent fit), whereas RMSEA is supposed to be lower than 0.05.

5. Results

5.1. Measurement model

The initial model – including factors coming from the EFA – was found to be discredited, as Composite reliability and Average Variance extracted for *FD-BNE, Emotional attachment and Reliance on past experiences*, as well as the model goodness-of-fit indicators, could not be considered satisfactory even for an exploratory stage. For this reason, three further items were eliminated (BNE6, EMO5 and PAS3), without compromising the theoretical validity of the three constructs.[7]

Table 5 shows the results of confirmatory factor analysis (CFA) for the new model. The fit indicators were now found to be satisfactory ($\chi^2 = 292.425$; χ^2/d.f. $= 1.730$; CFI $= .967$; TLI $= .951$; RMSEA $= .045$; CD $= .999$), as well as the factors reliability, as measured by the Composite Reliability (CR, Fornell and Larcker 1981). Additionally, convergent validity was assessed through significant loadings from all scale items on the hypothesised constructs, and through the Average Variance Extracted (AVE, Anderson and Gerbing 1988): AVE ranges between 37% and 57%.

Table 6 reports the summary statistics of the constructs and the correlation matrix. The correlations have been analysed using Pearson's *r* bivariate correlations. As an additional test for discriminant validity, we compared the squared correlation between two latent constructs to their AVE estimates (Fornell and Larcker 1981). According to this test, the AVE for each construct should be higher than the squared correlation between each pair of constructs. This condition is valid for all the constructs.

As we relied on a single respondent design, we controlled for common method bias in two ways, through the study procedure and through statistical control (MacKenzie and Podsakoff 2012). Regarding the procedure, the research project was labelled as a comprehensive overview of software development practices, therefore no explicit reference to development excess was made. Thus, respondents' attention was not drawn to the relations being targeted in this study. Survey questions related to the dependent and independent constructs were separated (Podsakoff et al. 2003), to prevent respondents from developing their own theories about possible cause–effect relationships. The questionnaire was carefully created and pretested, and the cover letter emphasized that participation was anonymous, no compensation would be provided, answers were neither right nor wrong, and the researchers would ensure maximum confidentiality and independence. Regarding statistical control, we performed the common latent factor technique (MacKenzie and Podsakoff 2012), used in other SEM studies (Luzzini et al. 2015; Gopal and Thakkar 2016), and we found that the common latent variable has a linear estimate of .661. This value, when squared, indicates a variance of .437 which is below the threshold of .50. This suggests that data analysis is not excessively affected by common method bias.

5.2. Structural model

A structural model was built to assess the path coefficients existing between the individual drivers and the constructs for excessive software development. Besides this study's main variables, the model includes the following controls: developer's age, organisation size, typical project team size, measured as categorical variables; a set of dummies that equal 1 if the developer is male, has a leadership role (functional manager, project manager, team leader), the main clients for his/her software are other businesses (B2B) and operate in high uncertainty markets (computer, telecommunications, aerospace, electronics, health) (Furr and Dyer 2014). We also include dummies to control for the virtual community to which the developer belongs, which helps address the nested nature of our data, and their location.

The postulated path model produced a sufficient fit to the data ($\chi^2 = 318.301.$; χ^2/d.f. $= 1.798$; RMSEA $= .048$; CFI $= .954$; TLI $= .945$; CD $= .999$). Table 7 and Figure 2 shows the corresponding empirical results.

The empirical results show that *FD-BNE* has a significant positive association with a developer's intuitive cognition ($p < .05$), emotional attachment ($p < .001$) and reliance on past experiences ($p < .001$), while no significance is found for rational cognition. The other form of BNE, *OSFA-BNE,* has also a significant positive association with emotional attachment

Table 5. Confirmatory factor analysis (CFA) of the constructs.

Construct	Factor Loadings	AVE[a]	CR[b]
Rational cognition		40.86%	0.733
RAZ2	0.692		
RAZ3	0.668		
RAZ4	0.597		
RAZ1	0.594		
Intuitive cognition		57.10%	0.799
INT4	0.824		
INT1	0.727		
INT5	0.711		
Emotional Attachment		37.62%	0.643
EMO1	0.583		
EMO2	0.604		
EMO3	0.651		
Reliance on past experiences		53.29%	0.691
PAS1	0.830		
PAS2	0.614		
FD-BNE		37.43%	0.705
BNE5	0.584		
BNE2	0.651		
BNE4	0.596		
BNE1	0.614		
OSFA-BNE		40.36%	0.575
BNE8	0.615		
BNE7	0.655		
BPE		36.43%	0.629
BPE3	0.601		
BPE1	0.689		
BPE4	0.507		

[a]Average Variance Explained.
[b]Composite Reliability.

Table 6. Correlation matrix and descriptive statistics.

