

QPLAN: Decision Support for Evaluating Planning Quality in Software Development ProjectsMarco Antônio Amaral Férís ^{a,*}, Ofer Zwikael ^b, Shirley Gregor ^b^a Cranfield School of Management, Cranfield University, Cranfield, United Kingdom^b Research School of Management, The Australian National University, Canberra, Australia

* Corresponding author

E-mail addresses: M.Amaral-Feris@cranfield.ac.uk (M. A. Amaral Feris), ofer.zwikael@anu.edu.au (O. Zwikael), shirley.gregor@anu.edu.au (S. Gregor).

Abstract

Decisions about whether or not to approve a project plan for execution are critical. A decision to continue with a bad plan may lead to a failed project, whereas requesting unnecessary additional planning for an already high-quality plan may be counterproductive. However, these decisions can be influenced by psychological biases, such as the endowment effect, optimism bias and ambiguity effect, which are enhanced when uncertainty is substantial and information incomplete. As a result, a non-biased model for evaluating the quality of project planning is important to improve planning approval decisions and resource allocation. This paper introduces a novel artifact (QPLAN) that evaluates and improves planning quality, and a case study to demonstrate its effectiveness within a business environment.

Keywords: Decision support tool; Dual process theory; Planning quality; Software projects; Planning-performance theory.

QPLAN: Decision Support for Evaluating Planning Quality in Software Development Projects

1. Introduction

The information technology (IT) industry, which encompasses enterprise software, data centers, devices, IT services and telecom services, was predicted to expend US\$3.5 trillion in 2017 [19]. Nonetheless, poor performance of software development projects has plagued the IT industry for years [33]. In 2013, for instance, organizations successfully completed only 39 percent of software projects [64], which led to estimated annual losses for the United States (US) and European Union (EU) markets of around US\$100 billion each [67].

Research has identified planning quality¹ as a key area for improvement in overcoming software projects' poor performance [13,21,66,74]. Similarly, planning-performance theory suggests a positive impact of planning formality on financial performance [5]. Among other advantages, planning allows managers to better understand project requirements [21] and business contexts [15], reduces projects' inherent uncertainty, and provides a reliable basis for monitoring and controlling projects [72]. Planning involves comparing alternative courses of actions [4,8], and establishing a pathway to accomplish the project's goals [59]. Because planning precedes major financial expenditures in a project, and because a small group of people is typically employed during this stage, improving a project plan, if needed, has a relatively low cost compared with its high value.

However, managers can find it difficult to determine if a plan is of sufficient quality as a basis to start software development or if it requires further improvements. In particular, managers are required to make an important decision at the end of the planning phase: whether or not to approve the plan as is, re-

¹ "Planning quality" is equivalent to "quality of planning" and refers to a project's planning stage, which a project manager leads and commences immediately after project approval. During the planning stage, the project manager develops a project plan that should be aligned with the business case and provide a clear direction for the work to be done during project execution. Planning quality measures the quality of the planning processes undertaken by the project manager and influences directly the quality of the project plan document.

work and improve it, or terminate the project before the organization invests more resources in it. An incorrect decision can cause managers to: (1) continue with a bad plan, which increases the chance of a failed project; (2) invest unnecessary additional resources in an already high-quality plan; and (3) not terminate bad projects on time.

Because the level of uncertainty is at its peak at the time of this decision [73], psychological biases can often prevent managers from making correct decisions regarding project plans. For example, an “endowment effect” causes managers to feel ownership of a project, hence valuing it more than its real worth [30], or defining requirements beyond the actual needs of the customer or the market [61]. Moreover, managers can have a tendency to overestimate the quality of project plans or be too optimistic about future projects’ performance (“optimism bias”) [29,69], hence approving project plans that are of questionable quality. Finally, an “ambiguity effect” supports a tendency to avoid decision making due to missing information [29], and managers can avoid making a decision on project termination when they should do so.

These psychological biases have a negative impact on decision making [44], especially under high levels of uncertainty when information is incomplete and ambiguous [29,35,46]. These biases are intensified when decision makers rely on their personal knowledge, known as an “inside view” [29,69]. In an “inside view”, decision makers build overly optimistic scenarios of events that may affect the project [17]. Alternatively, an “outside view” is based on previous actual objectives and comparable data from similar actions already completed. An “outside view” is more likely to produce a quality decision because it bypasses cognitive biases. In an “outside view”, decision makers consider similar past projects to forecast a project’s future [17,29].

Accordingly, dual process theory aims to explain the decision-making process from the view of decision makers through a general framework comprising two distinct systems of thinking: System 1 and System 2 [56]. System 1 corresponds to intuitive thinking, which is emotional, fast and unconscious thinking; System 2 corresponds to reasoned thinking, which is logical, controlled operations, slow and

conscious [29,43]. Systems 1 and 2 operate in parallel and interactively [56]. In case of high uncertainty levels, System 2 can monitor the information quality provided by System 1 by framing it to be subservient to System 2 [29], i.e., System 1 provides inputs to System 2, which rationalize the reasons why a particular judgment was made. Poor project plan decisions are more likely when only System 1 is in use, due to the psychological biases discussed above [43]. In adding System 2, the evaluation of a project plan is also based on relevant, reliable and quantitative information. As a result, introducing System 2 to decision making under uncertainty can reduce bias and improve decision quality.

Therefore, organizations can improve their decision making processes by using an objective tool that evaluates planning quality independently, thus encouraging System 2 thinking. The evaluation literature provides support for this argument by stating that: 1) better evaluation improves the quality of decision makers' thinking [16,42,50], and 2) evaluation requires measurement and learning [20]. Thus, we address the following research question: "How can the planning quality of software development projects be better evaluated and improved?"

To address this question, we introduce the quality of planning (QPLAN) tool, so named because it evaluates how well one plans software-development projects. QPLAN is an innovative decision support tool that helps managers make better decisions by reducing psychological biases and empirically evaluating and enhancing the quality of project planning.

This paper proceeds as follows: In Section 2, we review the relevant literature related to evaluating planning quality. In Section 3, we discuss our research approach, and in Section 4 describe how we developed, validated and used QPLAN in a case study. In Section 5, we apply QPLAN to an organization from the defense industry as a case study. Finally, in Section 6, we conclude the paper.

2. Existing methods for evaluating planning quality

The literature has identified three leading methods for evaluating planning quality in software-development projects: the project management planning quality (PMPQ) model, checklists, and metrics. We discuss each approach below.

2.1 PMPQ model

The PMPQ model evaluates the quality of project planning through key planning products [74]. The literature has extensively validated and used this model (e.g., Zwikaël and Globerson [75]; Masters and Frazier [41]; Zwikaël and Sadeh [71]; Papke-Shields et al. [47]; Zwikaël and Ahn [73]; Barry and Uys [10]; Rees-Caldwell and Pinnington [51]). The model's overall planning quality indicator comprises two sub-indices: planning quality by the organization (QPO), which evaluates organizational-support processes, and planning quality by a manager (QPM), which evaluates the planning-quality processes for which a project manager is responsible. The model measures planning quality by a weighted linear combination of 16 planning processes that the project management body of knowledge (PMBOK) defines [74] (Table 1).

Table 1

Sixteen core planning processes in the PMPQ model (adapted from Zwikaël and Globerson [74])

| Code | Planning process | Description |
|------|---------------------------------|--|
| 4.2 | Develop project management plan | Document actions necessary to define, prepare, integrate and coordinate all subsidiary plans |
| 5.2 | Define scope | Develop a detailed description of the project and product |
| 5.3 | Create work-breakdown structure | Subdivide project deliverables and project work into smaller, more manageable components |
| 6.1 | Define activities | Identify specific tasks to be performed in the project |
| 6.2 | Sequence activities | Identify and document relationships among activities |
| 6.3 | Estimate activity resources | Estimate type or quantities of material, people, equipment or supplies required to perform each activity |
| 6.4 | Estimate activity durations | Approximate the number of work periods needed to complete each activity |
| 6.5 | Develop schedule | Analyze activity sequences, durations, requirements and constraints to create the schedule |
| 7.1 | Estimate costs | Approximate the monetary resources needed to complete project activities |
| 7.2 | Determine budget | Aggregate the estimated costs of individual activities to establish an authorized cost baseline |
| 8.1 | Plan quality | Identify and document quality requirements and how the project will demonstrate compliance |
| 9.1 | Develop human resource plan | Identify and document roles, responsibilities and required skills and report relationships between them |
| 9.2 | Acquire project team | Confirm human resources availability and obtain the team necessary to complete project assignments |
| 10.2 | Plan communications | Determine project stakeholder information needs and define a communication approach |
| 11.1 | Plan risk management | Define how to conduct risk-management activities |
| 12.1 | Plan procurements | Determine purchasing approach and potential sellers |

2.2 Checklist approaches

Checklists are a common approach to assessing planning quality and risks [32]. Based on expert knowledge [24], software-development projects often use checklists to determine whether the planning phase has finished and the project can proceed with the next phase [32], guide reviews, and ensure project

teams adhere to procedures. Checklists are a common tool that various models and methods use, such as: (1) the capability maturity model integration (CMMI) [62]; (2) Six Sigma, a set of techniques and tools for improving manufacturing [12] and software-development processes [39]; (3) ISO/IEC 15939, a measurement process applicable to system and software engineering and management fields [25]; (4) process improvement training and appraisal program for software development [62]; and (5) the SQuaRE model, a general data-quality model for data retained in a structured format in a computer system [26]. Checklists can incorporate expert knowledge of a process, including lessons learnt from past projects [24].

