

Enhancing software development through project-based learning and the quality of planning

R&D plays a crucial role in developing new products, the commercialisation of which can drive corporate growth. Over three decades, research has focused the new product development (NPD) process and it is known that developing new products is a knowledge-intensive, risky activity. Since industry surveys show that many NPD projects, particularly software-based ones, fail to meet their schedules and objectives. Consequently, today's R&D managers still need ways to plan and conduct NPD more effectively. Project-based Learning (PBL)—the generation of specific technical and process knowledge during and after a project—is a potential way to improve NPD. Therefore, this paper investigated the research question: Does PBL enhance the quality of planning in subsequent software development projects? The study used a sample of 47 software development projects at three multinational organisations. Significantly, the findings show that PBL does enhance the quality of planning of subsequent software development projects. In particular, the quality of planning is increased in projects with high levels of uncertainty; where team members work in a project-based structure with strong collaboration; and when the pressure to deliver projects is high. The contribution of the research at a theoretical level is that it identified an important link between learning and the quality of planning in subsequent NPD projects. At a practical level, the study identifies specific steps R&D managers can take to improve the performance of software development projects, with all their associated challenges.

1. Introduction

The R&D function plays a fundamental role in developing new products, the commercialisation of which can enhance profit, strengthen competitiveness and boost corporate growth (Song et al., 2006; Goffin and Koners, 2011). Over recent decades, investment in R&D has grown significantly (Yigitcanlar et al., 2018), as companies invest more in new product development (NPD). Extensive research on the NPD process conducted over three decades (e.g. Balachandra and Friar, 1997; Cooper and Edgett, 2007; Cooper, 2017) has identified that developing products is a knowledge-intensive activity (Söderquist, 2006; Goffin and Koners, 2011). Despite much research, industry surveys still show that many NPD projects do not achieve their technical and market objectives, and fail to meet schedules and budgets (e.g. Markham and Lee, 2013). Therefore, today's R&D managers still require ways in which to plan and conduct NPD more effectively.

The literature highlights the importance of managing knowledge in NPD (e.g. Colomo-Palacios et al., 2018; Goffin and Koners, 2011; Boh et al., 2007) but many organisations do not apply what has been learnt from previous projects to current ones (Almeida and Soares, 2014; Cantner et al., 2011). In the context of NPD, a crucial aspect of knowledge management is Project-based Learning (PBL)—the generation of technical and process knowledge from a specific project. Such learning should occur both during and at the end of product development (Colomo-Palacios et al., 2018; Goffin and Koners, 2011). Learning can help improve the management of subsequent projects and help organisations build expertise (Ingram and Simons, 2002; Nonaka and von Krogh, 2009). It has been found that PBL enhances productivity (Prencipe and Tell, 2001); prevents companies from repeating mistakes (Yang et al., 2020; Goffin and Koners, 2011); and helps managers develop more realistic project plans (De Vasconcelos et al., 2017; Anbari et al., 2008). *Planning-performance theory* suggests that effective planning has a positive impact on not only on a financial performance (Brinckmann et al., 2010; Mueller et al., 2007) but also on the performance

of individual NPD projects (Van Oorschot et al., 2018; Amaral Féris et al., 2017; De Vasconcelos et al., 2017; Serrador and Turner, 2015). However, the relationship between learning from projects and the quality of planning of subsequent projects has not been investigated.

NPD projects are known for their uncertainty and for struggling to meet their objectives. Even more so, software development projects are notorious for being late and failing to meet their design goals (Serrador and Turner, 2015), with one recent study showing that only 39% of software projects were completed on-time, on-budget, and to-specification (Standish Group, 2015). At the same time, more and more products are software-based (Andreessen, 2011). In the car industry, for example, software-based navigation, entertainment and autonomous driving functionality have required massive investment (US\$ 16 billion in 2018; PwC, 2018) but, at the same time, have significantly increased car manufacturers' revenues (McKinsey & Company, 2019). Considering both their ubiquity and inadequacies, software-based projects are an appropriate context in which to study PBL, yet few studies have been conducted in this context (Akgün, 2020). Software projects account for a massive 78% of investment in NPD (Deloitte, 2019) and so understanding how they can be better managed is a pressing problem for both R&D managers and scholars.

Based on the gap in the literature, we chose the research question: *Does project-based learning enhance the quality of planning in subsequent software development projects?* To answer this question, we conducted a study of 47 projects conducted by three multinational organisations that conduct software development. The three organisations were an appropriate sample, as they operate in very competitive markets and are under constant pressure to deliver innovative products (c.f. Cheng and Yang, 2019). The results show that learning from previous projects *does* improve the quality of planning in subsequent software-based projects. Aligning this finding to planning-performance theory, where research has shown that planning improves project performance, reveals a potential way to improve the R&D management of software projects.

Studying potential moderators, it was found that learning is more effective when

team members work in project-based structures; when there is strong collaboration; when the pressure to deliver the project is high; and learning is more effective when project uncertainty is high. All four of these factors strengthen the positive impact of PBL on quality of planning of subsequent projects. The findings have strong implications for theory, in that they illustrate the relationship between learning and planning and they identify important moderating factors. For practitioners, the results indicate ways in which managers can counter the problem of poorly-performing software projects.

The rest of this paper is structured as follows. First, we review the literature on new product development, learning, and software-based projects. Second, we develop our hypotheses. Third, we describe the methodology adopted and, fourth, we explain the results. Fifth, we present the discussion and conclusions, including the contribution to theory and practice plus suggestions for future research.

2. Literature review

Although the importance of learning in NPD has been widely investigated, its relationship to planning has not. So, to provide apposite understanding of the literature, three areas were reviewed: NPD and learning; project planning; and the context of software development.

2.1. New product development and learning

New product development has R&D at its core but it is a cross-functional activity (Cooper, 2007), which is complex and inherently risky (Hird et al., 2016). This leads to widespread poor project performance (Markham and Lee, 2013) and to a high rate of market failure (Castellion and Markham, 2012). In addressing these problems, much research has focused on the NPD process (e.g. Balachandra and Friar, 1997; Cooper and Edgett, 2007; Cooper, 2017). As new products are reliant on the generation of new knowledge (Söderquist, 2006;), the knowledge management and learning perspective is also very relevant in the study of NPD (Goffin and Koners, 2011).

The knowledge management literature describes *learning processes* as building, storing and sharing knowledge (e.g. Argote and Hora, 2017). Learning processes can occur at the organisational-, project- and individual-levels (Prencipe and Tell, 2001). From the three levels, PBL is the most relevant in R&D. This is because PBL enables the transfer of knowledge between projects and ensures that learning from previous projects enhances the planning and performance of current projects. PBL can involve knowledge sharing between individuals in an ad-hoc and unstructured manner, or through formal mechanisms, such as databases, that capture an organisation's knowledge—and form its *organisational memory* (Olaisen and Revang, 2017; Frank et al., 2015; Boh and Wong, 2013). To strengthen a company's organisational memory, learning must consist of more than individuals gaining experience (Levitt and March, 1988) and of knowledge transfer between individuals. Rather, steps must be taken to ensure learning at an organisational-level (Aerts et al., 2017).

One particular challenge is that knowledge can be *tacit*, that is difficult to capture, document and share (Goffin and Koners, 2011). The transfer of tacit knowledge is dependent on individuals' willingness to share their knowledge (Hwang et al., 2018; Park and Lee, 2014; Ryan and O'Connor, 2013; Nonaka and von Krogh, 2009). As a continuous (Liberatore and Pollack-Johnson, 2013) and dynamic process (Tyagi et al., 2015), PBL promotes a constant interplay between an organisation's tacit and explicit knowledge (Mitchell et al., 2016; Nonaka and von Krogh, 2009; Marsh and Stock, 2006).

The temporary nature of projects often hampers the capture of knowledge within the organisation (Bakker et al., 2011) and interrupts the flow of learning between projects (Bartsch et al., 2013). One mechanism to promote PBL is *post-project reviews* (Koners and Goffin, 2007). Held at the end of projects, such reviews encourage team members to reflect on their shared experience and identify opportunities to improve the performance of subsequent projects.

Although learning from projects and transferring knowledge is appealing, achieving it can

be very difficult (Almeida and Soares, 2014; Lewin et al., 2011). Consequently, knowledge management is no longer regarded as a practical, effective tool for managers (Rigby and Bilodeau, 2018). To manage knowledge effectively, it is essential that tools and approaches are applied in different ways depending on, for example, the industry in question (Shenhar and Dvir, 2007). Then, despite the challenges involved, the application of learning from project teams can help improve subsequent projects (Akgün, 2020).

2.2. *Project planning*

Planning-performance theory suggests that the *quality of planning* has a direct impact on project performance (Van Oorschot et al., 2018; Amaral Féris et al., 2017; De Vasconcelos et al., 2017; Serrador and Turner, 2015). The work of “the project manager (PM) influences directly the quality of the project plan document” (Amaral Féris et al., 2017, p. 92), which supports the achievement of project objectives (Salomo et al., 2007). Project planning can help in the early stages before major investments are made and when uncertainty is high (Hird et al., 2016), through to the stages where both the effort and the level of investment is high. Overall, proper planning (Serrador and Turner, 2015) can reduce the high levels of uncertainty and complexity in NDP projects (Hird et al., 2016); mitigating the inevitable changes in customer requirements (Sutherland, 2014; Van Oorschot et al., 2010); and even changes in project objectives (Salomo et al., 2007).