	Mean	S.D.	1	2	3	4	5	6	7
1. Intuitive cognition	4.33	1.60	1						
2. Rational cognition	5.60	1.15	0.0084*	1					
3. Emotional Attachment	5.52	1.43	0.005	0.0046	1				
4. Reliance on past experiences	5.41	1.26	0.0171	0.004	0.0002	1			
5. FD-BNE	4.21	1.55	0.0095*	0.0018	0.0193*	0.0024	1		
6. OSFA-BNE	4.08	1.76	0.0014	0.0097	0.0199**	0.0142	0.0273**	1	
7. BPE	4.34	1.71	− 0.0033	0.0039	− 0.0319	0.0068*	0.0172	0.031	1

**$p < 0.01$; *$p < 0.05$.

($p < .01$) and reliance on past experiences ($p < .001$), however, as regards cognition, it has a significant positive association with rational cognition ($p < .01$) and not with intuitive cognition.

The data indicate that *BPE* has a significant negative association with the behavioural component, reliance on past experiences ($p < .05$) while no significance is found for the cognitive and affective components.

Some controls have significant association with the different forms of excess: *FD-BNE* is positively associated with being located in Europe and Asia, and negatively associated with belonging to Italia JavaScript and ASP.NET Italia communities; *OSFA-BNE* is positively associated with organisation size and being located in Europe; *BPE* is positively associated with gender, being located in Europe and belonging to Google Development Group Slack Milan.

Table 7. Parameter estimates.

	Coeff	Std error	*p* value
Feature-Driven BNE (FD-BNE)			
← Intuitive cognition	0.153	0.075	0.045*
← Rational cognition	0.085	0.074	ns
← Emotional Attachment	0.391	0.089	0.000***
← Reliance on past experiences	0.447	0.092	0.000***
One-Size-Fits-All BNE (OSFA-BNE)			
← Intuitive cognition	− 0.012	0.086	ns
← Rational cognition	0.262	0.093	0.003**
← Emotional Attachment	0.307	0.105	0.003**
← Reliance on past experiences	0.317	0.104	0.001***
BPE			
← Intuitive cognition	0.002	0.088	ns
← Rational cognition	0.024	0.087	ns
← Emotional Attachment	0.002	0.102	ns
← Reliance on past experiences	− 0.184	0.092	0.048*

**p < 0.05; **p < 0.01; ***p < 0.001 (two tailed test).*

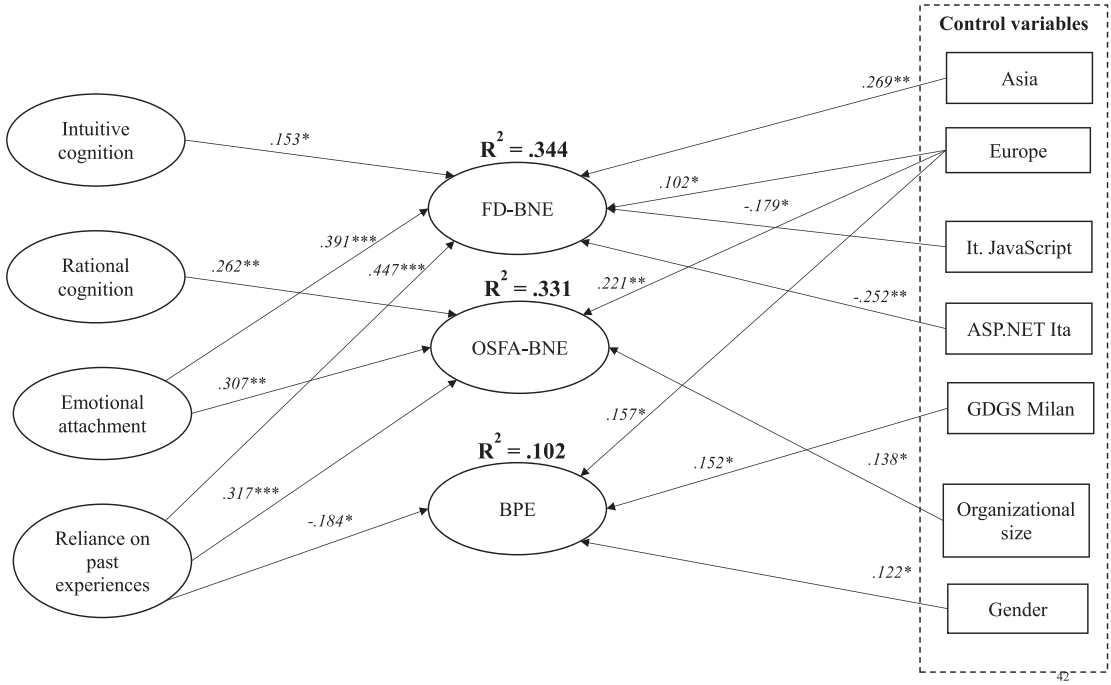

Figure 2. Structural Equation Model results.

6. Discussion and Conclusions

6.1. Research implications

This exploratory study has implications for research on innovation management and on new product development, for which software represents an increasingly important component. Furthermore, it holds relevant insights for current production research. The design and management of production technologies, such as Computer-Integrated Manufacturing and intelligent automation systems, have long depended on the development of the associated control software, on its quality and reliability (Chiu, Chu, and Sculli 2001; Kurihara et al. 2002). Software-based enterprise information systems for managing resources and processes in production and logistics, such as ERP, are key subject areas in the operations field (Maqueira, Moyano-Fuentes, and Bruque 2018). It is however evident that the centrality of software in production research will grow as a consequence of the exponential progress of computer technology and the shift to an information economy. The successful development of digital manufacturing technologies such as additive manufacturing and industrial robotics is critically

enabled by ongoing software innovations in the areas of, e.g. machine learning, cloud computing and computer-aided design (Khorram Niaki and Nonino 2017; Gunasekaran et al. 2018).