Checklists provide guidance on crucial questions that one needs to ask and a systematic approach to the various stages involved in planning. Checklists are perhaps the simplest and most productive tool for analyzing quality. However, excessive and uncritical use of checklists can be counterproductive [9,24]).

2.3. Metrics approaches

Metrics have a vital role in software development because of their potential to improve quality and productivity as efficient feedback mechanisms. The rationale for using metrics arises from the notion that one cannot improve something without first measuring it [20]. For this purpose, practitioners collect and analyze metrics relevant to software projects to better manage the software-development process and make necessary changes to increase productivity and quality and, thereby, reduce cycle times and costs in the long run [14]. To be successful, the implementation of a metrics program should have the support of the organization and be easy to use [20]. In addition, practitioners should understand that metrics are not the goal but an important tool that highlights problems and gives ideas as to what the organization can do [14].

2.4. Problems with current methods

Some limitations with the above models and approaches should be discussed. As a generic model, PMPQ does not specifically cater to software-development projects, does not consider the project manager's personal characteristics (e.g. appropriate approach for people management [51]), and does not consider the relationships among planning processes [36,50]. Checklists assume the experts' knowledge of

a process and lessons learned from past projects [24]. For example, checklists can help project managers identify risks, but a non-comprehensive checklist can also identify non-existent risks [32]. Moreover, checklists have not been tested systematically [32]. Metrics have high potential of business value return; however, few organizations succeed in their implementation [14,67] for the following reasons: a) metrics are based only on quantitative data and, thus, do not consider other factors that affect how plans are evaluated, such as pressure from marketing to deliver the product in the shortest timeframe (even with lower quality); b) metrics' implementation is usually a long process; c) practitioners' viewpoints are not always considered, hence lead to potential conflicts [20]; and d) irrelevant metrics are often defined [67]. Furthermore, none of these methods has been reported as aiming to reduce psychological biases that prevent managers from making correct planning decisions. Because of these limitations, we need a new approach to assess planning quality in software-development projects that, if possible, integrates the best of each prior approach and overcomes their limitations.

3. Research approach

We selected the design science research (DSR) as the research method, because of its strength in solving a real problem using applied research [23] in the field of information systems [11]; in our case a tool to improve the planning quality of software-development projects. DSR focuses on knowledge-intensive design [2], helps solve real problems through the development of innovative artifacts [6,11,23], and can be effective in the management field [3]. In addition, DSR is a complementary research approach to behavioral science, in that DSR aims to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts, and behavioral science aims to develop and verify theories that explain or predict human and organizational behavior [23]. We used the design science research process (DSRP) model [48] to develop the new tool. This model is consistent with design science processes in prior studies [22] and provides a general process for conducting DSR. In accordance with Peffers et al. [48], Fig. 1 shows the six DSRP steps, which we discuss in Sections 3.1 to 3.6.

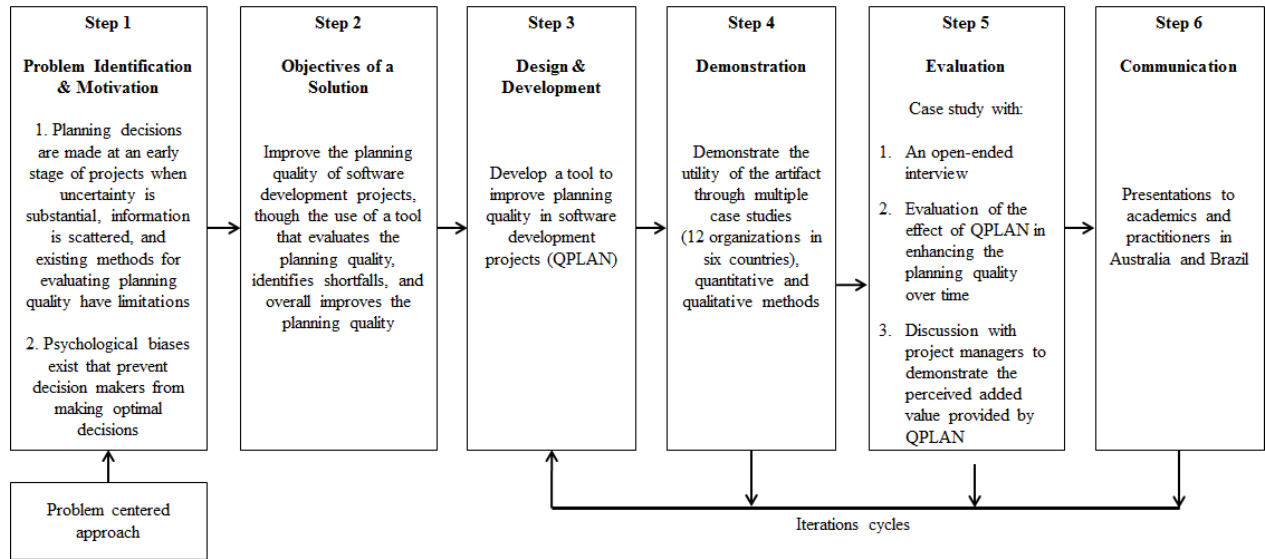


Fig. 1. Design Science Research Process applied to this study (adapted from Peffers et al. [48]).

In Step 1 (Problem Identification and Motivation), one identifies the research problem, defines the scope, and justifies the value of the proposed solution. This step helps one motivate the stakeholders interested in the research and allows them to understand the researcher's reasoning for addressing the problem. In this study we identified that planning quality is a key area for improvement in overcoming software projects' poor performance [13,21,66]. However, it is difficult for managers to make the correct decisions at the early planning stage, when uncertainty is substantial, information is scattered, and some psychological biases are intensified.

In Step 2 (Objectives of a Solution), one defines the objectives of a solution to the problem inferred in the previous step. In this study, we aim to improve decision and planning quality of software-development projects by developing a tool that evaluates planning quality and introduces best planning practices aimed at reducing psychological biases. This objective is motivated by the limitations identified in the existing approaches for planning quality evaluation.

In Step 3 (Design and Development), one designs and develops artifacts to address the research problem. The process involves defining requirements and designing the architecture for developing the

desired artifact, which can be constructs, models, methods, or instantiations [23,40]. We performed white-box and black-box testing to ensure QPLAN's functionality, completeness, and usability [1].

In Step 4 (Demonstration), one demonstrates the artifact's utility [23]. We performed multiple case studies in 12 organizations from eight types of industries (automation, IT, education, R&D, defense, pharmaceutical, logistics and banking) and across six countries (Australia, Brazil, the US, Israel, Germany and Italy) using a variety of quantitative and qualitative methods. We further refined QPLAN while conducting these case studies. The demonstration of QPLAN within the business environment was undertaken concurrently with the design and development (Step 3). Although this approach required us to maintain compatibility with project data already collected, feedback received from research participants allowed us to improve the software throughout the development and validation process, such as: a) QPLAN user interface; b) project reports; and c) organizational knowledge from previous projects. Thus, the development process became an iterative search that helped in finding an effective solution to the problems that motivated this research [23], and supported the essence of DSR [23], which is to develop an artifact that demonstrates its utility to researchers and practitioners.

In Step 5 (Evaluation), one evaluates the outcomes from the use of the artifact. In this paper we report on one of the case studies as a summative evaluation (in a multinational organization from the defense industry that has development centers in Brazil and Israel) to evaluate QPLAN's use and effectiveness. The case study includes: (1) an open-ended interview with the top manager responsible for developing software in the organization [54]; (2) evaluating the effect of QPLAN in enhancing planning quality over time [12]; and (3) discussing the value from using the artifact with project managers at that organization [20].

In Step 6 (Communication), one communicates one's findings. To date, we have presented and discussed QPLAN with a large audience of industry and academia participants (e.g., at a workshop promoted by the Australian and New Zealand Academy of Management, a seminar organized by the

Brazilian chapter of the Project Manager Institute and a seminar by P&D Brazil—a Brazilian association of organizations in the electronics field).

4. The QPLAN tool and approach

A design artifact is complete and effective when it satisfies the requirements and constraints of the problem it was meant to solve (Hevner et al. [23], p.85).

4.1. Overview

This research developed QPLAN, an innovative tool that one can use to evaluate the planning quality of any type of software project and hence support better decisions. As a knowledge management-based decision support system (KMDSS), QPLAN supports decision making by helping one store, process, and retrieve knowledge [7]. QPLAN has four components: (1) a model for evaluating planning quality (giving the QIPan index), (2) an extended Karnaugh map for identifying the strengths and weaknesses of planning [55], (3) a novelty, technology, complexity and pace (NTCP) diamond model of project characteristics [58], and (4) a knowledge base that supports learning from past projects developed by the same organization [20].