Quality of planning is a complex construct but the literature identifies *checklists*, *metrics* and *models* as the three most effective ways it can be evaluated. Based on experts’ knowledge and experience (Houston, 2004), checklists can be used to identify risks; to check whether the project can move to the next phase (Keil et al., 2008); to ensure that the project is compliance with the organisation’s policies; and to prevent errors or omissions that may impact the project significantly (Riek, 2001). However, the over-use of checklists can lead to distraction and focusing on non-existent risks (Keil et al., 2008), which is counterproductive (Houston, 2004).

Based on the principle that performance cannot be improved without first measuring it,

metrics are an efficient feedback mechanism for increasing productivity and product quality (Daskalantonakis, 1992). Organisations often define irrelevant metrics (Symons, 2010), which are distracting and have the potential to create conflicts. Metrics need to focus not only on the technical aspects of projects but also ensuring that stakeholders viewpoints are considered (Gopal et al., 2002).

There are a number of evaluation models that can be used to evaluate the quality of planning in a project, as shown in Table 1. It can be seen that, over time (2004-2018), these models have become more sophisticated, with more factors being considered. For example, the QPM Model assesses the quality of planning by having managers score 16 factors, whereas the QPLAN Model considers 55 factors and is based on contingency theory (Shenhar and Dvir, 2007). It should be noted that three of these models (UQIM, QPLAN and SPSRI) have specifically been applied to software development projects, the context that we will now discuss.

Table 1: Models that evaluate planning quality

Please Insert Table 1 here

2.3. Software Development projects

Despite the prevalence and importance of software development, poor project performance has plagued the industry for years (Sutherland, 2014; Drury-Grogan, 2014; Lehtinen et al., 2014; Shanling et al., 2010). One study showed that only 39% of software projects were completed on-time, on-budget and meeting the product specifications (Standish Group, 2015). A more recent study of IT projects showed that 71% met their business goals but only 64% were within budget, and only 59% were completed on time (PMI, 2020). So, in the software development context, there is a particular and urgent need to learn how to improve project performance. However, the volume of research on learning in the software development context is low. Extant studies have looked at the organisational- and project-levels, plus the types of project where knowledge is crucial.

The studies that have analysed knowledge at the organisational-level include, for example,

Khuong et al. (2014) who published a case study on a global IT company and showed that changing system specifications made knowledge difficult to manage. Goldoni and Oliveira's (2010) multiple case studies showed that a structured approach to managing the knowledge generated can lead to better project performance. Kautz and Thaysen (2001) conducted a single case study in a small organisation and found that having a formal knowledge management system promotes both individual and team learning. Segelod and Jordan (2004) employed multiple case studies to show the importance of external sources of knowledge in defining project scope. Note, however, none of these studies looked at the impact of learning on subsequent projects.

The studies that have analysed the impact of knowledge and learning at the project-level include De Vasconcelos et al. (2017), who conducted a case study on software maintenance (modifying software to correct errors). They found that effective management of project-related knowledge improves software and reduces maintenance effort. Serna et al. (2017) performed a systematic literature review and identified ambiguity in different individuals' interpretations of software specifications as an issue. Similarly, De Souza et al. (2015) looked at the literature, finding that knowledge management does not increase the effectiveness of software testing. Colomo-Palacios et al. (2018) used the case method to study the relationship between software development and its operational deployment, finding that knowledge management can positively impact current project performance. Similarly, Mitchell et al. (2016) used a case study and found that formal knowledge management improves both the development process and software quality. Again, it should be noted that none of these studies looked at the impact of learning on subsequent projects.

The studies that have analysed the relationship between knowledge and learning and the type of project include Rashid et al. (2019). They performed a systematic literature review focused on open source software (OSS) and found that not enough attention is given to managing knowledge. Iskoujina and Roberts (2015) used an online survey and found that the way OSS projects are managed influences contributors' motivation to share their knowledge. Li et al. (2006) found that

having a formal knowledge management system helped make enterprise resource planning (ERP) projects more effective. Again, it should be noted all of these studies looked at the learning *within* projects and did not consider the impact of knowledge and learning on subsequent projects. Thus, as noted by Ghobadi (2015), the extant literature on software development has not generated a complete picture of the factors involved.

3. Development of hypotheses

Our literature review showed that software development is under-researched although case study research has demonstrated that knowledge and learning are important in individual projects. What the extant literature has overlooked is the impact of learning on subsequent projects and this led to our research question: *Does project-based learning enhance the quality of planning in subsequent software development projects?* Based around this question, a number of hypotheses were developed, including:

H1: *PBL is positively associated with the quality of planning in subsequent software development projects.*

Based also on the literature, we identified four potential moderating factors that influence the impact of PBL on the planning of subsequent software development projects. These were: *project uncertainty* (Dönmez and Grote, 2018; Ahmad et al., 2012); *team collaboration* (Pemartin et al., 2018; Frank et al., 2015); *project-based structures* (Gemünden et al., 2018; Ren et al., 2018) and *time pressure* (Cheng and Yang, 2019; Van Berkel et al., 2016).

3.1. Project uncertainty

Project uncertainty has been conceptualised as the difference between the information needed to perform a task and the amount of information initially available within an organisation (Galbraith, 1973). High levels of uncertainty mean that risks need to be assessed during planning (Ahmad et al., 2012). Consequently, Dönmez and Grote (2018) proposed a set of practices for

addressing uncertainty, including flexible planning and considering alternative solutions. Similarly, for projects with high uncertainty, Peeters and Martin (2017) showed the importance of complementing external knowledge with internal perspectives. Uncertainty is often viewed negatively and so its potential to stimulate learning has tended to be overlooked by researchers. However, Um and Kim (2018) identified that higher levels of uncertainty can increase collaboration and Ruuska and Brady (2011) argued that teams are more likely to learn from past projects when they are working on highly uncertain projects. In line with these ideas, we argue that the team members are motivated to proactively search for knowledge from past projects, when their project encounters high uncertainty. This led to the hypothesis:

H2: PBL in software development projects is stronger when uncertainty levels are high.

3.2 Team collaboration

Software development requires deep collaboration between different experts (e.g. software engineers, systems engineers and quality assurance). At the project-level, intense collaboration can avoid the repetition of past mistakes (Yang et al., 2020; Goffin and Koners, 2011); it can enable strategies that were effective in previous projects to be applied again (Ruuska and Brady, 2011); and it can lead to ways to increase project performance (Bartsch et al., 2013). At the organisational-level, it enables the transfer of both tacit and explicit knowledge from individual projects into organisational memory (Olaisen and Revang, 2017; Frank et al., 2015; Boh and Wong, 2013). Such memory enhances an organisation's productivity (Prencipe and Tell, 2001) and profitability (Ingram and Simons, 2002). However, collaborative behaviour (Pemartin et al., 2018; Smets et al., 2016) is influenced by both individual-related factors (e.g. dependence and trust – Park and Lee, 2014) and organisational-related factors (e.g. managerial politics – Kyriazis et al., 2017), which can lead to deliberate withholding or distorting of information (Smets et al., 2016). Consequently, we posit the following hypothesis:

H3: *PBL is stronger when team collaboration is higher.*

3.3. *Project-based structure*

Given the importance of structure afforded by the project management field (e.g. Delisle, 2019; Hodgson and Cicmil, 2007), we adopted definitions from the PMBOK (PMI, 2017). These are: a) *Functional structure*, where team members will work in independent departments; b) *Matrix structure*, where team members have multiple managerial accountability and responsibility; and c) *Project-based structure*, where team members operate in a cohesive and cross-functional integrated atmosphere aimed at facilitates the knowledge sharing (Belout and Gauvreau, 2004). (Note: these are sometimes called *projectised* structures.) Researchers have argued that all three structures, to different extents, divide team members into groups, which are aligned to the achievement of specific goals (Jaakkola and Hallin, 2018). Such division affects the interaction between team members (Frank et al., 2015; Lewin et al., 2011); the frequency with which learning mechanisms are utilised (Slaughter and Kirsch, 2006); and the learning outcomes (Lechler and Dvir, 2010). The division can also lead to a lack of uniformity in the way organisational processes are applied (Almeida and Soares, 2014) and increase the risk of knowledge loss (Zhao et al., 2015). Thus, we test:

H4: *PBL is stronger when an organisation adopts a project-based structure (as opposed to matrix and functional structures).*

3.4. *Time pressure*

In psychology, the *behaviour theory of timing* (Killeen and Fetterman, 1988) suggests that time pressure is an omnipresent phenomenon, in both our personal and professional lives. Research indicates that individuals have an internal ‘clock-like system’, which helps stimulate the execution of a task in the time available (Rattat et al., 2018). In the innovation field, several researchers have claimed that there is an inverted U-shaped relationship between time pressure and innovation,

where moderate levels of pressure challenges team members and stimulates their creativity, whereas too little or too much pressure will constrain creativity (e.g. Khedhaouria et al., 2017). When managers focus only on short-term performance (Nohria and Gulati, 1996), it can lead to too much time pressure and project teams will not have enough slack to learn from the past (Ren et al., 2018; Wiengarten et al., 2017; Richter et al., 2013). However, these studies somewhat neglect other factors that can impact learning positively, such the high visibility levels of such projects (Van Berkel et al., 2016), and the opportunity for personal development and gain (Chong et al., 2011). Following this stream of research, we argue that time pressure will motivate team members to learn from past projects aimed at reducing the inherent uncertainty of NPD projects:

H5: *PBL is stronger under high time pressure.*

4. Methodology

4.1 Choice of method

Considering that the concepts PBL and the quality of planning have previously been investigated separately, it was decided that a quantitative investigation of their interaction was appropriate. The investigation was influenced by the principles of Design Science Research (DSR), which stress the importance of management *artefacts*—tools which can be applied by managers (Van Aken, 2004), to solve real problems (Hevner et al., 2004). This led to the adoption of an existing tool (QPLAN), which had already been evaluated and shown to evaluate the quality of planning in software development projects (Amaral Féris et al., 2017).