By creating novel measures that capture diverse facets of excess, this study adds to the existing blurred definitions and operationalizations of excess in various literature streams. Our factor analysis not only supports, as expected, a distinction between beyond needs and beyond plans forms of excess (Shmueli and Ronen 2017), but it suggests that each form might manifest in distinct ways as in the case of beyond needs excess. Our results help discriminate between a manifestation of BNE that emphasises the inclusion of non-required features in the new product scope and one that highlights the developers' attempt to target several customer segments at once and to attract an extremely broad user base. This distinction offers preliminary empirical support to the prediction advanced in conceptual articles (Coman and Ronen 2010) about the multidimensional nature of overdesign. Due to this study' limitations (see Section 6.3), our proposed twofold conceptualisation of BNE needs to be further validated by future research.

Given its focus on human drivers of excess, the core contribution of this study is to show that different manifestations of excess are associated with distinct traits of software developers and that a particular trait might exert a mixed influence depending on the type of excess. Based on our empirical analysis, we propose a set of formal propositions on these relationships for the direction of future research. As regards beyond needs excess, both feature-driven and one-size-fits-all forms appear to be influenced by the same affective and behavioural components, emotional attachment and reliance on past experiences. The former finding is consistent with the experiment by Shmueli, Pliskin, and Fink (2015) who find that developers who are emotionally involved with their creations are more likely to overdesign them. Hence, we propose that:

Proposition 1. Emotional attachment is positively associated to different forms of beyond needs excess.

Interestingly, our analysis indicates that *FD-BNE* and *OSFA-BNE*, while related, have different cognitive origins: an intuitive decision maker is more likely to adopt the former type while the latter type is more likely to emerge from a rational decision maker. One possible yet cautious interpretation of this result is that the act of including extra features not currently required by customers (*FD-BNE*) stems from optimistic and somewhat biased interpretations and predictions of what users might need, and so might be associated with more intuitive and instinctive judgment. One might argue that a feature orientation tends to characterise earlier stages of the development process which consists of explorative activities and more intuitive information processing (Leonard, Beauvais, and Scholl 2005; De Visser et al. 2014). On the other hand, the attempt to satisfy as many users as possible (*OSFA-BNE*) might be a conscious and deliberate tactic by a rational decision maker, who aim to maximise sales volume, at least in the short run (Thomson, MacInnis, and Park 2005). The desire to target a broad market could be argued to characterise later stages of the innovation process, where some uncertainty sources have been addressed and the higher availability of more reliable information allows for its analytical processing. Based on the analysis of our survey data, we thus posit that the association of each cognitive style with beyond needs excess is contingent on the specific form of this excess. The following propositions, which could be further elaborated and tested by future research, are formulated:

Proposition 2a. An intuitive cognitive style is positively associated to the form of beyond needs excess that consists of including extra features in the new product scope.

Proposition 2b. A rational cognitive style is positively associated to the form of beyond needs excess that consists of specifying a new product comprehensively so to satisfy as many users as possible.

Differently from BNE having both cognitive, affective and behavioural origins, BPE appears to associate only to the behavioural component of a developer's attitudes. The occurrence of late changes of plans, either as manifestation of scope creep or of an agile management approach, seems thus independent from the way an individual thinks and feels. An interesting result is that the behaviour of relying on experience gained in past projects to make decisions about current ones is found to be positively related to (both forms of) BNE and negatively related to BPE. The possibility of both negative and positive associations does not go against conceptual predictions in Section 3.5.

A tentative interpretation of this finding builds on the different nature of BNE vs. BPE. The former involves proactive anticipation, i.e. developers predicting the areas in which changes might occur and adding just-in-case buffers that will help manage eventual deviations, whereas the latter is about adapting on-the-fly by making reactive changes to the originally scope. In highly dynamic and uncertain environments such as software development, relying on past experiences might reduce the accuracy of estimations because knowledge of specific technologies and markets atrophy fast (MacCormack 2001). Indeed, Coman and Ronen (2010) report that in turbulent contexts, the number of design solutions that can be reused into next generation products is fewer and fewer. Projects are thus more likely to be one-of-a-kind. Because it involves anticipation and prediction, BNE is particularly exposed to anchoring bias and negative knowledge transfer which derive from overrelying on accumulated knowledge in past events that appear to be similar to the current one, but in reality are not, and so they lead developers to exceed what customers want from their products (Markovitch et al. 2015). Conversely,

the fact that BPE is emergent, reactive and characterised by a shorter lag between the decision to exceed and the resulting feedback might help developers better assess the similarity between past and current events and so the usefulness of prior knowledge as reference information for the case at hand. With this more accurate assessment, developers harnessing past experience might reduce the occurrence of late scope changes in the current project. Naturally, an in-depth analysis of what type of prior information developers rely on to make current decisions, and how they use it, is needed to assess the validity of this interpretation. Given the differences between BNE and BPE, we formulate two distinct propositions for their relationship with reliance on past experiences:

Proposition 3a. Reliance on past experiences is positively associated to different forms of beyond needs excess.