4.1.1. Evaluation model

In QPLAN, evaluating planning quality involves two measures: the planning quality of the manager (QPM) and planning quality through cognitive map factors (QCM), as follows: QPM is an existing index used in the PMPQ model to evaluate planning quality by assessing 16 planning products (top-down approach) (Table 1). QCM uses cognitive maps, which is a methodology based on expert knowledge [63] that graphically describes the behavior of a system [53]. According to Stach et al. [63], a single expert can reliably develop cognitive maps; however, because a group of experts usually improves scale reliability, we also had other project managers examine the factors. Moreover, according to Rodriguez-Repiso et al. [53], even if the initial mapping of the factors is incomplete or incorrect, further additions to the map may be included. We developed QCM in this study to evaluate planning quality via

the factors that affect planning processes (a bottom-up approach). The combination of both approaches should help one more accurately evaluate planning [28].

The output of the model is the planning quality (QIPan) index, which one calculates as the average of QPM and QCM and ranges from 0.0 (lowest) to 1.0 (highest) (see Fig. 2). QPLAN reports on all three metrics (QIPan, QPM, and QCM).

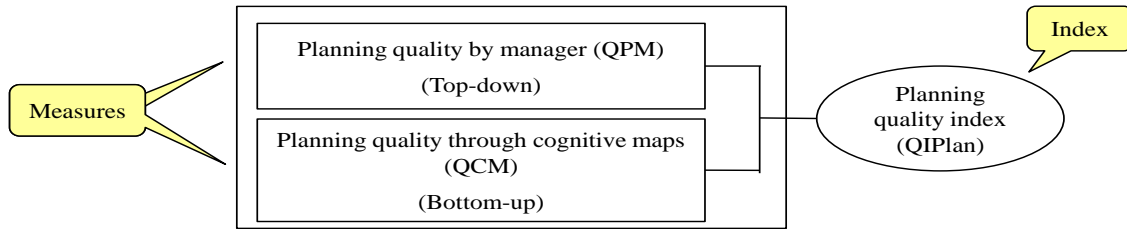


Fig. 2. Model for evaluating planning quality

Whereas QPM is an established measure, we developed QCM based on 55 factors: the tool assesses 23 factors when one begins to plan and the remaining 32 factors when finishing. We used the following process to identify a concise list of generic project management factors and specific software development factors that affect planning. After searching leading project management, general management, and computer science journals (e.g., *IEEE Transactions on Software Engineering* and *Information Systems Journal*) between 1986 and 2012 using the keywords “project success”, “planning”, “project management”, and “software development”, we identified 37 relevant papers and, from them, 211 factors that affect project planning. Example factors included “realistic effort estimates” [18,27,45,70] and “secured funding” [37,68]. Motivated by the fact that many factors to be evaluated by the project managers would cause an additional workload that could derail this study, based on the researchers’ judgment and experience, we reduced the number of factors from 211 to 55.

We then grouped the 55 factors into 21 cognitive maps. The relationship between the cognitive maps and project factors is illustrated in the “Top Management Support” cognitive map. This cognitive map indicates the extent to which senior managers provide support to the project [76]. Fig. 3 shows how this cognitive map is related to its six factors: (1) quality of organization project planning (the average of

the QIPan index of projects already undertaken by the organization); (2) appropriate project manager assignment [8,18,74]; (3) involvement of the project manager during the initiation phase [18]; (4) confidence of top manager support in the project [8,18]; (5) secured project funding [37,68]; and (6) sufficient resources for the project [18,37].

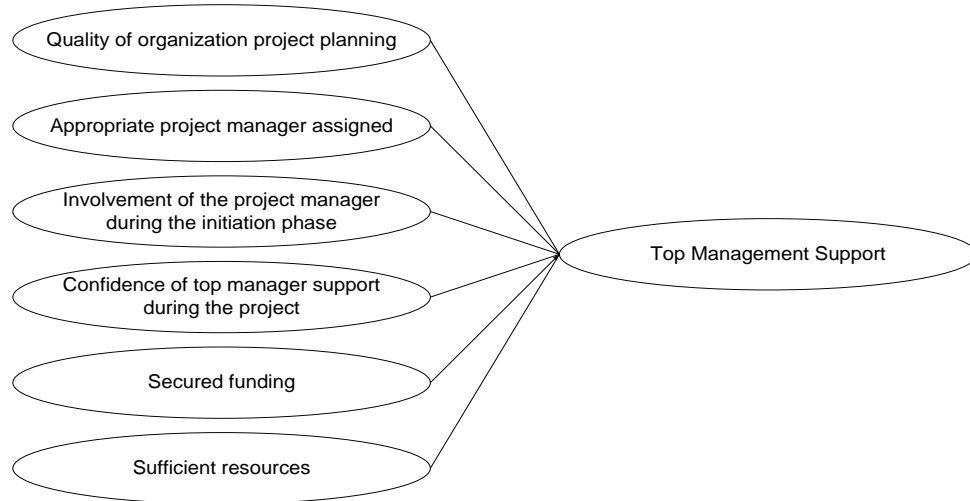


Fig. 3. The structure of the “Top Management Support” cognitive map

To enable one to compare both measures, QCM has 16 cognitive maps to represent the same core planning processes that QPM uses (Table 1), though measured bottom-up rather than top-down. In particular, QCM includes 55 factors and five cognitive maps that enhance the measurement of the 16 core planning processes. For example, the “Top Management Support” cognitive map illustrated in Fig. 3 is one of five maps that, according to the software project success literature, are linked to the core planning process of “develop project management plan”. The other maps (represented in Fig. 4) are: “Project Manager Characteristics”, “Technological Expertise”, “Enterprise Environmental Factors”, and “Quality of Methods and Tools” [66]. This means that the development of the project management plan is affected by success factors that refer to project manager characteristics [51], support from top management to the project (that can lead to its success or failure [76]), environmental factors that affect the quality of planning, and the infrastructure that surrounds or influences a project’s performance [71,74]. In addition, the “Technological Expertise” cognitive map impacts the “Define Scope”, “Create Work Breakdown

Structure”, “Estimate Costs” and “Determine Budget” cognitive maps. This means that the planning process related to scope (“Define Scope” and “Create Work Breakdown Structure”) and cost (“Estimate Costs” and “Determine Budget”) are affected by factors that refer to the knowledge and experience available in the project team (“Technological Expertise”) [21,27,28,36]. All cognitive maps are measured on a five-point Likert scale and normalized to a value that ranges from 0.0 to 1.0.