4.2. Data collection procedure

For each of the 47 projects, we conducted an open-ended preliminary interview with the respective senior manager (c.f. Rossman and Rallis, 2003). These interviews verified that senior managers did not identify factors that were not covered in the QPLAN tool. Then, for each project,

multiple respondents completed questionnaires at three points during the project life cycle. First, before planning commenced, project managers reported on the independent, moderators and dummy variables. Second, at the end of planning, project managers were asked to evaluate planning quality. Third, at the end of the project, project managers were asked to identify insights that may improve the performance of subsequent projects (a simple form of PPR). Questionnaires were administrated in English by e-mail and online between 2011 and 2018. Data from the questionnaires were input into the QPLAN Model, which generated an evaluation of the quality of planning and suggestions to improve it. The content of these reports was then discussed with the respective project managers.

4.3. Sample

The unit of analysis was a project. From a large, longitudinal study of over 161 projects in 13 companies, three multinational companies based in Brazil, US and Israel agreed to provide recurrent access to a total of 47 software development projects (see Table 2), on condition of anonymity. The three companies operate in the industrial automation (Company ‘A’), printing (Company ‘B’) and defence (Company ‘C’) sectors and the projects chosen all developed software that is *embedded* in hardware devices. Many products require embedded software to function correctly, often in safety critical situations, and so this is a particularly important category of software (Garousi et al., 2018). The projects ranged in duration from three to 72 months with an average of 2.3 years.

Table 2: Sample profile

Please Insert Table 2 here

It should be noted that these organisations used the same NPD process (c.f. Kleinschmidt et al., 2007) and technology platforms (c.f. Olaisen and Revang, 2017), such as IBM DOORS (Hajri et al., 2018) and SVN (Greene et al., 2017) to manage their projects in different locations, with

some projects being conducted simultaneously in two locations. This means that there was no technical difference between the software development processes in the different countries of these organisations.

4.4. Measures

The constructs used in the questionnaires will be discussed. The Appendix lists all constructs, showing their respective items and references from the literature.

4.4.1. Dependent variable (DV)—Planning quality

The dependent variable planning quality was measured using QPM (see Table 1), which has been validated and widely-applied (e.g. Papke-Shields et al., 2010; Rees-Caldwell and Pennington, 2013; Zwikael et al., 2014; Amaral Féris et al., 2017). It is based on project managers' evaluations of the quality of 16 different *outputs* of planning (see Table 3). (Note: the project management literature refers to the 16 outputs as “Planning Products” but we have not adopted this term, as it is confusing when used in relation to NPD projects.) Managers gave their evaluations of the quality of each planning output, for which project managers are responsible (Zwikael and Globerson, 2004), using a five-point Likert scale an aggregate measure calculated through a weighted linear combination as defined in the PMBOK (PMI, 2017). The scale's alpha coefficient was calculated to be 0.93.

Table 3: Sixteen outputs of project planning defined in PMBOK (Zwikael and Globerson, 2004)

Please Insert Table 3 here

4.4.2. Independent variable (IV)—Project-based Learning

Following Godener and Söderquist's (2004) recommendation that NPD performance is best determined by multiple measures, we used two items for PBL. The first item was: “The organisation has past experience with similar projects”; this captured the amount of experience the organisation possessed from comparable projects (Shenhar and Dvir, 2007; Ahmad et al., 2012). The second

item was: “There is historical data that can be used for the development of the project”; this captured the degree of explicit knowledge available in organisational memory. Each of the items was measured on a five-point Likert scale (5=“Strongly agree” and 1=“Strongly disagree”).

4.4.3. Moderating variables (MV)

We included four moderating variables to test our hypotheses: (1) Project uncertainty was measured using a reverse 5-point Likert Scale (1=“Strongly agree” and 5=“Strongly disagree”); (2) Team collaboration was measured using a 5-point Likert Scale (5=“Strongly agree” and 1=“Strongly disagree”); (3) Project-based structure was coded as “0” for the project-based; “1” for matrix organisations and “2” for functional structures; and (4) Time pressure. was measured using a reverse Likert Scale (1=“Strongly agree” and 5=“Strongly disagree”).

4.4.4. Control variables (CV)

In addition, we included five control variables in our study: (1) “Concluded project” was used to control for projects that had been completed (coded “1”) versus those which were on-going (coded. “0”); (2) “Manager’s work experience” (in years) checked whether work experience impacts project outcomes; (3) “Project novelty” was used to control for measuring uncertainty of requirements (c.f. Shenhar and Dvir, 2007) and coded projects by whether they were extensions or improvements in currents products’ platforms (new generation of current product) or breakthroughs (new products); (4) “Technological novelty” checked for the uncertainty of know-how (c.f. Shenhar and Dvir, 2007) using coding of “low-tech”, “medium-tech”, “high-tech” and “super high-tech” (technologies that did not previously exist) and (5) “Organisation culture” controlled for whether there was a political environment that impacts on resources, priority and climate—this was measured on a 5-point Likert Scale.

4.4.5. Dummy variables (DMY)

Finally, we included dummy variables to control for fixed effects. “Strategic goal” was included because a project may receive different resourcing according to its strategic importance.

This variable was classified as (a) *Extension* (i.e. improvements in an existing product); (b) *Strategic* (i.e. strategic position for the business through NPD); (c) *Problem-solving* (i.e. acquiring or developing a new technology/capability); (d) *Maintenance* (i.e. routine maintenance or fixing regular problems). e) *Research* (i.e. exploring future ideas and without product in mind) (Shenhar and Dvir, 2007). Moreover, we included the dummy variables of the three organisations to control the time-invariant organisational-level effects, including organisational structure, climate, and product market. Last, we included the dummy variables of the three countries of the projects to control for the variations brought by the locations.

4.5 Data analysis

The statistical package used in this study was STATA. We regressed the *planning quality* on *PBL* to examine H1. We used principle component analysis (PCA) to generate the integrated variable of *PBL*. In a separate analysis, we generated the *PBL* variable through averaging the standardised scores of the two items. Both approaches yield similar results in the data analysis. To examine H2 to H4, we inserted the interaction terms between *PBL* and the four moderators separately. We performed standardisation to the *PBL* and continuous moderators. We also performed a natural logarithm to manager's work experience to correct for the skewness. Multicollinearity concerns were addressed by calculating the variance inflation factor–VIF. The average value was 2.38 and there was no variable's VIF value larger than 10 (Cohen et al., 2003).

5. Results

The research tested five hypotheses and investigated the potential effects of four intermediate variables. The descriptive statistics and correlations among the variables are presented in Table 4. It can be seen that the strongest correlation is 0.52 (shown bold in Table 4). Since no correlation coefficient is larger than 0.75, it can be concluded that there are no strong correlations among the independent variables (Lind et al., 2006).

Table 4: Descriptive statistics and correlations

Please Insert Table 4 here

The regression analysis is presented in Table 5. *Model 1* included all control variables, while *Model 2* added the independent variable to test hypothesis H1. The coefficient of PBL was significantly positive (0.073, $p < 0.01$). Moving the PBL from the mean to one standard deviation above the mean, planning quality increased by 0.073, or 11.1% $((0.073/0.655)*100\%)$. Thus, H1 is supported, meaning that PBL enhances the quality of planning of subsequent software development projects. The quality of *Models 1* and *2* was estimated by *Akaike Information Criterion* (AIC). The AIC of *Model 2* (-1.206) was smaller than in *Model 1* (-0.981), suggesting that adding the variable of PBL improved the goodness-of-fit of *Model 2*.

Table 5: Regression analysis of planning quality (direct effect)

Please Insert Table 5 here

The marginal effect of H1 is illustrated in Figure 1, where the x-axis is the PBL (IV) and the y-axis is the planning quality (DV).

Please Insert Figure 1 here

Figure 1: Marginal effect of planning quality (DV) as a function of PBL (IV)

In Table 6, *Model 3* examined the negative moderating effect of project uncertainty (H2) by adding the interaction term between PBL and project uncertainty. The coefficient of the interaction term was significantly negative (-0.041, $p < 0.05$). *Low level* of moderator (project uncertainty) was defined as one standard deviation below the mean (represented in Figure 2 by the line with the symbol “●”); while *high level* of moderator was defined as one standard deviation above the mean (“▲” line). Moving project uncertainty from the *mean* (“■” line), to the high level, the slope of PBL was attenuated by 0.041 or 56.2% $([(1-0)*0.041/0.073]*100\%)$. H2 is hence supported,

meaning that PBL is more effective in high uncertainty projects has a bigger impact on the quality of planning of subsequent projects.

Table 6: Regression analysis of planning quality (moderating effects)

Please Insert Table 6 here

Model 4 examined the positive moderating effect of *team collaboration* (H3) by adding the interaction term between PBL and *team collaboration*. The coefficient of the interaction term was significantly positive (0.060, $p < 0.01$). Moving team collaboration from the mean to the high level, the slope of PBL was amplified by 0.060 or 82.2% ($[0.060/0.073]*100\%$), as shown in Figure 2 (on the top, right-hand side). Thus, H3 is supported, meaning that PBL is more effective when team collaboration is high.