Proposition 3b. Reliance on past experiences is negatively associated to beyond plans excess.

All in all, while the conceptual work by Shmueli and Ronen (2017) suggests the existence of mutual, overlapping causes for different excess forms, this study finds substantial diversity. We argue that different human origins are motivated by the distinct nature of excess categories.

6.2. Managerial implications

While this study does not investigate the performance consequences of an 'above and beyond' approach to software development, it provides insights to project managers and developers on the individual traits that might lead to various forms of excess. Depending on whether managers are seeking to avoid excess because of its inherent risks or to pursue excess for flexibility advantages, this suggests possible remedies or catalysts for excess. First, managers aiming to go beyond customer needs should trigger strong feelings by developers toward their new software creations, e.g. by giving them more autonomy and ambitious challenges. Second, if managers' goal is to develop innovations that cater for the needs of a broad user base, they should form development teams with predominantly rational decision makers. They should instead involve intuitive thinkers when the purpose is to design feature-rich products. Third, managers can reduce scope creep by promoting the use of knowledge from past experiences as a basis for decision making in current projects but they should be aware that this can lead to overshooting customer needs.

6.3. Limitations and directions for future research

This study has limitations that call for future research work. Subjective and common method biases may exist due to the single-respondent retrospective approach used in this empirical analysis. A key limitation is that this study's unit of analysis is the aggregate software development process. Therefore, it does not measure project-level dimensions, e.g. size and complexity, and it cannot assess whether these affect the relationships here investigated. Future research should examine the impact of contextual factors on the association between individual drivers and excess forms. Because survey participants belong to different communities, sample data are nested. Unfortunately, the sample size does not allow to explore, separately, how the relationships vary across communities. A larger sample size and higher response rate would provide more robust statistical calculation and analysis and would aid in determining the generalizability of the findings. Focusing on fewer communities and using comparative methodologies (e.g. multi-group or multi-level SEM) to explore more macro-level effects on excess is an avenue for further research.

The study relies on novel, perceptual questionnaire items for measuring development excess. Despite having used statistical tests to assess their validity and reliability, they have limitations and need further verification. In-depth case studies and longitudinal data from multiple informants would strongly contribute to this. In hindsight, the survey instrument could have been improved by including more and more focused questions on the different types of excess, especially the beyond plans category. It is thus advisable to treat the research results accordingly.

Our empirical setting is the software industry. The information-based nature of the good and the high levels of volatility create peculiar conditions for the emergence of excess (MacCormack, Verganti, and Iansiti 2001), which may make our results not universally applicable in all kinds of environments. However, they should be to some extent transferable to highly turbulent environments, such as electronics, telecommunications and automation. Moreover, trends like digitisation are not only increasing the share of software components in new products and production technologies, but they are also favoring the adoption of software development practices in hardware-based contexts (Conforto et al. 2014). Future research should test the relationships investigated in this study, in other industries beyond software.

Finally, this study includes only a limited set of individual developer-related antecedents. Future research should investigate other drivers of excess, related to the developer (e.g. intrinsic motivation), to the user (e.g. risk propensity), to the

product (e.g. modular architecture) and to the organisation (e.g. incentive systems). It should also examine the interaction between different forms of excess, their effects on multiple performance dimensions and how they are moderated by management approaches.