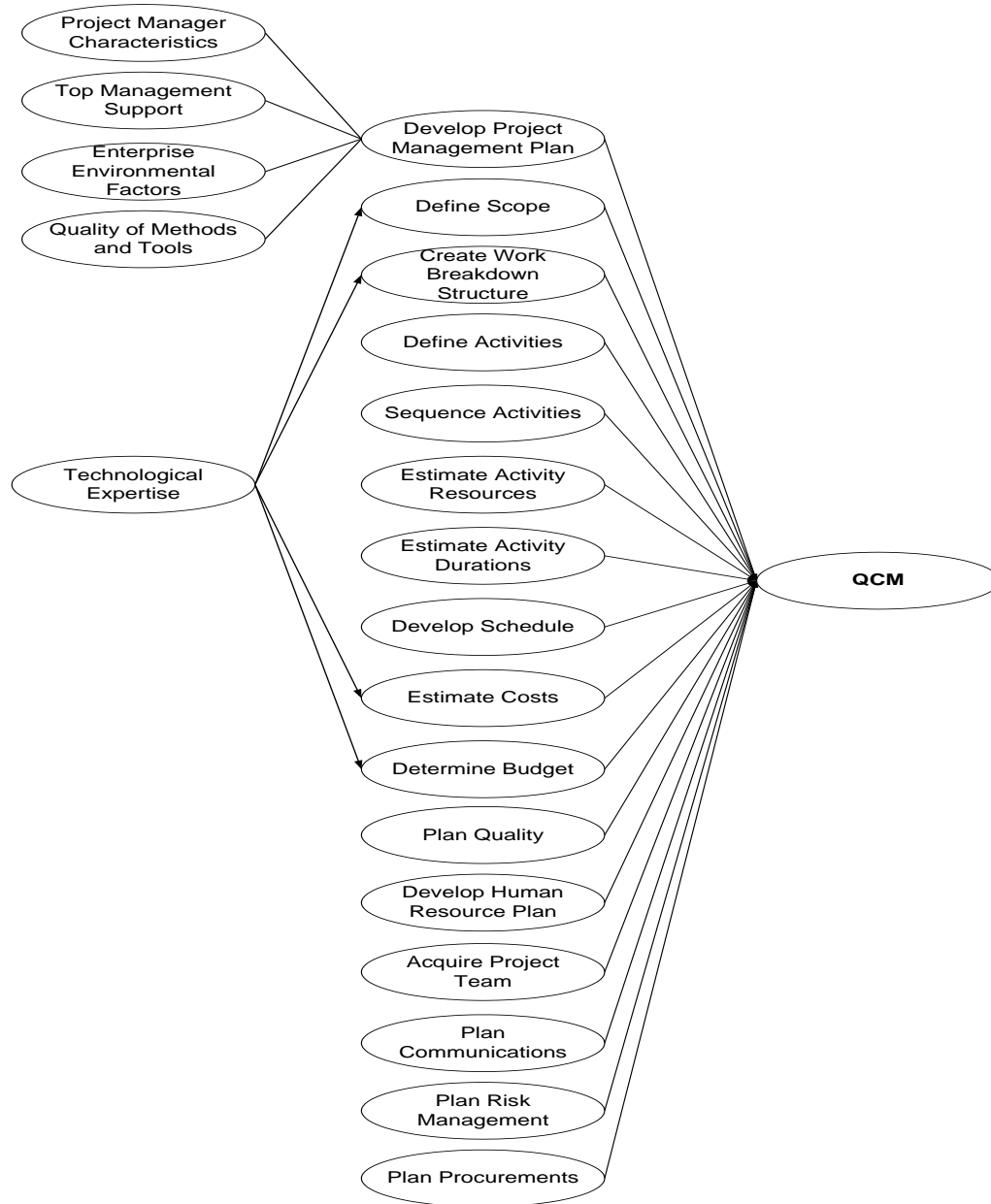


Fig. 4. The 21 cognitive maps of QCM

4.1.2. *Expanded Karnaugh map*

The Karnaugh map is a method from the electronics literature that was developed by Karnaugh in 1953 to simplify real-world logic requirements [31]. Karnaugh maps use the human brain's pattern-matching capability to attain simple logic expressions. This method is mostly used in the electronics industry; however, there are creative exceptions. For example, Sedoglavich expanded the original Karnaugh map into three discrete zones (low, medium and high) for identifying strengths and weaknesses of New Zealand high-tech small and medium enterprises (SMEs) in the agro-technology sector [55]. In QPLAN, the expanded Karnaugh map is an output from QPLAN used for contrasting results from the two measures (QPM and QCM).

4.1.3. *NTCP diamond model*

The NTCP is a model based on contingency theory developed by Shenhar and Dvir [58] for project classification. This is a free-of-context model that helps the project manager to plan the project according to its characteristics. The NTCP diamond model has four dimensions: novelty, technology, complexity and pace.

- *Novelty*: the uncertainty of requirements. The scale is composed of derivative (improvements in current products), platform (new generation of current product) and breakthrough (new product).
- *Technology*: the uncertainty of know-how. The scale is composed of low-tech, medium-tech, high-tech and super high-tech, which are technologies that did not previously exist.
- *Complexity*: the number and diversity of elements in the system. The scale is composed of assembly (performs a single function), system (set of subsystems in a product) and array (dispersed set of systems interconnected).
- *Pace*: the urgency and available timeframe in time management activities and team autonomy. The scale is composed of regular (delays not critical), fast-competitive (time is important), time-critical (crucial) and blitz (needs immediate solution).

4.1.4. Knowledge base

The knowledge base of QPLAN is a database comprising qualitative and quantitative data formed from data from past projects developed by the participating organization. It is available during the entire project lifecycle and serves as a reference for the project manager to check whether the evaluation of a factor that affects the quality of planning, a planning process or even the final quality of planning, is being overestimated or underestimated.

4.2. QPLAN user interface

QPLAN is a desktop application for Microsoft Windows developed in C#, an object-oriented programming language from Microsoft [38], in the integrated development environment (IDE) Microsoft Visual Studio [52]. Fig. 5 shows QPLAN's primary interface with data from a fictitious project as an example.

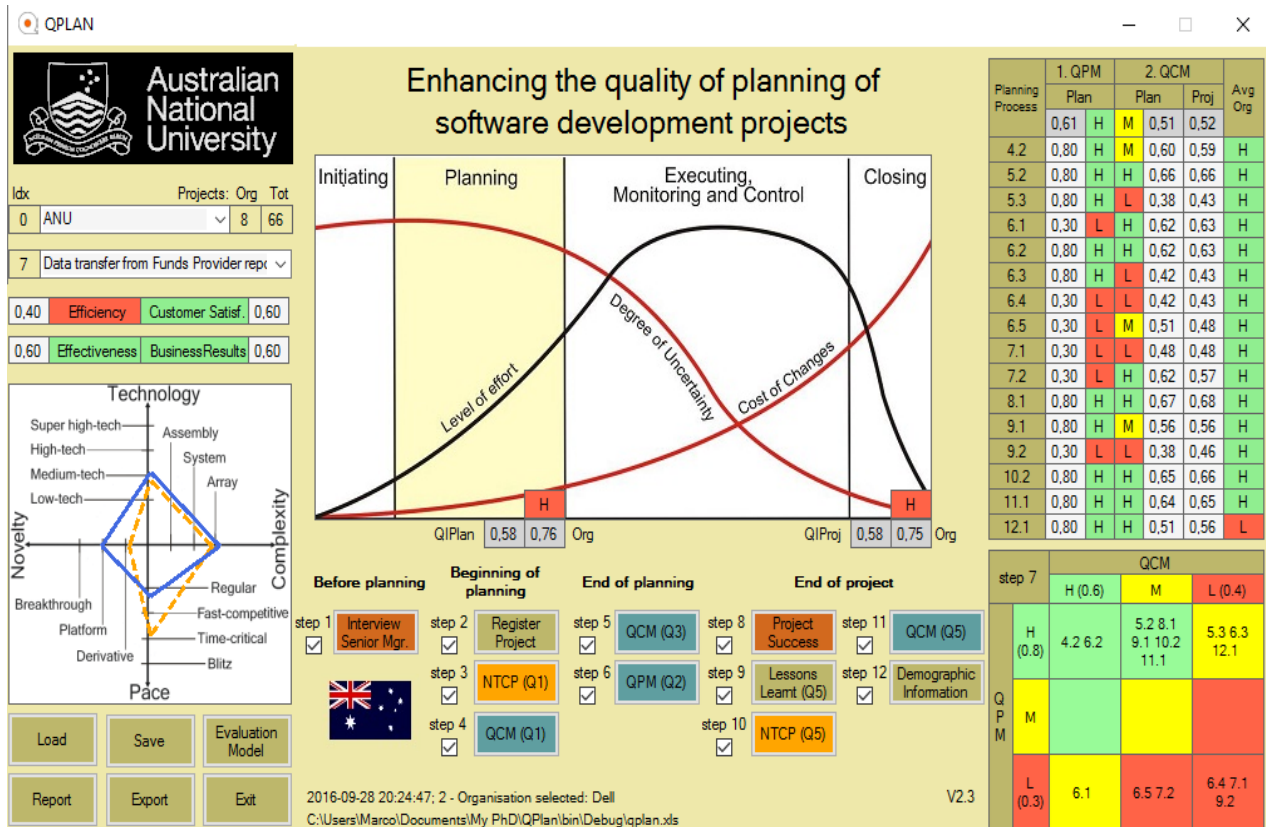


Fig. 5. QPLAN main screen.

The main screen of QPLAN is divided into three main areas. On the left-hand side, from top to bottom, there is the logo of the organization that participated in this research (in this example, the Australian National University), index of the organization number in the QPLAN knowledge base (e.g. zero), organization name (e.g. ANU), number of projects provided by the organization (e.g. 8) and total number of projects in the QPLAN knowledge base (e.g. 66). Below that, there is an indication of the project's success according to the Lechler and Dvir [34] definition (e.g. efficiency of 0.4 of 1) and the graphic representation of the NTCP diamond model in the planning phase (the orange dotted-line) and as perceived at the end of the project (the solid line). In addition, there are six management buttons, including load (to load data from the knowledge base), save (to save data to the knowledge base), Evaluation Model (to access the model that evaluates planning quality), report (that generates project and organization reports), export (to export raw data from the knowledge base) and exit (to exit QPLAN).

The center portion of the screen shows the typical project life cycle adapted from PMBOK [49], with the level of effort, degree of uncertainty and cost of changes across the four project phases. At the end of the planning, and at the end of the project again, the level of risk is represented as high ('H'), medium ('M') or low ('L'). In addition, the QIplan planning quality score is presented with values ranging between 0 and 1, e.g. 0.56. Below that, there are 12 buttons associated with the step number that correspond to the QPLAN data collection approach.

The right-hand side contains a table with the evaluations made by QPM and QCM measures (0.61 and 0.51 in Fig 5, respectively), the average planning quality scores of past projects developed by the same organization (in this example, planning processes are 'L' and 'M'), and the quality indices calculated for each of the 16 core planning processes, coded according to Table 1 [74]. Below this table, an expanded Karnaugh map [55] contrasts the scores from QPM and QCM, which is graphically represented by a 3x3 matrix, i.e., it ranks each of the 16 core planning processes (coded according to Table 1) in nine color-coded quality zones.

4.3. Using QPLAN

The implementation of QPLAN in organizations is based on continuing evaluations of planning quality and introduction of best planning practices from within and outside the organization into the process. The four main phases and 12 main steps in using QPLAN are presented in Fig. 6 and discussed next. Each step also introduces the relevant questionnaire used for data collection. The questionnaires used in steps 4, 5, 6, 8 and 11 measured their items using a five-point Likert Scale (5=“Strongly agree” and 1=“Strongly disagree”).