Model 5 examined the positive moderating effect of a *project-based structure* (H4) by adding the interaction term between PBL and *project-based structure*. The coefficient of the interaction term was found to be significantly positive (0.094, $p < 0.01$). The slope of PBL was amplified by 0.094 or 128.8% ($[0.094/0.073]*100\%$) in the project-based structure, compared to matrix and functional structures, as shown in Figure 2 (on the bottom, left-hand side). Therefore, H4 is supported, meaning that PBL is more effective when organisations adopt a project-based structure.

Model 6 examined positive moderating effect of *time pressure* (H5) by adding the interaction term between PBL and *time pressure*. The coefficient of the interaction term was found to be significantly positive (0.052, $p < 0.05$). Moving *time pressure* from the mean to the high level, the slope of PBL was amplified by 0.052, or 71.2% ($[0.052/0.073]*100\%$). Therefore, H5 is supported, meaning that PBL is more effective when team members are under high time pressure. The AICs of *Models 3 to 6* (Table 6) are all smaller than the AIC of *Model 2* (Table 5), which suggests that the goodness-of-fit of the models are improved by adding these moderators as shown in Figure 2.

Figure 2: Moderating effects

6. Discussion and Conclusions

The current research was motivated by three main considerations. First, by the substantial shift towards software-based projects in R&D. Second, by the poor performance of the majority of software-related projects. Third, by the observation that studies of software projects have looked at learning in isolation (e.g. Akgün, 2020; Ghobadi, 2015) without considering project planning. So, by studying not only project-based learning but also its impact on the quality of planning in subsequent software development, the current study investigated an important and timely topic.

6.1 Summary of Findings

A rich set of data from 47 software development software development projects in three multinational organisations was collected. Based on the data, five hypotheses could be tested and all five were supported (see Table 7). This means that PBL can enhance the quality of planning of software development projects (H1). An investigation was made of four potential moderating factors that influence the strength of the relationship between PBL and the quality of planning of software development projects. These moderators are known from the literature to influence PBL (the independent variable) but have not previously been shown to strengthen the impact of PBL on the quality of planning of software development projects (the dependent variable).

The results showed not only that PBL was stronger in projects with high levels of uncertainty (H2) [which is in line with the research carried out by Dönmez and Grote (2018); Peeters and Martin (2017)] but also that PBL strengthened the impact on the quality of planning of subsequent projects. Testing H3 showed that strong team collaboration increased PBL [confirming the findings of Yang et al. (2020) and Akgün (2020)] and strengthened the impact on the quality of planning of subsequent projects. Similar to Zwikaël and Sadeh (2007), we found that project-based structures

strengthened the positive impact of PBL on the quality of planning of subsequent projects (H4). Finally, we identified a positive correlation between time pressure and PBL [which is aligned with Van Berkel et al. (2016) and Chong et al. (2011) but partially contradicts Richtnér et al (2014)], again moderating the impact on the quality of planning of subsequent projects (H5). All these findings are supported by the statistical analysis that included moderating, control and dummy variables aimed at increasing robustness.

Table 7: Summary of findings

Please Insert Table 7 here

6.2 Contribution to theory

Based on the findings, Figure 3 is a tentative conceptual model, clearly showing how PBL (independent variable) in software development projects has a positive impact on the quality of planning of subsequent projects (dependent variable), moderated by four variables: project uncertainty, team collaboration, project-based structure and time pressure. The relationship between the different concepts can now be better understood. Furthermore, research based on planning-performance theory has produced strong empirical evidence (e.g. Serrador and Turner, 2015; Hird et al., 2016; Sutherland, 2014; Van Oorschot et al., 2010; Salomo et al., 2007) that planning improves project performance in NPD. Consequently, when our findings (Figure 3) are aligned with planning-performance theory, this reveals a way to improve the performance of software development projects—*by investing resources in project-based learning*.

Please Insert Figure 3 here

Figure 3: Tentative conceptual model based on the results

Practitioners perceive knowledge management activities to be difficult. However, if the difficulties can be overcome, our study indicates that efforts to enhance PBL can bring potentially

rich rewards, by improving the performance of software development projects. Since 41% or more of software projects are not on-time, to-budget and on-specification (PMI, 2020; Standish Group, 2015), there is extensive scope for better PBL in software development. Surprisingly, other researchers have already unlocked some of the mysteries of PBL (e.g. Richner, et al 2014), including how to stimulate the transfer of tacit knowledge (Goffin and Koners, 2011), but many organisations still do not apply what has been learnt from previous projects (Almeida and Soares, 2014; Cantnera et al., 2011) and knowledge management is perceived as difficult (Rigby and Bilodeau, 2018). Therefore, our findings should give scholars new momentum to investigate practical ways in which project-based learning can be stimulated in software development and NPD in general.

Four factors were known to influence PBL but our study shows that they also moderate the quality of planning of subsequent projects. This is an important result as all the factors are linked to perceptions and behaviour. For example, the perception of time pressure was found to be explained by the theory of timing from psychology. Such a behavioural perspectives will enhance our understanding of NPD teams. For example, project-based teams generally have more autonomy, management sponsorship, and access to resources. Presumably, the positive feelings this generates and the confidence this gives teams enables them to learn more and plan better. Similarly, uncertain projects have high visibility with management and so, although challenging, could lead to positive team behaviour. All such perceptions of teams and their impact on learning are important as more uncertainty leads to more collaboration (Um and Kim. 2018), which is particularly important in breakthrough and radical projects.

6.3 Contribution to practice

Recent research indicates that practitioners are losing confidence in the ability of knowledge management to improve project performance (Rigby and Bilodeau, 2018). Going against this, our research indicates that managers *should* attempt to improve knowledge management and learning

in R&D, as it can improve the quality of planning on subsequent projects which, in turn, will improve project performance. Our findings should give practitioners new impetus to work on ways to stimulate project-based learning. Although there has been a mass of research on NPD processes over the last three decades, this has not eradicated poor project performance. Currently, there are widespread discussions on the utility of agile NPD processes but our research indicates that these discussions should not be conducted without considering PBL. NPD processes need to be re-designed so that they not only embed learning opportunities into current projects but also exploit the learning generated by previous projects.

In addition to our quantitative data, our interviews with project managers generated a wealth of complementary, supportive qualitative data. These illuminate practitioners' perspectives on the factors discussed in this study on project performance. For example, one project managers from Company A identified a way to decrease inherent project uncertainty: *"We should work on software prototypes during the planning phase, so we can minimize the risks that will come after we start the implementation"*. Another project manager from the same organisation claimed that team collaboration improves project efficiency: *"Better communication between [the] hardware and [the] software designer could reduce the effort and refactoring of software development"*. One project manager from Company C made it clear that the project structure adopted (i.e. matrix) led to poor planning and a structure with more focus on the project was (i.e. a project-based structure): *"Planning was extremely poor: the matrix organisational structure did not help the technical manager to work with focus, which implied in overload and additional risks"*. Finally, a project manager from Company B said implicitly that extreme time pressure had caused the team not to consult the organisation's knowledge management system, which led to the project being delayed because of rework: *"Software modules developed in other projects were not reused. As a result, the project has delayed due to having to develop some modules from scratch"*. These exemplary quotes show that software managers do, indeed, perceive the value of project-based learning.

In addition, managers at the three organisations involved in the study found that the answering the QPLAN questions helped them enhance the quality of their NPD project planning, at both the organisational- and project-levels. For example, Company A decide to revise its internal processes from the experience of using QPLAN in eight of their software projects. A senior project manager said *“The company intends to carry out the necessary reviews in its internal processes, seeking greater efficiency related to their costs and deadlines”*. A senior manager from Company B recognised the value of a high-quality planning: *“Planning and tracking made after the third project manager joined the team was well-done, external dependencies were well tracked and response to scope changes was quick and effective”*. This finding is aligned with the feedback provided by a project manager from Company C: *“We performed a detailed work plan since from the beginning of the project, which included risk assessment in early stage”*.

6.4 Limitations and further research

There are three number of limitations of our study that should be considered. First, we only looked at software projects and not at NPD in general, however, as software-related projects constitute the majority of R&D work, this is not a serious limitation. Second, our sample of 47 embedded software projects was not representative of all types of software development. This means that future research should look to test our findings in other types of software projects, such as stand-alone programs and web-based applications. Third, our data looked at PBL and its impact on the quality of planning of subsequent projects, not the direct impact on the performance of subsequent projects (as indicated by Figure 3). Although evidence for planning-performance theory allowed us to theorise from our findings that PBL leads to better performance of subsequent projects, this should be tested directly. Here case study research at the project-level would be appropriate, particularly if ethnographic methods were used to understand behavioural factors (as mentioned earlier). Ethnography would generate a detailed understanding of social interaction (c.f. Stigliani and Ravasi, 2012) of how PBL takes place, how knowledge is transferred to other projects,

and how it positively impacts the performance of these subsequent projects. Such qualitative research would identify many ways in which R&D managers can implement PBL successfully.

7. Summary

This study investigated software development, a known challenge for R&D managers. The research question investigated was: Does project-based learning (PBL) enhance the quality of planning in subsequent software development projects? The results from 47 software development show that PBL not only enhances the quality of planning of subsequent projects but also (drawing on planning-performance theory) will improve the performance of software development projects. This is an important finding for scholars, as it links PBL and the quality of planning for the first time. For the majority of project managers in R&D that struggle to deliver software on-time, on-budget and on-spec, the results show the urgent need for practitioners to apply knowledge management ideas and, specifically, to apply tools to stimulate PBL.