Notes

1. Another tactic that offers development flexibility is the use of a modular architecture for the new product (Peng and Mu 2018). Modular designs enhance the ability to accept changes and additional features in the course of a project and throughout the product lifecycle. We thank an anonymous reviewer for this comment. While potentially a limitation, the role of a modular design for the new software is not included in the scope of this study.
2. Another tactic that offers development flexibility is the use of a modular architecture for the new product (Peng and Mu 2018). Modular designs enhance the ability to accept changes and additional features in the course of a project and throughout the product lifecycle. We thank an anonymous reviewer for this comment. While potentially a limitation, the role of a modular design for the new software is not included in the scope of this study.
3. A person's cognitive style refers to his/her preferred way of organizing and processing information to reach a decision (Messick 1976). A rational style involves slow, conscious, and analytical processing of information, while an intuitive style is fast, unconscious and instinctive (Kahneman 2003). Emotional attachment is defined as an emotion-laden target-specific bond between a person and a specific object (Thompson, Hamilton, and Rust 2005). Reliance on past experiences is here defined as using knowledge gained in previous projects as reference information to make decisions about the current initiative (Shmueli, Pliskin, and Fink 2015).
4. Software development projects are naturally characterized by heterogeneous levels of uncertainty and volatility. Because the unit of analysis in this study is the aggregate development process, it does not measure project-level factors. Although this is a limitation, which is common to other studies (Tripp and Armstrong 2016), Furr and Dyer (2014) rank the software industry as a whole in the top three for technical and market uncertainty.
5. A first round of data collection targeted four communities, Italia JavaScript (152 of the total 870 members responded to the questionnaire, a raw response rate of 17.4%), Google Development Group Slack Milan (27 of the total 83 members responded, a raw response rate of 32.5%), Google Development Group Florence (40 of the total 219 members responded, a raw response rate of 18.3 %), ASP.NET Italia (30 of the total 552 members responded, a raw response rate of 5.4%). This dataset was used for the empirical study by Bianchi, Marzi, and Guerini (2018), which addresses a different research question and draws on different concepts and theories than the present study. A second round of data collection shortly after targeted the remaining six communities: WordPress Italia (53 of the total 209 members responded to the questionnaire, a raw response rate of 25.4%), Gnome (45 of the total 649 members responded to the questionnaire, a raw response rate of 6.9%), Grav Developers (42 of the total 2096 members responded to the questionnaire, a raw response rate of 2%), WordPress International Slack (39 of the total 1574 members responded to the questionnaire, a raw response rate of 2.5%), Gimp Developers (33 of the total 92 members responded to the questionnaire, a raw response rate of 35.9%), Joomla Core Code Developers (18 of the total 54 members responded to the questionnaire, a raw response rate of 33.3%). These additional data are not used in previous publications.
6. The final sample consists of: 117 members of Italia JavaScript; 21 members of Google Development Group Slack Milan; 26 members of Google Development Group Florence; 17 members of ASP.NET Italia; 24 members of WordPress Italia; 28 members of Gnome; 27 members of Grav Developers; 23 members of WordPress International Slack; 7 members of Gimp Developers; 17 members of Joomla Core Code Developers. Because the sample size is low compared to the large population of Internet software developers and because individual developer effects can be nested in other higher-level, e.g., community-level, effects, results should be interpreted with caution.
7. The elimination of item BNE6 from factor *FD-BNE* should actually strengthen the validity of this factor, because its formulation only implicitly related to the inclusion of extra, non-core features, differently from the other *FD-BNE* items that contain explicit links.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Mattia Bianchi http://orcid.org/0000-0002-3799-5887
Giacomo Marzi http://orcid.org/0000-0002-8769-2462
Lamberto Zollo http://orcid.org/0000-0002-8162-8136
Andrea Patrucco http://orcid.org/0000-0002-9367-1561

References

Anderson, J. C., and D. W. Gerbing. 1988. "Structural Equation Modeling in Practice: A Review and Recommended Two-Step Approach." *Psychological Bulletin* 103 (3): 411–423.

Aranda, J., and S. Easterbrook. 2005. "Anchoring and Adjustment in Software Estimation." *Proceedings of the 10th European Software Engineering Conference ESEC/FSE-13* 30 (5): 346–355.

Bagozzi, R. P., and Y. Yi. 1988. "On the Evaluation of Structural Equation Models." *Journal of the Academy of Marketing Science* 16 (1): 74–94.

Baron, R. A. 2007. "Behavioral and Cognitive Factors in Entrepreneurship: Entrepreneurs as the Active Element in New Venture Creation." *Strategic Entrepreneurship Journal* 1 (1–2): 167–182.

Bazerman, M., and D. A. Moore. 2013. *Judgment in Managerial Decision Making*. 8th ed. Hoboken, NJ: John Wiley & Sons.

Belk, R. W., and G. S. Coon. 1993. "Gift Giving as Agapic Love: An Alternative to the Exchange Paradigm Based on Dating Experiences." *Journal of Consumer Research* 20 (3): 393–417.

Belvedere, V., A. Grando and B. Ronen. 2013. "Cognitive Biases, Heuristics, and Overdesign: An Investigation on the Unconscious Mistakes of Industrial Designers and on Their Effects on Product Offering." In *Behavioral Issues in Operations Management: New Trends in Design, Management, and Methodologies*, edited by I. Giannoccaro, 125–139. London: Springer.

Bettis, R., A. Gambardella, C. Helfat, and W. Mitchell. 2014. "Quantitative Empirical Analysis in Strategic Management." *Strategic Management Journal* 35 (7): 949–953.

Bianchi, M., G. Marzi, and M. Guerini. 2018. "Agile, Stage-Gate and Their Combination: Exploring How They Relate to Performance in Software Development." *Journal of Business Research*. doi:10.1016/j.jbusres.2018.05.003.

Bjarnason, E., K. Wnuk, and B. Regnell. 2012. "Are You Biting off More Than You can Chew? A Case Study on Causes and Effects of Overscoping in Large-Scale Software Engineering." *Information and Software Technology* 54 (10): 1107–1124.

Boehm, B. W. 1991. "Software Risk Management: Principles and Practices." *IEEE Software* 8 (1): 32–41.

Buschmann, F. 2009. "Learning From Failure, Part 1: Scoping and Requirements Woes." *IEEE Software* 26 (6): 68–69.

Chen, C. C., C. C. Law, and S. C. Yang. 2009. "Managing ERP Implementation Failure: A Project Management Perspective." *IEEE Transactions on Engineering Management* 56 (1): 157–170.

Chiu, K. F., L. K. Chu, and D. Sculli. 2001. "Modular Generic Software System for Cellular FMS." *International Journal of Production Research* 39 (12): 2545–2566.