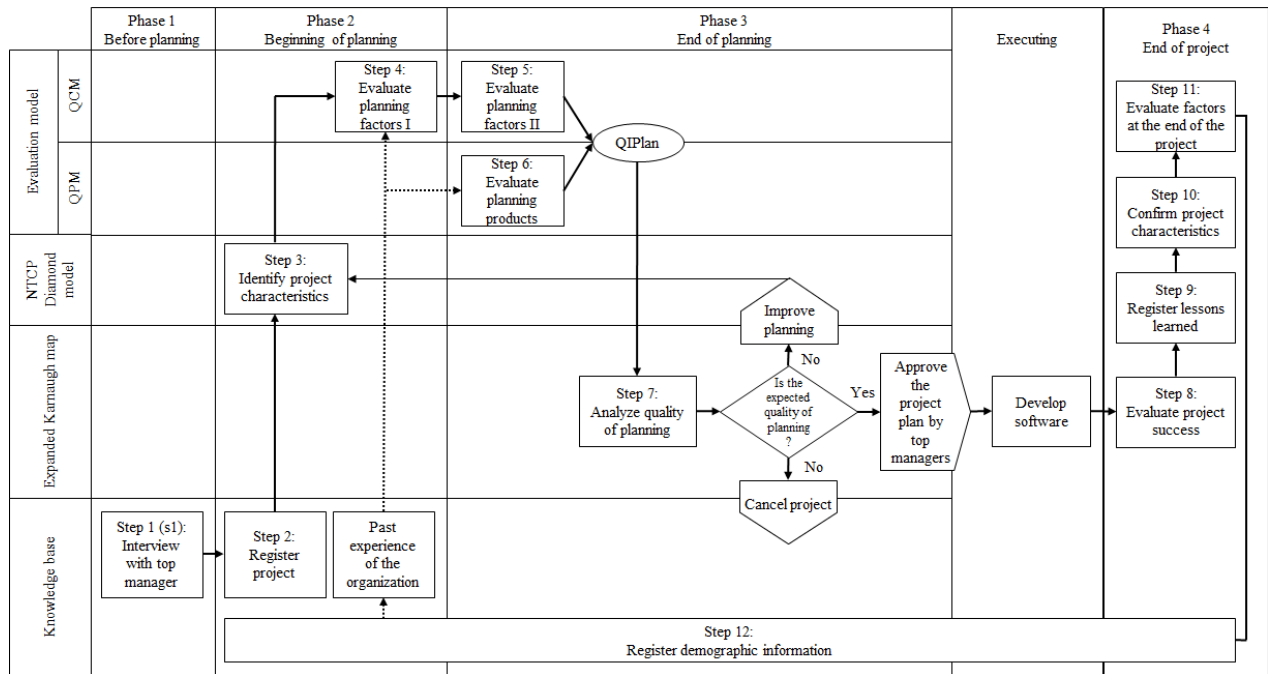


Fig. 6. QPLAN methodology implementation

4.3.1. Phase 1—Before planning

- *Step 1—Interview with top manager:* This step involves a) definition of the success factors adopted in each organization for validating whether the success factors adopted by QPLAN are suitable for software development projects (as the Lechler and Dvir [34] definition is not specific for software development projects), and b) identification of the barriers that have the most significant effect on

the project's success for verifying whether there are factors other than those considered by the evaluation model that affect the quality of planning.

4.3.2. Phase 2—Beginning of planning

- *Step 2—Register project:* This step involves registering the project in the QPLAN knowledge base (Section 4.1.4). This step involves a questionnaire that has eight items. An example item is “Project name”.
- *Step 3—Identify project characteristics:* This step involves classifying the project according to the NTCP diamond model characteristics for helping the project manager select an appropriate managerial approach for planning (Section 4.1.3). The relevant questionnaire has four items that were measured using the scale defined by Shenhar and Dvir [58]. An example item is “How new is the product to customers and users?”
- *Step 4—Evaluate planning factors I:* This step involves evaluating 23 planning factors at the beginning of planning through QCM (Section 4.1.1). It is the first set of factors for the bottom-up planning quality evaluation. The relevant questionnaire has 23 items. An example item is “This project has clear and realistic objectives”.

4.3.3. Phase 3—End of planning

- *Step 5—Evaluate planning factors II:* This step involves evaluating 32 planning factors at the end of planning through QCM (Section 4.1.1). It is the second and last set of factors for the bottom-up planning quality evaluation. The relevant questionnaire has 32 items. An example item is “The project plan has small software releases planned”.
- *Step 6—Evaluate planning products:* This step involves evaluating core planning products at the end of planning through QPM (Sections 2.1 and 4.1.1). It is the set of 16 factors for the top-down planning quality evaluation. The relevant questionnaire has 16 questions. An example item is “The project plan is able to deliver the scope required on-time”.

- *Step 7—Analyze quality of planning:* This step involves analyzing planning quality by the project manager through data collected in previous steps, including: (1) the expanded Karnaugh map; (2) a project report, which has all project data, suggestions for enhancing planning quality, and comparisons with past projects; (3) an organization report, which has a roadmap of projects, their performance, and a list of common planning issues that project managers have reported; and (4) raw data existing in the knowledge base, which one can export to other tools. The analysis available at the end of this step serves to determine whether the project should progress to the execution phase, planning should be improved until better results are achieved (by focusing on the most important planning issues identified in this step), or the project should be terminated before investing more resources.

4.3.4. Phase 4—End of project

- *Step 8—Evaluate project success:* This step involves evaluating the project's success at the end of the project based on the measures defined by Lechler and Dvir [34]: (1) efficiency: the extent to which the project has met time and cost plans; (2) effectiveness: the extent of the benefits that the project has brought to its client; (3) customer satisfaction: the extent of satisfaction with the benefits that the project has provided; and (4) business results: the perceived value of the project to the client organization. The relevant questionnaire has 12 items. An example item is "The project was completed on schedule".
- *Step 9—Register lessons learned:* This step involves registering two open-ended questions for capturing what went well and what should be done differently in the future through a qualitative approach [27].
- *Step 10—Confirm project characteristics:* This step involves confirming project characteristics entered in Step 3 through the NTCP diamond model (Section 4.1.3) to support the lessons learned process. The relevant questionnaire has four items that were measured using the scale defined by Shenhar and Dvir [58]. An example item is "How complex is the system and its subsystems?"

- *Step 11—Evaluate project performance factors:* This step involves evaluating 12 project performance factors at the end of the project (i.e., actual versus planned factors) to support the lessons learned process. The relevant questionnaire has ten items. An example item is “It was a high-risk project”.
- *Step 12— Register demographic information:* This step involves finalizing all project information registered in the QPLAN knowledge base (Section 4.1.4). Data from this project will serve as a reference for the next project to be undertaken by the organization. The relevant questionnaire has eight items. An example item is “Number of employees in your organization”.

5. Evaluation case of QPLAN in the defense industry

5.1. Overview

In this section, we describe how one organization used QPLAN as a summative evaluation case. We performed the case study in a multinational organization focused on defense, homeland security and commercial products. The organization had approximately 12,000 employees. The case study was longitudinal and included qualitative feedback from a top manager and two project managers in Brazil and Israel.

Note that, because QPLAN represents intellectual property and may be marketed, the case study organization did not use the final tool. Instead, it used a surrogate method with a “QPLAN” questionnaire that included all the factors assessed in QPLAN and was emailed to managers. We then used QPLAN to create project reports based on their questionnaire responses, and we discussed and analyzed these reports with the project managers.

5.2 Longitudinal study of QPLAN use

The case study involved analyzing 20 projects that ranged in duration from three to 72 months (mean = 2.6 years). We divided the projects into two groups: one group included ten projects that had

occurred in the past (group 1) and the second group an additional ten current projects (group 2). The data provided by past projects (group 1) served to build the knowledge base of the organization to allow project managers to use them as reference for the planning of current projects, i.e., projects from group 1 did not receive the QPLAN intervention, whereas projects from group 2 did. However, we calculated the QIPlan index for all projects to analyze the impact of the intervention on the organization.

We collected data from 18 project managers (some managers had more than one project) between January 2011 and October 2012 using questionnaires. These questionnaires are paper-based surrogates for the questionnaires used, as described in Steps 1 to 12 in section 4.2. Project managers were asked to identify the project, classify it, and evaluate the initial conditions at the beginning of the planning. At the end of the planning, project managers were asked to evaluate its quality. At the end of the project, supervisors were asked to evaluate its success, and project managers were asked to identify enhancement opportunities and compare actual data against planned data, as well as fill out the demographic information sheet. Questionnaires were completed with an average duration of 20 minutes (each). The QIPlan index for all projects had an alpha coefficient of .828, which suggests acceptable internal reliability. We then conducted a comparison analysis by assessing the efficiency of the observed QPLAN process in terms of differences in the QIPlan index [65] between groups 1 and 2. Results show that the organizational planning quality index had a positive and significant slope over time ($\beta = .896$, $p < .001$, $t = 8.337$). Fig. 7 shows that the average planning quality (QIPlan, ranging from “0” to “1”) for the ten projects conducted without the QPLAN intervention and those ten conducted with the support of the tool were .52 and .61, respectively. In addition, the QIPlan range (difference between maximum and minimum scores) reduced from .32 (group 1) to .23 (group 2), suggesting the organization had a more controlled planning process after using QPLAN, i.e., QPLAN intervention reduced process variation [12].