References

- Aerts, G., Dooms, M. and Haezendonck, E. (2017) Knowledge transfers and project-based learning in large scale infrastructure development projects: an exploratory and comparative ex-post analysis. *International Journal of Project Management*, **35**, 3, 224-40.
- Ahmad, S., Mallick, D.N. and Schroeder, R.G. (2012) New Product Development: Impact of Project Characteristics and Development Practices on Performance. *Journal of Product Innovation Management*, **30**, 2, 331-348.
- Akgün, A.E. (2020) Team wisdom in software development projects and its impact on project performance. *International Journal of Information Management*, **50**, 228-243.

- Almeida, M.V. and Soares, A.L. (2014) Knowledge sharing in project-based organizations: Overcoming the informational limbo. *International Journal of Information Management*, **34**, 770–779.
- Amaral Féris, M.A. and Zwikael, O. (2017) Evaluating the Quality of Planning in New Product Development Projects. *In: Proceedings of the R&D Management Conference, Leuven*.
- Amaral Féris, M.A., Zwikael, O. and Gregor, S. (2017) QPLAN: Decision Support for Evaluating Planning Quality in Software Development Projects. *Decision Support Systems*, **96**, 92-102.
- Anbari, F.T., Carayannis, E.G. and Voetsch, R.J. (2008) Post-project reviews as a key project management competence. *Technovation*, **28**, 10, 633-643.
- Andreessen, M. (2011) *Why Software Is Eating the World*. Wall Street Journal. Retrieved 24/5/2020, from <http://online.wsj.com/articles/SB10001424053111903480904576512250915629460>
- Argote, L. and Hora, M. (2017) Organizational Learning and Management of Technology. *Production and Operations Management*, **26**, 4, 579–590.
- Bakker, R.M., Cambré, B., Korlaar, L. and Raab, J. (2011) Managing the project learning paradox: A set-theoretic approach toward project knowledge transfer. *International Journal of Project Management*, **29**, 5 494-503.
- Balachandra, R., and Friar, J.H. (1997) Factors for Success in R&D Projects and New Product Innovation: A Contextual Framework. *IEEE Trans. on Engineering Management*, **44**, 3, 276-287.
- Banker, R.D. and Hwang, I. (2008) Importance of Measures of Past Performance: Empirical Evidence on Quality of e-Service Providers. *Contemporary Accounting Research*, **25**, 2, 307-337.

- Bartsch, V., Ebers, M. and Maurer, I. (2013) Learning in project-based organizations: The role of project teams' social capital for overcoming barriers to learning. *International Journal of Project Management*, **31**, 2, 239-251.
- Belout, A. and Gauvreau, C. (2004) Factors influencing project success - the impact of human resource management. *International Journal of Project Management*, **22**, 1, 1–11.
- Boh, W.F. and Wong, S.S. (2013) Organizational Climate and Perceived Manager Effectiveness: Influencing Perceived Usefulness of Knowledge Sharing Mechanisms. *Journal of the Association for Information Systems*, **14**, 3, 122-152.
- Breiman, L. (2001) Random forests. *Machine Learning*, **45**, 1, 5–32.
- Cantner, U., Joel, K. and Schmidt, T. (2011) The effects of knowledge management on innovative success – An empirical analysis of German firms. *Research Policy*, **40**, 1453–1462.
- Castellion, G. and Markham, S.K. (2012) Perspective: New Product Failure Rates: Influence of Argumentum ad Populum and Self-Interest *Journal of Product Innovation Management*, **30**, 5, 976-79.
- Cheng, C. and Yang, M. (2019). Creative process engagement and new product performance: The role of new product development speed and leadership encouragement of creativity. *Journal of Business Research*, **99**, 215-225.
- Chong, D.S.F., Chai, K.H., Van Eerde, W. and Rutte, C.G. (2011) A double-edged sword: The effects of challenge and hindrance time pressure on new product development teams. *IEEE Transactions on Engineering Management*, **58**, 1, 71-86.
- Cohen, J., Cohen, P., West, S.G., & Aiken, L.S. (2003) *Applied multiple regression/correlation analysis for the behavioral sciences*. Mahwah, NJ: Lawrence Erlbaum.

- Colomo-Palacios, R., Fernandes, E., Soto-Acosta, P. and Larrucea, X. (2018) A case analysis of enabling continuous software deployment through knowledge management. *International Journal of Information Management*, **40**, 186-189.
- Cooper, R.G. and Edgett, S.J. (2007) *Generating Breakthrough New Product Ideas: Feeding the Innovation Funnel*. Product Development Institute.
- Cooper, R.G., (2017). Idea-to-Launch Gating Systems: Better, Faster, and More Agile. *Research Technology Management*, 60(1), pp. 48-52.
- Daskalantonakis, M K. (1992) A Practical View of Software Measurement and Implementation Experiences Within Motorola. *IEEE Transactions on Software Engineering*, **18**, 11, 998-1010.
- De Souza, E.F., Falbo, R., De A. and Vijaykumar, N.L. (2015) Knowledge management initiatives in software testing: A mapping study. *Information and Software Technology*, **57**, 378–391.
- De Vasconcelos, J.B., Kimble, C., Carreteiro, P. and Rocha, A. (2017) The application of knowledge management to software evolution. *International Journal of Information Management*, **37**, 1499-1506.
- Delisle, J. (2019) Uncovering temporal underpinnings of project management standards. *International Journal of Project Management*, **37**, 8, 968-978.
- Deloitte (2019) Software really is "eating the (tech) world". Retrieved 24/5/2020, from <https://www2.deloitte.com/us/en/pages/technology-media-and-telecommunications/articles/software-growth-in-tech.html>
- Dönmez, D. and Grote, G. (2018). Two sides of the same coin – how agile software development teams approach uncertainty as threats and opportunities. *Information and Software Technology*, **93**, 94-111.

- Drury-Grogan and M.L. (2014) Performance on agile teams: Relating iteration objectives and critical decisions to project management success factors. *Information and Software Technology*, **56**, 5, 506-515.
- Frank, A.G., Ribeiro, J.L.D. and Echeveste, M.E. (2015) Factors influencing knowledge transfer between NPD teams: a taxonomic analysis based on a sociotechnical approach. *R&D Management*, **45**, 1, 1-22.
- Galbraith, J. 1973. *Designing complex organisations*. Boston, MA, Addison-Wesley Longman Publishing Co.
- Garousi, V., Felderer, M., Karapıçak, C.M. and Yılmaz, U. (2018) Testing embedded software: a survey of the literature. *Information and Software Technology*, **104**, pp.14-45.
- Gemünden, H.G., Lehner, P. and Kock, A. (2018) The project-oriented organization and its contribution to innovation. *International Journal of Project Management*, **36**, 1, 147-160.
- Ghobadi, S. (2015) What drives knowledge sharing in software development teams: A literature review and classification framework. *Information and Management*, **52**, 1, 82-97.
- Godener, A. and Söderquist, K.E. (2004) Use and impact of performance measurement results in R&D and NPD an exploratory study. *R&D Management*, **34**, 2, 191-219.
- Goffin, K. and Koners, U. (2011) Tacit Knowledge, Lessons Learnt, and New Product Development. *Journal of Product Innovation Management*, **28**, 2, 300-318.
- Goldoni, V. and Oliveira, M. (2010) Knowledge management metrics in software development companies in Brazil. *Journal of Knowledge Management*, **14**, 2, 301-313.
- Gopal, A., Krishnan, M.S., Mukhopadhyay, T. and Goldenson, D.R. (2002) Measurement Programs in Software Development: Determinants of Success. *IEEE Transactions on Engineering Management*, **28**, 9, 863-875.

- Greene, G.J., Esterhuizen, M. and Fischer, B. (2017) Visualizing and exploring software version control repositories using interactive tag clouds over formal concept lattices. *Information and Software Technology*, **87**, 223-241.
- Gregor, S. and Hevner, A.R. (2013) Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, **37**, 2, 337-355.
- Hajri, I., Goknil, A., Briand, L.C. and Stephany, T. (2018) Change impact analysis for evolving configuration decisions in product line use case models. *Journal of Systems and Software*, **39**, 211-37.
- Hassan, I.U., Ahmad, N. and Zuhaira, B. (2018) Calculating completeness of software project scope definition. *Information and Software Technology*, **94**, 208-233.
- Hevner, A.R.H., March, S.T., Park, J. and Ram, S. (2004) Design Science in Information Systems Research. *MIS Quarterly*, **28**, 1, 75-105.
- Hird, A., Mendibil, K., Duffy, A. and Whitfield, R. I. (2016) New product development resource forecasting. *R&D Management*, **46**, 5, 857–871.
- Hodgson, D. and Cicmil, S. (2007) The politics of standards in modern management: Making 'The project' a reality. *Journal of Management Studies*, **44**, 3, 431-p450.
- Houston, D. (2004) The Value of a Good Checklist. *Software Quality Professional*, **6**, 2, 17-26.
- Hu, Y., Du, J., Zhang, X., Hao, X., Ngai, E.W.T., Fan, M. and Liu, M. (2013) An integrative framework for intelligent software project risk planning. *Decision Support Systems*, **55**, 4, 927-937.
- Hwang, Y., Lin, H. and Shin, D. (2018) Knowledge system commitment and knowledge sharing intention: The role of personal information management motivation. *International Journal of Information Management*, **39**, 220-227.
- Ingram, P. and Simons, T. (2002) The transfer of experience in groups of organisations: Implications for performance and competition. *Management Science*, **48**, 12, 1517-1533.