Choi, K., and D. H. Bae. 2009. "Dynamic Project Performance Estimation by Combining Static Estimation Models with System Dynamics." *Information and Software Technology* 51 (1): 162–172.

Chow, T., and D. B. Cao. 2008. "A Survey Study of Critical Success Factors in Agile Software Projects." *Journal of Systems and Software* 81 (6): 961–971.

Christensen, C. M., and J. L. Bower. 1996. "Customer Power, Strategic Investment, and the Failure of Leading Firms." *Strategic Management Journal* 17 (3): 197–218.

Christensen, C. M., and M. E. Raynor. 2003. "Why Hard-Nosed Executives Should Care About Management Theory." *Harvard Business Review* 81 (9): 66–75.

Churchill, G. A. 1979. "A Paradigm for Developing Better Measures of Marketing Constructs." *Journal of Marketing Research* 16 (1): 64–73.

Coman, A., and B. Ronen. 2010. "Icarus' Predicament: Managing the Pathologies of Overspecification and Overdesign." *International Journal of Project Management* 28 (3): 237–244.

Conforto, E. C., F. Salum, D. C. Amaral, S. L. da Silva, and L. F. M. de Almeida. 2014. "Can Agile Project Management be Adopted by Industries Other Than Software Development?" *Project Management Journal* 45 (3): 21–34.

Cule, P., R. Schmidt, K. Lyytinen, and M. Keil. 2000. "Strategies for Heading Off is Project Failure." *Information Systems Management* 17 (2): 61–69.

Dalecki, M. G., J. C. Whitehead, and G. C. Blomquist. 1993. "Sample Non-Response Bias and Aggregate Benefits in Contingent Valuation: An Examination of Early, Late and Non-Respondents." *Journal of Environmental Management* 38 (2): 133–143.

Damian, D., and J. Chisan. 2006. "An Empirical Study of the Complex Relationships Between Requirements Engineering Processes and Other Processes That Lead to Payoffs in Productivity, Quality, and Risk Management." *IEEE Transactions on Software Engineering* 32 (7): 433–453.

DeMarco, T., and T. Lister. 2003. "Risk Management During Requirements." *IEEE Software* 20 (5): 99–101.

De Visser, M., D. Faems, K. Visscher, and P. de Weerd-Nederhof. 2014. "The Impact of Team Cognitive Styles on Performance of Radical and Incremental NPD Projects." *Journal of Product Innovation Management* 31 (6): 1167–1180.

Eliëns, R., K. Eling, S. Gelper, and F. Langerak. 2018. "Rational Versus Intuitive Gatekeeping: Escalation of Commitment in the Front End of NPD." *Journal of Product Innovation Management* 35 (6): 890–907.

Ettlie, J. E., and J. M. Elsenbach. 2007. "Modified Stage-Gate® Regimes in New Product Development." *Journal of Product Innovation Management* 24 (1): 20–33.

Fontela, E., J. Guzmán, M. Pérez, and F. J. Santos. 2006. "The Art of Entrepreneurial Foresight." *Foresight (Los Angeles, California)* 8 (6): 3–13.

Fornell, C., and D. F. Larcker. 1981. "Structural Equation Models with Unobservable Variables and Measurement Error: Algebra and Statistics." *Journal of Marketing Research* 18 (3): 382–388.

Furr, N. R., and J. Dyer. 2014. *The Innovator's Method: Bringing the Lean Startup Into Your Organization*. Boston, MA: Harvard Business Press.

Gino, F., and G. Pisano. 2008. "Toward a Theory of Behavioral Operations." *Manufacturing & Service Operations* 10 (4): 676–691.

Gopal, P. R. C., and J. Thakkar. 2016. "Sustainable Supply Chain Practices: An Empirical Investigation on Indian Automobile Industry." *Production Planning & Control* 27 (1): 49–64.

Gunasekaran, A., Y. Y. Yusuf, E. O. Adeleye, and T. Papadopoulos. 2018. "Agile Manufacturing Practices: The Role of big Data and Business Analytics with Multiple Case Studies." *International Journal of Production Research* 56 (1–2): 385–397.

Hair, J. F., C. M. Ringle, and M. Sarstedt. 2011. "PLS-SEM: Indeed a Silver Bullet." *Journal of Marketing Theory and Practice* 19 (2): 139–152.

Hair, J. F., W. C. Black, B. J. Babin, R. E. Anderson, and R. L. Tatham. 2006. *Multivariate Data Analysis*. 6th ed. Upper Saddle River, NJ: Prentice-Hall International.

Hoang, H. A., and F. T. Rothaermel. 2010. "Leveraging Internal and External Experience: Exploration, Exploitation, and R&D Project Performance." *Strategic Management Journal* 31 (7): 734–758.

Hodgkinson, G. P., and E. Sadler-Smith. 2003. "Reflections on Reflections . . . on the Nature of Intuition, Analysis and the Construct Validity of the Cognitive Style Index." *Journal of Occupational and Organizational Psychology* 76 (2): 279–281.

Hu, L. T., and P. M. Bentler. 1999. "Cutoff Criteria for Fit Indexes in Covariance Structure Analysis: Conventional Criteria Versus New Alternatives." *Structural Equation Modeling: A Multidisciplinary Journal* 6 (1): 1–55.