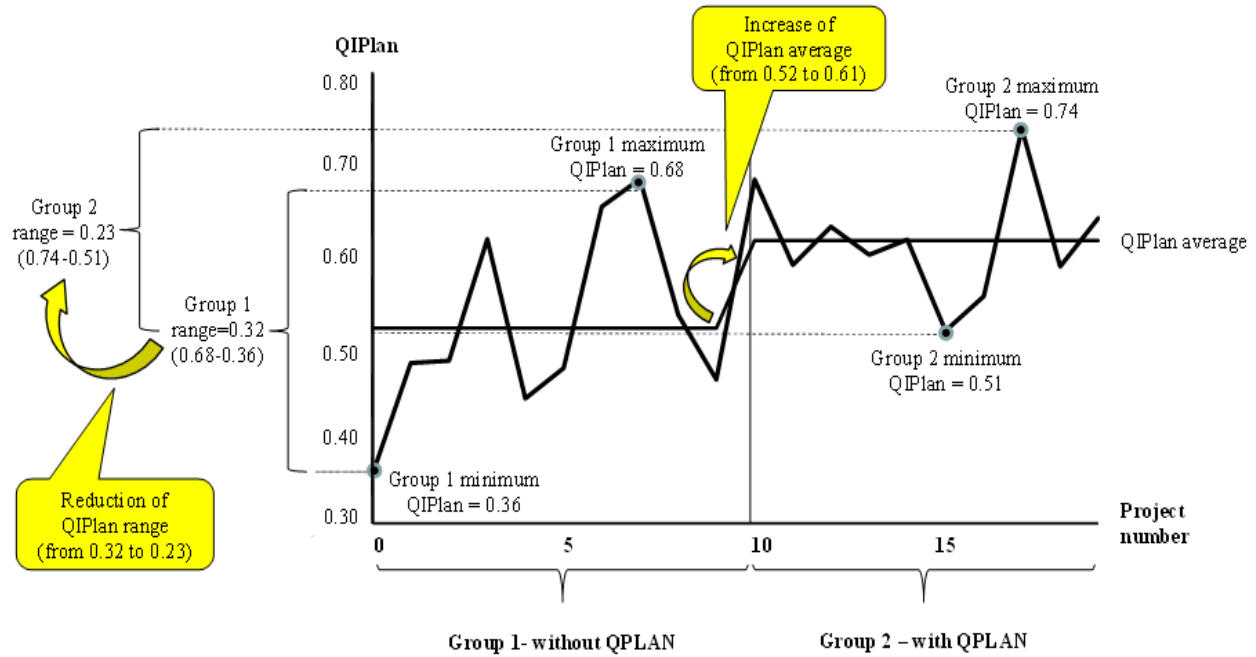


Fig. 7 The effect of QPLAN on planning quality enhancement.

Results of the longitudinal study showed that the QPLAN approach was adequate for enhancing the planning quality of software development projects over time in the company. This finding served to demonstrate the utility of the artifact, which is the essence of DSR [23].

5.3 Qualitative feedback

We obtained qualitative data from a top manager in an interview and from two project managers using questionnaires. We conducted a 20-minute interview with the vice-president of engineering (VPE) in the Brazil branch, which had approximately 90 engineers who developed software, hardware, and mechanics projects. We conducted the interview face-to-face in April 2012. We asked the VPE how the company measured project success and to identify the common factors that led to project failure in the organization. In response to the first question, the VPE answered that delivering on time and on cost and to customer satisfaction were the success measures that the organization adopted. QPLAN covers these success measures as efficiency and customer satisfaction. For the second question, he identified two main issues that led to project failure: (1) requirements instability, which typically led to software redesign; and

(2) the software-certification process, which, due to its complexity, led to budget and schedule overruns. Two QCM factors cover these issues: ineffective change management and unrealistic effort estimates.

Over the entire study, 20 project managers (out of 48) provided feedback about QPLAN. During the qualitative analysis of the transcriptions, three major feedback insights emerged relating to questionnaires, reports and QPLAN applicability to agile projects. This feedback has contributed to the improvement of QPLAN throughout the research.

In the summative case study, two project managers provided feedback about their experience of the use of QPLAN in their projects. Some insights from this feedback are as follows. One project manager commented that the QPLAN-based questionnaire covered planning aspects that software development teams usually neglect: “The questionnaires contain sets of very interesting questions that cover many program management aspects that sometimes are not completely understood or considered by the development teams.” In addition, he commented that the QPLAN surrogate helped the development of better planning: “It leads to very interesting thoughts that sometimes change the perception of how we used to see the program by understanding the different points of view”. Another project manager said that the information that QPLAN provided made sense and portrayed what actually happened in the project: “The planning that seemed to be chaotic by someone who experienced the whole process, started to look not so bad after putting things together”.

5.4 Discussion

The QPLAN approach implemented in this case study integrated existing and new methods for evaluating planning quality, including the PMPQ generic planning evaluation model (which has been validated and used extensively by the literature), QIPlan index (a software development evaluation metric), and QPLAN questionnaires (which serve as checklists). This case study showed that with a QPLAN intervention the average planning quality increased and the variation of planning process reduced. In addition, project managers recognized that QPLAN covers planning aspects that are sometime neglected, increases the understanding of what happens during this project phase, and organizes planning information.

QPLAN reduced planning bias and enhanced project performance in this organization by introducing a consultative and structured planning process, evidence-based evaluation process, and high quality information presented to decision makers. However, one should note that even a comprehensive and quality plan cannot guarantee success [53,58,60]. For example, volatility of requirements, intangibility of software products, and high level of system complexity remain challenges to software projects during their execution, hence require continuous changes to the original plan [45].

6. Conclusion

Motivated by the significance of the contemporary software industry in the global economy and the poor performance of software-development projects, the literature has highlighted planning quality as a major factor that has the potential to enhance project and organizational performance. Better support is needed for managers in evaluating planning quality correctly and hence making better decisions on whether to approve a project plan and develop the software, improve the plan, or terminate the project. In this study, we identified limitations of existing methods for evaluating planning quality and the psychological biases that prevent managers from making correct decisions about projects.

As discussed previously, dual process theory argues that the frequency of poor decisions increases due to psychological biases, such as the endowment effect, optimism bias and ambiguity effect. Because QPLAN is an empirical and evidence-based tool, it can resolve this problem by providing high quality planning information when the level of uncertainty is at its peak and thus improve the quality of decisions, i.e., QPLAN incorporates System 2 analysis into System 1 intuitions. In addition, as QPLAN is designed to be used by practitioners, it overcomes an important issue in decision support system research: the lack of relevance for IT professionals [6].

To develop QPLAN, we first examined the decision science, project management, information systems and psychology literatures. We selected DSR [23] as a research approach and used DSRP [48] as a research process because it provides a process for conducting DSR and a mental model for the research

output. We validated QPLAN in the business environment, concurrently with the design and development, through multiple case studies and a variety of quantitative and qualitative methods. The data collected represented a significant and rich sample of software development projects that were analyzed through quantitative and qualitative methods.

As a result, the research question could be answered. The quality of planning of software development projects can be better evaluated through a model that combines two distinct and complementary analyses (top-down and bottom-up), and the quality of planning can be improved through the identification of the strengths and weaknesses of planning, the identification of project characteristics and a knowledge base for organization learning. This claim is supported by the results provided from the test of the long-term effect of QPLAN in enhancing the quality of planning over time and the feedback provided by practitioners in a summative case study.

Nonetheless, a methodological limitation of this paper is the single case study approach, which does not allow the full validation of QPLAN. One may argue that the approach for enhancing planning quality discussed in this paper may not be as effective under different cultural and organizational circumstances. For example, QPLAN may be more effective where the levels of project complexity and uncertainty are higher, and information is incomplete or ambiguous. Differences may also be noticed when comparing different types of organizational approaches to project management. For example, software development organizations that use waterfall project management methodologies (where planning needs to be more comprehensive) may find QPLAN more effective than those using agile methodologies (where deliverables are set as late as possible [57] and planning is hence becoming an evolving process during project execution). Accordingly, future research could compare QPLAN effectiveness and bias reduction in different countries, industries, and project management approaches.

Given the massive costs that can result from suboptimal decisions [43] caused by psychological biases, this work relies on replacing the instinctive and emotional decision-making (system 1) with a

formal analytic process (system 2), by introducing a decision support tool (QPLAN) that requires one to analyze data and evaluate planning quality rationally before making a decision.