- Iskoujina, Z. and Roberts, J. (2015) Knowledge sharing in open source software communities: Motivations and management. *Journal of Knowledge Management*, **19**, 4, 791-813.
- Jaakkola, E. and Hallin, A. (2018) Organizational Structures for New Service Development. *Journal of Product Innovation Management*, **35**, 2, 280-297.
- Kautz, K. and Thaysen, K. (2001) Knowledge, learning and IT support in a small software company. *Journal of Knowledge Management*, **5**, 4, 349-357.
- Keil, M., Li, L., Mathiassen, L. and Zheng, G. (2008) The influence of checklists and roles on software practitioner risk perception and decision-making. *Journal of Systems and Software*, **81**, 6, 908–919.
- Khedhaouria, A., Montani, F. and Thurik, R. (2017) Time pressure and team member creativity within R&D projects: The role of learning orientation and knowledge sourcing. *International Journal of Project Management*, **35**, 6, 942-954.
- Khuong, L., Harindranath, G., Dyerson, R. (2014) Understanding knowledge management software-organisation misalignments from an institutional perspective: A case study of a global IT-management consultancy firm. *International Journal of Information Management*, **34**, 2, 226-247.
- Killeen, P.R. and Fetterman, J.G. (1988) A Behavioral Theory of Timing. *Psychological Review*, **95**, 2, 274-95.
- Kleinschmidt, E.J., De Brentani, U. and Salomo, S. (2007) Performance of Global New Product Development Programs: A Resource-Based View. *Journal of Product Innovation Management*, **24**, 5, 419-441.
- Koners, U. and Goffin, K. (2007) Learning from Post-Project Reviews: A Cross-Case Analysis. *Journal of Product Innovation Management*, **24**, 3, 242-258.

- Kyriazis, E., Massey, G., Couchman, P. and Johnson, L. (2017) Friend or foe? The effects of managerial politics on NPD team communication, collaboration and project success. *R&D Management*, **47**, 1, 61-74.
- Lechler, T. G. and Dvir, D. (2010) An alternative taxonomy of project management structures: linking project management structures and project success. *IEEE Transactions on Engineering Management*, **57**, 2, 198-210.
- Lehtinen, T.O.A., Mäntylä, M.V., Vanhanen, J., Itkonen, J. and Lassenius, C. (2014) Perceived causes of software project failures - an analysis of their relationships. *Information and Software Technology*, **56**, 6, 623-43.
- Levitt, B. and March, J.G. (1988) Organisational learning. *Annual Review of Sociology*, **14**, 319–340.
- Lewin, A.Y., Massini, S. and Peeters, C. (2011) Microfoundations of Internal and External Absorptive Capacity Routines. *Organization Science*, **22**, 1, 81-98.
- Li, Y., Liao, X.W. and Lei, H.Z. (2006) A Knowledge Management System for ERP Implementation. *Systems Research and Behavioral Science*, **23**, 2, 157-168.
- Liberatore, M.J. and Pollack-Johnson, B. (2013) Improving Project Management Decision Making by Modeling Quality, Time, and Cost Continuously. *IEEE Transactions on Engineering Management*, **60**, 3, 518-528.
- Lind, D. A., Marchal, W. G., and Wathen, S. A. (2006) Basic statistics for business & economics. Boston: McGraw-Hill/Irwin.
- Markham, S.K. and Lee, H. (2013) Product development and management association's 2012 comparative performance assessment study. *Journal of Product Innovation Management*, **30**, 3, 408-429.

- Marsh, S.J. and Stock, G.N. (2006) Creating Dynamic Capability: The Role of Intertemporal Integration, Knowledge Retention, and Interpretation. *Journal of Product Innovation Management*, **23**, 5, 422-436.
- McKinsey & Company (2019) *Automotive software and electronics 2030: Mapping the sector's future landscape*. Retrieved 24/5/2020, from <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/mapping-the-automotive-software-and-electronics-landscape-through-2030>
- Mitchell, S.M. and Seaman, C.B. (2016) Could removal of project-level knowledge flow obstacles contribute to software process improvement? A study of software engineer perceptions. *Information and Software Technology*, **72**, 151-170.
- Nohria, N. and Gulati, R. (1996) Is slack good or bad for innovation? *Academy of Management Journal*, **39**, 5, 1245-1264.
- Nonaka, I. and von Krogh, G. (2009) Tacit Knowledge and Knowledge Conversion: Controversy and Advancement in Organisational Knowledge Creation Theory. *Organisation Science*, **20**, 3, 635-652.
- Olaisen, J. and Revang, O. (2017) Working smarter and greener: Collaborative knowledge sharing in virtual global project teams. *International Journal of Information Management*, **37**, 1a, 1441-1448.
- Olesen, O.B. and Petersen, N.C. (2016) Stochastic Data Envelopment Analysis—A review. *European Journal of Operational Research*, **251**, 1, 2-21.
- Papke-Shields, K.E., Beise, C. and Quan, J. (2010) Do project managers practice what they preach, and does it matter to project success? *International Journal of Project Management*, **28**, 7, 650–662.

- Park, J.-G and Lee, J. (2014) Knowledge sharing in information systems development projects: Explicating the role of dependence and trust. *International Journal of Project Management*, 32, 1, 153-165.
- Peeters, T. and Martin, X. (2017) Strategies for knowledge use in R&D and their implications for innovative performance. *R&D Management*, 47, 1, 47-60.
- Pemartin, M., Rodriguez-Escudero, A.I. and Munuera-Aleman, J.L. (2018). Effects of collaborative communication on NPD collaboration results: two routes of influence. *Journal of Product Innovation Management*, 35, 2, 184-208.
- PMI (2017) *A Guide to the Project Management Body of Knowledge (PMBOK Guide) – 6th ed.*, Project Management Institute, Philadelphia, PA.
- PMI (2020). Research Highlights by Industry and Region. Pulse of the Profession | Project Management Institute - PMI. Retrieved 24/5/2020 from <https://www.pmi.org/learning/thought-leadership/pulse>
- Prencipe, A. and Tell, F. (2001) Inter-project learning: processes and outcomes of knowledge codification in project-based firms. *Research Policy*, 30, 9, 1373-1394.
- PwC (2018) *What the Top Innovators Get Right. 2018 Global Innovation 1000 Study*. Retrieved 24/5/2020 from <https://www.strategyand.pwc.com/gx/en/insights/innovation1000/2018-global-innovation-1000-fact-pack.pdf>
- Rahmani, H., Sami, A. and Khalili, A. (2016) CIP-UQIM: A unified model for quality improvement in software SME's based on CMMI level 2 and 3. *Information and Software Technology*, 71, 27-57.
- Rashid, M., Clarke, P.M. and O'Connor, R.V. (2019) A systematic examination of knowledge loss in open source software projects. *International Journal of Information Management*, 46, 104-123.

- Rattat, A., Matha, P. and Cegarra, J. (2018) Time flies faster under time pressure. *Acta Psychologica*, **185**, 81-86.
- Rees-Caldwell, K. and Pinnington, A.H. (2013) National culture differences in project management: Comparing British and Arab project managers' perceptions of different planning areas. *International Journal of Project Management*, *31*, 2, 212-227.
- Ren, X., Deng, X. and Liang, L. (2018) Knowledge transfer between projects within project-based organizations: the project nature perspective. *Journal of Knowledge Management*, **22**, 5, 1082-1103.
- Richtnér, A., Åhlstrom, P. and Goffin, K. (2014) Squeezing R&D': A Study of Organizational Slack and Knowledge Creation in NPD, Using the SECI Model. *Journal of Product Innovation Management*, **31**, 6, 1268-1290.
- Riek, R.F. (2001) From experience: Capturing hard-won NPD lessons in checklists. *Journal of Product Innovation Management*, **18**, 5, 301-313.
- Rigby, D. and Bilodeau, B. (2018) *Management Tools & Trends*. Bain & Company. Retrieved 24/5/2020, from <https://bit.ly/2HxGTZw>
- Rossmann, G.B. and Rallis, S.F. (2003) *Learning in the Field: An Introduction to Qualitative Research (2nd ed.)*, Sage Publications, Thousand Oaks.
- Ruuska, I. and Brady, T. (2011) Implementing the replication strategy in uncertain and complex investment projects. *International Journal of Project Management*, **29**, 4, 422–431.
- Ryan, S. and O'Connor, R.V. (2013) Acquiring and sharing tacit knowledge in software development teams: An empirical study. *Information and Software Technology*, **55**, 9, 1614-1624.
- Salomo, S., Weise, J. and Gemünden, H.G. (2007) NPD Planning Activities and Innovation Performance: The Mediating Role of Process Management and the Moderating Effect of Product Innovativeness. *Journal of Product Innovation Management*, **24**, 4, 285-302.

- Segelod, E. and Jordan, G. (2004) The use and importance of external sources of knowledge in the software development process. *R&D Management*, **34**, 3, 239-252.
- Serna M., E., Bachiller S., O. and Serna A., A. (2017) Knowledge meaning and management in requirements engineering. *International Journal of Information Management*, **37**, 3, 155-161.
- Serrador, P. and Turner, R. (2015) What is Enough Planning? Results from a Global Quantitative Study. *IEEE Transactions on Engineering Management*, **62**, 4, 462-474.
- Shanling, L., Shang, J. and Slaughter, S.A. (2010) Why Do Software Firms Fail? Capabilities, Competitive Actions, and Firm Survival in the Software Industry from 1995 to 2007. *Information Systems Research*, **21**, 3, 631-654.
- Shenhar, A. and Dvir D. (2007) *Reinventing project management: the diamond approach to successful growth and innovation*. Harvard Business School Press. Boston, MA.
- Slaughter, S.A. and Kirsch, L.J. (2006) The Effectiveness of Knowledge Transfer Portfolios in Software Process Improvement: A Field Study. *Information Systems Research*, **17**, 3, 301-320.
- Smets, L.P.M., Langerak, F. and Tatikonda, M.V. (2016) Collaboration between competitors' NPD teams: In search of effective modes of management control. *R&D Management*, **46**, 244-260.
- Söderquist, K.E. (2006) Organising Knowledge Management and Dissemination in New Product Development: Lessons from 12 Global Corporations. *Long Range Planning*, **39**, 5, 497-523.
- Standish Group (2015) *CHAOS Report 2015*. Retrieved 24/5/2020, from www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf
- Stigliani, I. and Ravasi, D. (2012) Organizing thoughts and connecting brains: Material practices and the transition from individual to group-level prospective sense-making. *Academy of Management Review*, **55**, 1232–1259.