Iansiti, M. 1995. "Shooting the Rapids: Managing Product Development in Turbulent Environments." *California Management Review* 38 (1): 37–58.

Jarvis, C. B., S. B. MacKenzie, and P. M. Podsakoff. 2003. "A Critical Review of Construct Indicators and Measurement Model Misspecification in Marketing and Consumer Research." *Journal of Consumer Research* 30 (2): 199–218.

Jaworski, B. J., and A. K. Kohli. 1993. "Market Orientation: Antecedents and Consequences." *Journal of Marketing* 57 (July): 53–70.

Jayaram, J., A. Oke, and D. Prajogo. 2014. "The Antecedents and Consequences of Product and Process Innovation Strategy Implementation in Australian Manufacturing Firms." *International Journal of Production Research* 52 (15): 4424–4439.

Kahneman, D. 2003. "A Perspective on Judgment and Choice: Mapping Bounded Rationality." *American Psychologist* 58 (9): 697–720.

Karlsson, A., L. Larsson, and A. Öhrwall Rönnbäck. 2018. "Product-service System Innovation Capabilities: Linkages Between the Fuzzy Front End and Subsequent Development Phases." *International Journal of Production Research* 56 (6): 2218–2232.

Karlström, D., and P. Runeson. 2006. "Integrating Agile Software Development Into Stage-Gate Managed Product Development." *Empirical Software Engineering* 11 (2): 203–225.

Keil, M., and M. Mähring. 2010. "Is Your Project Turning Into a Black Hole?" *California Management Review* 53 (1): 6–31.

Khorram Niaki, M., and F. Nonino. 2017. "Additive Manufacturing Management: A Review and Future Research Agenda." *International Journal of Production Research* 55 (5): 1419–1439.

Kirton, M. J. 2003. *Adaption and Innovation in the Context of Diversity and Change*. London: Routledge.

Kurihara, K., S. Takigawa, N. Nishiuchi, and M. Kitaoka. 2002. "Factory Automation Control Software Designing Method Based on Petri Nets." *International Journal of Production Research* 40 (15): 3605–3625.

Lee, G., and W. Xia. 2010. "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility." *Mis Quarterly* 34 (1): 87–114.

Lee-Kelley, L., and T. Sankey. 2008. "Global Virtual Teams for Value Creation and Project Success: A Case Study." *International Journal of Project Management* 26 (1): 51–62.

Leonard, N. H., L. L. Beauvais, and R. W. Scholl. 2005. "A Multi-level Model of Group Cognitive Style in Strategic Decision Making." *Journal of Managerial Issues* 17 (1): 119–138.

Lettl, C., C. Hienerth, and H. G. Gemuenden. 2008. "Exploring How Lead Users Develop Radical Innovation: Opportunity Recognition and Exploitation in the Field of Medical Equipment Technology." *IEEE Transactions on Engineering Management* 55 (2): 219–233.

Liao, Y., F. Deschamps, E. D. F. R. Loures, and L. F. P. Ramos. 2017. "Past, Present and Future of Industry 4.0-a Systematic Literature Review and Research Agenda Proposal." *International Journal of Production Research* 55 (12): 3609–3629.

Loch, C. H., A. DeMeyer, and M. Pich. 2011. *Managing the Unknown: A new Approach to Managing High Uncertainty and Risk in Projects*. Hoboken, NJ: John Wiley & Sons.

Lovallo, D., and D. Kahneman. 2003. "Delusions of Success." *Harvard Business Review* 81 (7): 56–63.

Lovallo, D., and O. Sibony. 2010. "Taking the Bias Out of Meetings." *McKinsey Quarterly* 2 (2010): 68–69.

Luzzini, D., M. Amann, F. Caniato, M. Essig, and S. Ronchi. 2015. "The Path of Innovation: Purchasing and Supplier Involvement Into New Product Development." *Industrial Marketing Management* 47 (2015): 109–120.

MacCormack, A. 2001. "How Internet Companies Build Software." *MIT Sloan Management Review* 42 (2): 75–84.

MacCormack, A., R. Verganti, and M. Iansiti. 2001. "Developing Products on "Internet Time": The Anatomy of a Flexible Development Process." *Management Science* 47 (1): 133–150.

MacKenzie, S. B., and P. M. Podsakoff. 2012. "Common Method Bias in Marketing: Causes, Mechanisms, and Procedural Remedies." *Journal of Retailing* 88 (4): 542–555.

Maqueira, M. J., J. Moyano-Fuentes, and S. Bruque. 2018. "Drivers and Consequences of an Innovative Technology Assimilation in the Supply Chain: Cloud Computing and Supply Chain Integration." *International Journal of Production Research*. doi:10.1080/00207543.2018.1530473.

Markovitch, D. G., J. H. Steckel, A. Michaut, D. Philip, and W. M. Tracy. 2015. "Behavioral Reasons for New Product Failure: Does Overconfidence Induce Overforecasts?" *Journal of Product Innovation Management* 32 (5): 825–841.