7. References

- [1] A. Abran, P. Bourque, R. Dupuis, J. Moore, Guide to the software engineering body of knowledge - SWEBOK, (2004) 1–202. doi:10.1109/SESS.1999.767664.
- [2] J.E. van Aken, Design science and organization development interventions, *The Journal of Applied Behavioral Science*. Sci. 43 (1) (2007) 67–88. doi:10.1177/0021886306297761.
- [3] J.E. van Aken, Management research on the basis of the design paradigm: The quest for field-tested and grounded technological rules, *Journal of Management Studies*. 41 (2) (2004) 219–246. doi: 10.1111/j.1467-6486.2004.00430.x
- [4] A.A. Alblas, J.C. Wortmann, Managing large engineering changes, *International Journal of Operations and Production Management*. 32 (11) (2012) 1252–1280. doi: 10.1108/01443571211274549.
- [5] J.S. Armstrong, The value of formal planning for strategic decisions: Review of empirical research, *Strategic Management Journal*. 3 (3) (1982) 197–211. doi: 10.1002/smj.4250030303.
- [6] D. Arnott, G. Pervan, A critical analysis of decision support systems research revisited: The rise of design science, *Journal of Information Technology*. 29 (4) (2014) 269–293. doi:10.1057/jit.2014.16.
- [7] D. Arnott, G. Pervan, Eight key issues for the decision support systems discipline, *Decision Support Systems*. 44 (3) (2008) 657–672. doi:10.1016/j.dss.2007.09.003.
- [8] P.L. Bannerman, Risk and risk management in software projects: A reassessment, *Journal of Systems and Software*. 81 (12) (2008) 2118–2133. doi:10.1016/j.jss.2008.03.059.
- [9] R.S. Barbour, Checklists for improving rigour in qualitative research: A case of the tail wagging the dog? *British Medical Journal*. 322 (2001) 1115–1117. doi:10.1136/bmj.322.7294.1115.

- [10] L. Barry, M.-L. & Uys, An investigation into the status of project management in South Africa, *South African Journal of Industrial Engineering*. 22 (1) (2011) 29–44. doi: <http://dx.doi.org/10.7166/22-1-31>.
- [11] R. Baskerville, What design science is not, *European Journal of Information Systems*. 17 (5) (2008) 441–443. doi:10.1057/ejis.2008.45.
- [12] F.W. Breyfogle, *Implementing six sigma: Smarter solutions using statistical methods*, second edition, John Wiley & Sons, Inc., Hoboken, NJ, 2003. doi:10.2307/1271088.
- [13] D.C. Conforto, E.C. Amaral, Evaluating an agile method for planning and controlling innovative Projects, *Project Management Journal*. 41 (2) (2010) 73–80. doi: 10.1002/pmj.20089.
- [14] M.K. Daskalantonakis, A practical view of software measurement and implementation experiences within Motorola, *IEEE Transactions on Software Engineering*. 18 (11) (1992) 998–1010. doi: 10.1109/32.177369.
- [15] D.J. Flynn, E.A. Arce, A CASE tool to support critical success factors analysis in IT planning and requirements determination, *Information and Software Technology*. 39 (5) (1997) 311–321. doi:10.1016/S0950-5849(96)01150-0.
- [16] B. Flyvbjerg, M.K. Skamris Holm, S.L. Buhl, How (in)accurate are demand forecasts in public works projects? The case of transportation, *Journal of the American Planning Association*. 71 (2) (2005) 131–146. doi:10.1080/01944360508976688.
- [17] B. Flyvbjerg, Quality control and due diligence in project management: Getting decisions right by taking the outside view. *International Journal of Project Management*. 31 (5) (2013) 760–774.
- [18] J. Fortune, D. White, Framing of project critical success factors by a systems model, *International Journal of Project Management*. 24 (1) (2006) 53–65. doi:10.1016/j.ijproman.2005.07.004.
- [19] Gartner says worldwide IT spending forecast to grow 2.7 percent in 2017. <http://www.gartner.com/newsroom/id/3568917>.

- [20] A. Gopal, M.S. Krishnan, T. Mukhopadhyay, D.R. Goldenson, Measurement programs in software development: Determinants of success, *IEEE Transactions on Engineering Management*. 28 (9) (2002) 863–875. doi:10.1109/TSE.2002.1033226.
- [21] I.B.M. Staff, IBM rational unified process: Best practices for software development teams, (2003). <http://www.ibm.com/developerworks/rational/library/253.html>.
- [22] S. Gregor, A.R. Hevner, Positioning and presenting design science research for maximum impact, *MIS Quarterly*. 37 (2) (2013) 337–355. doi:10.2753/MIS0742-1222240302.
- [23] A.R. Hevner, S.T. March, J. Park, S. Ram, Design science in information systems, *MIS Quarterly*. 28 (1) (2004) 75–105.
- [24] D. Houston, The value of a good checklist, *Software Quality Professional*. 6 (2) (2004) 17–26. doi:10.1109/MC.2010.181.
- [25] International Organization for Standardization, ISO/IEC 15939:2007 - Systems and software engineering - Measurement process (2007). http://www.iso.org/iso/catalogue_detail.htm?csnumber=44344.
- [26] International Organization for Standardization, ISO/IEC FCD 25040, Software engineering - software product quality requirements and evaluation (SQuaRE) - data quality model (2009). http://www.iso.org/iso/catalogue_detail.htm?csnumber=35736.
- [27] M. Jørgensen, T.M. Gruschke, The impact of lessons-learned sessions on effort estimation and uncertainty assessments, *IEEE Transactions on Software Engineering*. 35 (3) (2009) 368–383.
- [28] M. Jørgensen, Top-down and bottom-up expert estimation of software development effort, *Information and Software Technology*. 46 (1) (2004) 3–16. doi:10.1016/S0950-5849(03)00093-4.
- [29] D. Kahneman, *Thinking, Fast and Slow*, first edition, Farrar, Straus and Giroux, New York, 2011.
- [30] D. Kahneman, J.L. Knetsch, R.H. Thaler, The endowment effect, loss aversion, and status quo bias: Anomalies. *Journal of Economic Perspectives*, 5 (1) (1991) 193-206. doi: 10.1257/jep.5.1.193.

- [31] M. Karnaugh, The map method for synthesis of combinational logic circuits. Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics, 72 (5) (1953) 593–599. doi: 10.1109/TCE.1953.6371932.
- [32] M. Keil, L. Li, L. Mathiassen, G. Zheng, The influence of checklists and roles on software practitioner risk perception and decision-making. Journal of Systems and Software. 81 (6) (2008), 908–919. doi: 10.1016/j.jss.2007.07.035.
- [33] M. Krishnan, C. Kriebel, An empirical analysis of productivity and quality in software products, Management Science. 46 (6) (2000) 745–759. doi:10.1016/S0267-3649(00)88914-1.
- [34] T. Lechler, D. Dvir, An alternative taxonomy of project management structures: Linking project management structures and project success, IEEE Transactions on Software Engineering. 57 (2) (2010) 198-210. doi: 10.1109/TEM.2010.2044441.
- [35] C. Lin, G. Pervan, The practice of IS/IT benefits management in large Australian organizations. Information and Management. 41 (1) (2003) 13–24. doi:10.1016/S0378-7206(03)00002-8.
- [36] F.Y.Y. Ling, S.P. Low, S.Q. Wang, H.H. Lim, Key project management practices affecting Singaporean firms' project performance in China, International Journal of Project Management. 27 (1) (2009) 59–71. doi:10.1016/j.ijproman.2007.10.004.
- [37] T.C. Loh, S.C.L. Koh, Critical elements for a successful enterprise resource planning implementation in small- and medium-sized enterprises, International Journal of Production Research. 42 (17) (2004) 3433–3455. doi:10.1080/00207540410001671679.
- [38] M.H. Lutz, P.A. Laplante, C# and the .NET framework: Ready for real time? IEEE Software. 20 (1) (2003) 74–80. doi:10.1109/MS.2003.1159034.
- [39] R. Mahanti, A. Jiju, Six Sigma in the Indian software industry: Some observations and results from a pilot survey, The TQM Journal. 21 (6) (2009) 549–564. doi: 10.1108/17542730910995837

- [40] S. March, G. Smith, Design and natural science research on information technology, *Decision Support Systems*. 15 (4) (1995) 251–266. doi:10.1016/0167-9236(94)00041-2. doi:10.1016/0167-9236(94)00041-2.
- [41] B. Masters, G.V. Frazier, Project quality activities and goal setting in project performance Assessment, *Quality Management Journal*. 14 (3) (2007) 25–35.
- [42] J. Mihm, Incentives in New product development projects and the role of target costing, *Management Science*. 56 (8) (2010) 1324–1344. doi:10.1287/mnsc.1100.1175.
- [43] K.L. Milkman, D. Chugh, M.H. Bazerman. How can decision making be improved? Perspectives on *Psychological Science*. 4 (4) (2009) 379-383. doi: 10.1111/j.1745-6924.2009.01142.x.
- [44] H. Mintzberg, D. Raisinghani, A. Théorêt. The structure of "unstructured" Decision Processes. *Administrative Science Quarterly*. 21 (2) (1976) 246-275. doi: 10.2307/2392045.
- [45] N.P. Napier, M. Keil, F.B. Tan, IT project managers' construction of successful project management practice: A repertory grid investigation, *Information Systems Journal*. 19 (3) (2009) 255–282. doi:10.1111/j.1365-2575.2007.00264.x.
- [46] G.D. Norris, Post-investment appraisal, in: L. Willcocks, *Investing in information systems: Evaluation and Management*, London, Chapman & Hall, 1996, pp. 193–223.
- [47] K.E. Papke-Shields, C. Beise, J. Quan, Do project managers practice what they preach, and does it matter to project success? *International Journal of Project Management*. 28 (7) (2010) 650–662. doi:10.1016/j.ijproman.2009.11.002.
- [48] Peffers, K., Tuunamen, T., Rotenberger, M.A., Chatterjee, S., A Design science research methodology for information systems research. *Journal of Management Information Systems*. 24 (3) (2007) 45-77. doi:10.2753/MIS0742-1222240302.
- [49] Project Management Body of Knowledge (PMBOK ®) Guide – fifth edition. Project Management Institute, Newtown Square, PA, 2013.