- Sutherland, J. (2014) *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown Pub Inc. New York. ISBN. 978-0-385-34645-0.
- Symons, C. (2010) Software Industry Performance: What You Measure Is What You Get. *IEEE Software*, **27**, 6, 66-72.
- Tyagi, S., Cai, X., Yang, K. and Chambers, T. (2015) Lean tools and methods to support efficient knowledge creation. *International Journal of Information Management*, **35**, 2, 204-214.
- Um, K.-H. and Kim, S.-M. (2018) Collaboration and opportunism as mediators of the relationship between NPD project uncertainty and NPD project performance. *International Journal of Project Management*, **36**, 4, 659-672.
- Van Aken, J.E. (2004) Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules. *Journal of Management Studies*, **41**, 2, 219-246.
- Van Berkel, F.J.F.W., Ferguson, J.E. and Groenewegen, P. (2016) Speedy Delivery Versus Long-term Objectives: How Time Pressure Affects Coordination Between Temporary Projects and Permanent Organizations. *Long Range Planning*, **49**, 6, 661-673.
- Van Oorschot, K., Eling, K. and Langerak, F. (2018) Measuring the Knowns to Manage the Unknown: How to Choose the Gate Timing Strategy in NPD Projects. *Journal of Product Innovation Management*, **35**, 2, 164-183.
- Van Oorschot, K., Sengupta, K., Akkermans, H. and Van Wassenhove, L. (2010) Get Fat Fast Surviving Stage-Gate® in NPD. *Journal of Product Innovation Management*, **27**, 6, 828-839.
- Wiengarten, F., Fan, D., Lo, C. K. and Pagell, M. (2017) The differing impacts of operational and financial slack on occupational safety in varying market conditions. *Journal of Operations Management*, **52**, 30-45.

- Yang, Y., Brosch, G., Yang, B. and Cadden, T. (2020) Dissemination and communication of lessons learned for a project-based business with the application of information technology: a case study with Siemens. *Production Planning and Control*, **31**, 4, 273-p286.
- Yigitcanlar, T., Sabatini-Marques, J., Kamruzzaman, Md., Camargo, F., da-Costa, E.M., Ioppolo, G. and Palandi, F.E.D. (2018) Impact of funding sources on innovation: evidence from Brazilian software companies. *R&D Management*, **48**, 4, 460-484.
- Zhao, D., Zuo, M. and Deng, X. (2015) Examining the factors influencing cross-project knowledge transfer: An empirical study of IT services firms in China. *International Journal of Project Management*, **33**, 325–340.
- Zwikael, O. and Globerson S. (2004) Evaluating the quality of project planning: a model and field results. *International Journal of Production Research*, **42**, 8, 1545–1556.
- Zwikael, O. and Sadeh, A. (2007) Planning effort as an effective risk management tool. *Journal of Operations Management*, **25**, 4, 755-767.
- Zwikael, O., Pathak, R.D., Singh, G. and Ahmed, S. (2014) The moderating effect of risk on the relationship between planning and success. *International Journal of Project Management*, **32**, 3, 435-441.

Appendix. Questionnaires

Variables		Items	Reference
IV	PBL	The organisation has past experience with similar projects	Ruuska and Brady, 2011
		There is historical data that can be used for the development of the project	Yang et al., 2020; Banker and Hwang, 2008
MV1	Project uncertainty	This is a high-risk project	Zwikael and Sadeh, 2007
MV2	Team collaboration	There is collaboration between project team members	Olaisen and Revang, 2017; Frank et al., 2015
MV3	Project-based structure	Type of organisation structure	Belout and Gauvreau, 2004
MV4	Time pressure	The time pressure on the project is high	Khedhaouria et al., 2017
CV1	Concluded project		-
CV2	Manager's work experience		-
CV3	Product novelty	How new is the product to customers and users	Shenhar and Dvir, 2007
CV4	Technology novelty	How much new technology is used	Shenhar and Dvir, 2007
CV5	Organisation culture	Culture in the organisation is too political	Kyriazis et al., 2017
DMY	Strategic goal	Extension, Strategic, Problem solving, Maintenance, Research	Shenhar and Dvir, 2007
DV Planning quality		For each planning product written, please mark the most suitable answer referring to the projects you were recently involved in, according to the following scale:	Zwikael and Globerson, 2004
		- Project plan	
		- Project deliverables	
		- Work Breakdown Structure chart	
		- List of project activities	
		- PERT or Gantt chart	
		- Activity required resources	
		- Activity duration estimates	
		- Activity start and end dates	
		- Resource cost	
		- Time-phased budget	
		- Quality management plan	
		- Role and responsibility assignments	
		- Project staff assignments	
		- Communications management plan	
		- Risk management plan	
		- Procurement management plan	

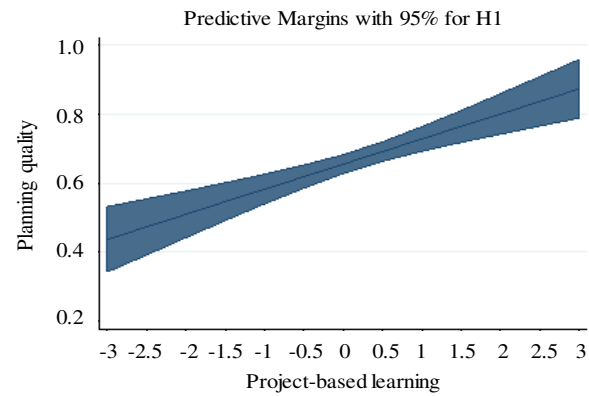


Figure 1: Marginal effect of planning quality (DV) as a function of PBL (IV)