Messick, S. 1976. "Personality Consistencies in Cognition and Creativity." In *Individuality in Learning*, edited by S. Messick, 4–22. San Francisco, CA: Jossey-Bass.

Misra, S. C., V. Kumar, and U. Kumar. 2010. "Identifying Some Critical Changes Required in Adopting Agile Practices in Traditional Software Development Projects." *International Journal of Quality & Reliability Management* 27 (4): 451–474.

Nadkarni, A., and S. G. Hofmann. 2012. "Why do People Use Facebook?" *Personality and Individual Differences* 52 (3): 243–249.
Nunnally, J. C., and I. H. Bernstein. 1994. *Psychometric Theory*. 3rd ed. New York: McGraw-Hill.
Olsson, U. H., T. Foss, S. V. Troye, and R. D. Howell. 2000. "The Performance of ML, GLS, and WLS Estimation in Structural Equation Modeling Under Conditions of Misspecification and Non-Normality." *Structural Equation Modeling* 7 (4): 557–595.
Peng, G., and J. Mu. 2018. "Do Modular Products Lead to Modular Organisations? Evidence From Open Source Software Development." *International Journal of Production Research* 56 (20): 6719–6733.
Perols, J., C. Zimmermann, and S. Kortmann. 2013. "On the Relationship Between Supplier Integration and Time-to-Market." *Journal of Operations Management* 31 (3): 153–167.
Petrovčič, A., G. Petrič, and K. L. Manfreda. 2016. "The Effect of Email Invitation Elements on Response Rate in a Web Survey Within an Online Community." *Computers in Human Behavior* 56 (March): 320–329.
Podsakoff, P. M., S. B. MacKenzie, J. Y. Lee, and N. P. Podsakoff. 2003. "Common Method Biases in Behavioral Research: A Critical Review of the Literature and Recommended Remedies." *Journal of Applied Psychology* 88 (5): 879–903.
Repenning, N. P. 2001. "Understanding Fire Fighting in New Product Development." *Journal of Product Innovation Management* 18 (5): 285–300.
Robinson, P. B., D. V. Stimpson, J. C. Huefner, and H. K. Hunt. 1991. "An Attitude Approach to the Prediction of Entrepreneurship." *Entrepreneurship Theory and Practice* 15 (4): 13–32.
Rogelberg, S. G., and J. M. Stanton. 2007. "Introduction: Understanding and Dealing with Organizational Survey Nonresponse." *Organizational Research Methods* 10 (2): 195–209.
Ronen, B., and S. Pass. 2008. *Focused Operations Management: Doing More with Existing Resources*. New York: John Wiley & Sons.
Ropponen, J., and K. Lyytinen. 2000. "Components of Software Development Risk: How to Address Them? A Project Manager Survey." *IEEE Transactions on Software Engineering* 26 (2): 98–112.
Rust, R. T., D. V. Thompson, and R. Hamilton. 2006. "Defeating Feature Fatigue." *Harvard Business Review* 84 (2): 37–47.
Schmidt, R., K. Lyytinen, and M. Keil. 2001. "Identifying Software Project Risks: An International Delphi Study." *Journal of Management Information Systems* 17 (4): 5–36.
Shaver, K. G. 1987. *Principles of Social Psychology*. 3rd ed. Cambridge, MA: Winthrop.
Shmueli, O., and B. Ronen. 2017. "Excessive Software Development: Practices and Penalties." *International Journal of Project Management* 35 (1): 13–27.
Shmueli, O., N. Pliskin, and L. Fink. 2015. "Explaining Over-Requirement in Software Development Projects: An Experimental Investigation of Behavioral Effects." *International Journal of Project Management* 33 (2): 380–394.
Shmueli, O., N. Pliskin, and L. Fink. 2016. "Can the Outside-View Approach Improve Planning Decisions in Software Development Projects?" *Information Systems Journal* 26 (2016): 395–418.
Steenkamp, J. B. E., and I. Geyskens. 2006. "How Country Characteristics Affect the Perceived Value of Web Sites." *Journal of Marketing* 70 (3): 136–150.
Sull, D. N. 2004. "Disciplined Entrepreneurship." *MIT Sloan Management Review* 46 (1): 71–77.
Thomke, S., and D. Reinertsen. 1998. "Agile Product Development: Managing Development Flexibility in Uncertain Environments." *California Management Review* 41 (1): 8–30.
Thompson, D. V., R. W. Hamilton, and R. T. Rust. 2005. "Feature Fatigue: When Product Capabilities Become Too Much of a Good Thing." *Journal of Marketing Research* 42 (4): 431–442.
Thomson, M., D. MacInnis, and W. Park. 2005. "The Ties That Bind: Measuring the Strength of Consu- Mer's Emotional Attachment to Brands." *Journal of Consumer Psychology* 15 (1): 77–91.
Tripp, J. F., and D. J. Armstrong. 2016. "Agile Methodologies: Organizational Adoption Motives, Tailoring, and Performance." *Journal of Computer Information Systems* 58 (2): 1–10.
Tversky, A., and D. Kahneman. 1974. "Judgment Under Uncertainty: Heuristics and Biases." *Science* 185 (4157): 1124–1131.