- [50] S. Raghunathan, Impact of information quality and decision-maker quality on decision quality: A theoretical model and simulation analysis, *Decision Support Systems*. 26 (4) (1999) 275–286. doi:10.1016/S0167-9236(99)00060-3.
- [51] K. Rees-Caldwell, A.H. Pinnington, National culture differences in project management: Comparing British and Arab project managers' perceptions of different planning areas, *International Journal of Project Management*. 31 (2) (2013) 212–227. doi:10.1016/j.ijproman.2012.04.003.
- [52] A.R. Rezaei, T. Çelik, Y. Baalousha, Performance measurement in a quality management system, *Transaction E, Industrial Engineering*. 18 (3) (2011) 742–752. doi:10.1016/j.scient.2011.05.021.
- [53] L. Rodriguez-Repiso, R. Setchi, J.L. Salmeron, Modelling IT projects success with fuzzy cognitive maps, *Expert Systems with Applications*. 32 (2) (2007) 543–559. doi:10.1016/j.eswa.2006.01.032.
- [54] S.F. Rossman, G.B., Rallis, *Learning in the field: An introduction to qualitative research*, second edition, Sage Publications, Thousand Oaks, CA, 2003.
- [55] V. Sedoglavich, Absorptive capacity and internalization of New Zealand high-tech SMEs in the agro-technology sector (Thesis, Doctor of Philosophy (PhD)). The University of Waikato (2008). <http://researchcommons.waikato.ac.nz/handle/10289/2606>.
- [56] E. Salas, M.A. Rosen, Expertise-based intuition and decision making in organizations. *Journal of Management*. 36 (4) (2010) 941-973. doi:10.1177/0149206309350084.
- [57] P. Serrador, J.K. Pinto, Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management*. 33 (5) (2015) 1040-1051. doi: 10.1016/j.ijproman.2015.01.006.
- [58] A. Shenhar, D. Dvir, *Reinventing project management: The diamond approach to successful growth and innovation*. Harvard Business School Press. Boston, MA, 2007
- [59] M. Shepperd, M. Cartwright, Predicting with sparse data, *IEEE Transactions on Software Engineering*. 27 (11) (2001) 987–998. doi:10.1109/32.965339.

- [60] O. Shmueli, N. Pliskin, L. Fink, Can the outside-view approach improve planning decisions in software development projects? *Information Systems Journal*. 26 (4) (2016) 395-418. doi: 10.1111/isj.12091.
- [61] O. Shmueli, N. Pliskin, L. Fink, Explaining over-requirement in software development projects: An experimental investigation of behavioral effects. *International Journal of Project Management*. 33 (2) (2015) 380–394. doi: 10.1016/j.ijproman.2014.07.003
- [62] Software Engineering Institute, CMMI® for Development, Version 1.3, (2010).
<http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9661>.
- [63] W. Stach, L. Kurgan, W. Pedrycz, M. Reformat, Genetic learning of fuzzy cognitive maps, *Fuzzy Sets and Systems*. 153 (3) (2005) 371–401. doi:10.1016/j.fss.2005.01.009.
- [64] Standish Group, Chaos manifesto 2013: Think big, act small, (2013) 1–52.
<https://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf>.
- [65] Z. Stojanov, D. Dobrilovic, J. Stojanov, Analyzing trends for maintenance request process assessment: Empirical investigation in a very small software company, *Theory and Applications of Mathematics & Computer Science*. 3 (2) (2013) 59–74.
- [66] G.P. Sudhakar, A model of critical success factors for software projects, *Journal of Enterprise Information Management*. 25 (6) (2012) 537–558. doi:10.1108/17410391211272829.
- [67] C. Symons, Metrics Software industry performance, *IEEE Software*. 27 (6) (2010) 66–72. doi:10.1109/MS.2009.162.
- [68] D. Tesch, T.J. Kloppenborg, M.N. Frolick, IT project risk factors: The project management professional's perspective, *Journal of Computer Information Systems*. 47 (4) (2007) 61–69.
- [69] M. Weick, A. Guinote, How long will it take? Power biases time predictions, *Journal of Experimental Social Psychology*. 46 (4) (2010) 595–604. doi:10.1016/j.jesp.2010.03.005.
- [70] D. White, J. Fortune, Current practice in project management — an empirical study, *International Journal of Project Management*. 20 (1) (2002) 1–11. doi:10.1016/S0263-7863(00)00029-6.

- [71] O. Zwikael, A. Sadeh, Planning effort as an effective risk management tool, *Journal of Operations Management*. 25 (4) (2007) 755–767. doi:10.1016/j.jom.2006.12.001.
- [72] O. Zwikael, Critical planning processes in construction projects, *Construction Innovation*. 9 (4) (2009) 372–387. doi:10.1108/14714170910995921.
- [73] O. Zwikael, M. Ahn, The effectiveness of risk management: An analysis of project risk planning across industries and countries, *Risk Analysis: An International Journal*. 31 (1) (2011) 25–37.
- [74] O. Zwikael, S. Globerson, Evaluating the quality of project planning: A model and field results, *International Journal of Production Research*. 42 (8) (2004) 1545–1556.
doi:10.1080/00207540310001639955.
- [75] O. Zwikael, S. Globerson, From critical success factors to critical success processes, *International Journal of Production Research*. 44 (17) (2006) 3433–3449. doi:10.1080/00207540500536921.
- [76] O. Zwikael, Top management involvement in project management: Exclusive support practices for different project scenarios. *International Journal of Managing Projects in Business*. 1 (3) (2008) 387–403. doi: 10.1108/17538370810883837.



Dr. Marco Féris is a lecturer in programme and project management at Cranfield School of Management. Dr. Féris has PhD in project management, master and bachelor degrees in computer science, graduate certificate in business management and six sigma black belt and green belt certifications. Dr. Féris' research interests are focused on planning quality, process improvement and software development projects. As a program/project manager, Dr. Féris worked in significant organizations, including AEL/Elbit, Australian National University (ANU), Hewlett-Packard (HP), Dell Computers and Altus Automation Systems, in four countries (Brazil, Australia, Israel and US) and in different types of industries (IT, R&D, automation, education and defense).



Dr. Ofer Zwikael is an Associate Dean in the College of Business and Economics at the Australian National University. His research focuses on project selection, management and evaluation. Dr. Zwikael is the author of three books and more than 100 scholarly peer-reviewed papers published in leading outlets, such as the Journal of Operations Management, Journal of Management in Engineering and the British Journal of Management. He currently serves on the Editorial Board of the International Journal of Project Management. Dr. Zwikael is the recipient of the International Project Management Association's Outstanding Research Contributions Award. In addition, he has been awarded multiple best paper awards by the Academy of Management and the British Academy of Management. For the impact of his research, Ofer Zwikael received the Emerald 2015 Citations of Excellence award.



Dr. Shirley Gregor is Professor of Information Systems at the Australian National University, Canberra, where she is a Director of the National Centre for Information Systems Research. Professor Gregor's research interests include the adoption and strategic use of information and communications technologies, intelligent systems, human-computer interaction and the philosophy of technology. She has published in journals such as *MIS Quarterly*, *Journal of the Association of Information Systems*, *International Journal of Electronic Commerce*, *International Journal of Human Computer Studies*, *European Journal of Information Systems*, and *Information Technology & People*. Professor Gregor was made an Officer of the Order of Australia in June 2005 for services as an educator and researcher in the field of information systems and for work in ecommerce in the agribusiness sector. She is a Fellow of the Australian Computer Society and a Fellow of the Association for Information Systems.

QPLAN: Decision support for evaluating planning quality in software development projects

Amaral Féris, Marco Antônio

2017-02-20

Attribution-NonCommercial-NoDerivatives 3.0

Marco Antônio Amaral Féris, Ofer Zwikael, Shirley Gregor, QPLAN: Decision support for evaluating planning quality in software development projects, *Decision Support Systems*, Vol. 96, April 2017, pp. 92-102

<http://dx.doi.org/10.1016/j.dss.2017.02.008>

Downloaded from CERES Research Repository, Cranfield University