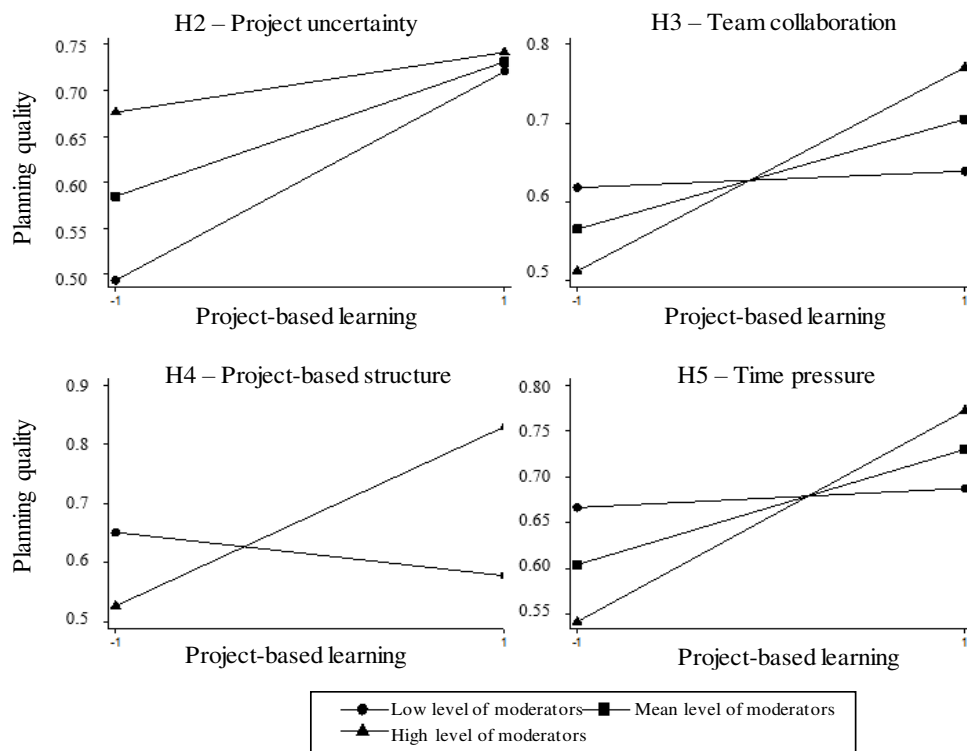


Figure 2: Moderating effects

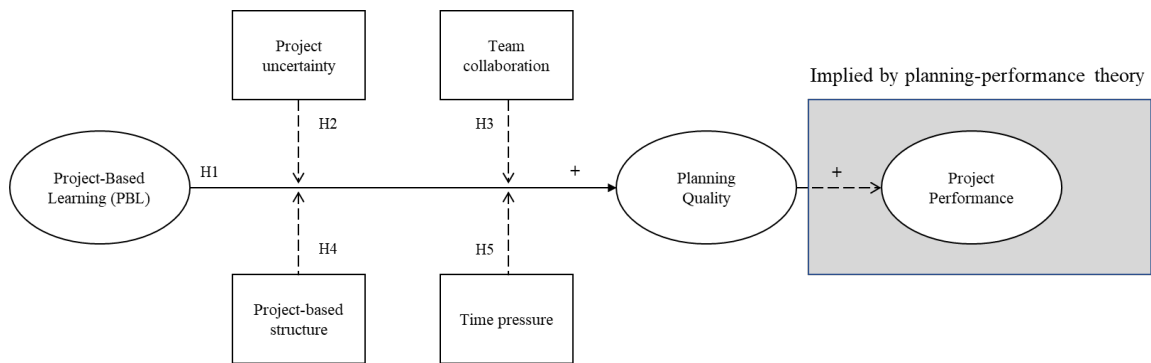


Figure 3: Tentative conceptual model based on the results

Table 1: Models that evaluate planning quality

#	Model	Evaluation Mechanism	Measurement scales	Method	Sample	References
1	Quality of Planning by Manager (QPM)	Quality of 16 planning outputs from 16 core planning processes defined in the PMBOK	16 factors, five-point Likert scale	Survey, PMs from nine organisations	282 projects	Zwikaël and Globerson, 2004;
2	Many-to-many actionable knowledge discovery (MMAKD)	Quality of risk planning through the database of the organisation	27 factors, input to a Random forest algorithm ¹	Case study, social media platform	269 answers	Hu et al., 2013
3	Unified Quality Improvement Model (UQIM)	Quality of software products and services (integrates concepts from CMMI-DEV, ISO 9001 and PMBOK)	Conceptual model only	N/A	None	Rahmani et al., 2016
4	Quality of software planning (QPLAN)	Factors that impact the planning of software development projects	55 factors, organised in 21 cognitive maps (bottom-up approach) and 16 factors used by QPM (top-down approach), five-point Likert scale	Case study, large defence organisation	20 projects	Amaral Féris et al., 2017; Amaral Féris and Zwikaël, 2017
5	Software project scope rating index (SPSRI)	Completeness of software project scope definition	Four-point scale	Survey, PMs, system analysts, team leaders and developers	20 projects	Hassan et al., 2018

Table 2: Sample profile

Organisation	Industry Type	Brazil	US	Israel	Total
A	Industrial Automation	8			8
B	Printing	13	1		14
C	Defence	21		4	25
Total		42	1	4	47

¹ “Random forest” is a machine learning method that performs both regression and classification tasks by using multiple decision trees (Breiman, 2001)

Table 3: Sixteen outputs of project planning defined in PMBOK (Zwikael and Globerson, 2004)

	Planning Process	Outputs from Planning Process ("Planning Products")
1	Develop project management plan	Identification of actions necessary to define, integrate and coordinate all subsidiary plans
2	Define scope	Detailed description of the project and product
3	Create work breakdown structure	Subdivides project deliverables and project work into more manageable components
4	Define activities	Identifications of the specific actions to be performed to produce the project deliverables
5	Sequence activities	Identification of the relationships among activities
6	Estimate activity resources	Estimation of material/people/equipment/supplies required to perform each activity
7	Estimate activity durations	Definition of the number of work periods needed to complete each activity
8	Develop schedule	Gantt chart, which shows activity sequences, durations, requirements and constraints
9	Estimate costs	Estimation of the monetary resources needed to complete project activities
10	Determine budget	Authorised cost baseline from the aggregation of costs estimation of individual activities
11	Plan quality	Definition of quality requirements and how the project will demonstrate compliance
12	Develop human resource plan	Identification of roles, responsibilities, their relationship and required skills
13	Acquire project team	Confirmation of human resources (HR) availability to complete project assignments
14	Plan communications	Identification of information needs and definition of a communication approach
15	Plan risk management	Definition of how to conduct risk management activities
16	Plan procurements	Documenting purchasing decisions, approach to be adopted, and potential sellers

Table 4: Descriptive statistics and correlations

Variables	Mean	Std	DV	IV	MV1	MV2	MV3	MV4	CV1	CV2	CV3	CV4
DV Planning quality	0.65	0.14										
IV PBL	0.00	1.00	0.52									
MV1 Project uncertainty	0.00	1.00	0.30	0.08								
MV2 Team collaboration	0.00	1.00	0.26	0.34	0.18							
MV3 Project-based structure	0.28	0.45	0.02	-0.01	-0.04	0.22						
MV4 Time pressure	0.00	1.00	-0.29	-0.24	-0.02	0.14	0.10					
CV1 Concluded project	0.49	0.51	-0.20	-0.30	-0.33	-0.32	-0.13	0.02				
CV2 Manager's work experience	2.51	0.86	0.02	0.11	-0.13	-0.12	-0.20	0.03	0.15			
CV3 Product novelty	0.23	0.43	0.02	-0.25	-0.22	-0.02	0.00	0.07	0.26	0.12		
CV4 Technology novelty	2.32	0.81	-0.15	-0.23	-0.48	0.05	0.35	0.08	0.19	0.04	0.41	
CV5 Organisation culture	0.49	0.25	-0.10	-0.04	-0.44	-0.19	0.02	-0.27	0.34	0.12	0.30	0.38

Note: DV =Dependent variable; IV = Independent variable; MV = Moderating variable; CV = Control variable

Table 5: Regression analysis of planning quality (direct effect)

		Model 1 (CV)			Model 2 (CV + IV)		
Variables		Coef.	Std Err	p-value	Coef.	Std Err	p-value
MV1	Project uncertainty	0.035	0.025	0.157	0.042	0.025	0.089
MV2	Team collaboration	0.032	0.020	0.103	-0.006	0.017	0.730
MV3	Project-based structure	0.021	0.047	0.653	0.025	0.033	0.441
MV4	Tightly coupled projects	-0.039	0.017	0.017	-0.019	0.015	0.208
CV1	Concluded project	-0.006	0.041	0.881	0.022	0.032	0.493
CV2	Manager's work experience	-0.006	0.030	0.833	0.006	0.024	0.800
CV3	Product novelty	0.060	0.054	0.267	0.050	0.046	0.275
CV4	Technology novelty	0.007	0.028	0.796	0.020	0.022	0.372
CV5	Organisation culture	-0.125	0.089	0.162	-0.104	0.069	0.132
	Intercept	0.698	0.097	0.000	0.637	0.087	0.000
IV	PBL				0.073	0.015	0.000
DMY	Organisation dummies	Included			Included		
	Strategic goal dummies	Included			Included		
	Countries	Included			Included		
	Log-likelihood	38.061		0.000	44.352		0.000
	AIC	-0.981			-1.206		
Note: robust standard error; n = 47 projects							

Note: DV =Dependent variable; IV = Independent variable; MV = Moderating variable; CV = Control variable

Table 6: Regression analysis of planning quality (moderating effects)

		Model 3			Model 4			Model 5			Model 6		
		Coef.	Std Err	p-value	Coef.	Std Err	p-value	Coef.	Std Err	p-value	Coef.	Std Err	p-value
MV1	Project uncertainty	0.051	0.020	0.012	0.016	0.018	0.363	0.038	0.022	0.083	0.018	0.021	0.403
MV2	Team collaboration	0.006	0.015	0.704	0.007	0.015	0.668	-0.003	0.018	0.886	-0.006	0.017	0.697
MV3	Project-based structure	0.028	0.032	0.377	0.022	0.029	0.461	0.032	0.032	0.320	0.013	0.033	0.707
MV4	Time pressure	-0.021	0.014	0.126	-0.018	0.016	0.261	-0.020	0.014	0.157	-0.010	0.015	0.501
CV1	Concluded project	0.032	0.031	0.316	0.021	0.028	0.460	0.018	0.032	0.567	0.006	0.030	0.848
CV2	Manager's work experience	0.015	0.021	0.481	-0.012	0.022	0.594	0.007	0.023	0.769	-0.008	0.024	0.731
CV3	Product novelty	0.048	0.050	0.339	0.057	0.040	0.151	0.036	0.043	0.400	0.110	0.056	0.050
CV4	Technology novelty	0.016	0.021	0.438	0.022	0.020	0.272	0.011	0.019	0.569	0.018	0.020	0.365
CV5	Organisation culture	-0.129	0.064	0.043	-0.190	0.081	0.019	-0.097	0.064	0.129	-0.188	0.081	0.020
	Intercept	0.623	0.075	0.000	0.681	0.077	0.000	0.634	0.078	0.000	0.802	0.105	0.000
H1	PBL	0.073	0.015	0.000	0.070	0.013	0.000	0.057	0.016	0.000	0.063	0.016	0.000
H2	PBL x project uncertainty	-0.041	0.016	0.013									
H3	PBL x team collaboration				0.060	0.021	0.004						
H4	PBL x project-based structure							0.094	0.036	0.009			
H5	PBL x time pressure										0.052	0.023	0.026
DMY	Organisation dummies	Included			Included			Included			Included		
	Strategic goal dummies	Included			Included			Included			Included		
	Country dummies	Included			Included			Included			Included		
	Log-likelihood	47.440			49.457			46.143			46.866		
	AIC	-1.295			-1.381			-1.240			-1.271		

Note: robust standard error; n=47 projects

Table 7: Summary of findings

Research Question	<i>Does project-based learning enhance the quality of planning in subsequent software development projects?</i>	
Hypothesis	Finding:	Practical ways this can be achieved
H1	PBL enhances planning quality of subsequent software development projects	Use a knowledge management system for supporting transfer knowledge acquired in each project into organisation's memory and thus, increase planning quality of subsequent software development projects.
H2	PBL is more effective when project uncertainty is high	Build an organisational environment that enables team members in sharing ideas and proactively search the knowledge from past projects.
H3	PBL is more effective when the team collaboration is high	Build an organisational environment focused on face-to-face communication, with regular and direct interactions aimed at increasing trust and commitment among them.
H4	PBL is more effective in a project-based structure	Same recommendation as above.
H5	PBL is more effective under high time pressure	In projects with high visibility levels that allow personal development and future gains.