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

Marco Antônio Amaral Féris a,⁎, Ofer Zwikaël 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

A R T I C L E   I N F O
Article history:
Received 4 May 2016
Received in revised form 7 February 2017
Accepted 15 February 2017
Available online 20 February 2017

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

A B S T R A C T
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.

© 2017 Elsevier B.V. All rights reserved.

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 [20]. Nonetheless, poor performance of software development projects has plagued the IT industry for years [34]. In 2013, for instance, organizations successfully completed only 35% of software projects [65], which led to estimated annual losses for the United States (US) and European Union (EU) markets of around US$100 billion each [68].

Research has identified planning quality1 as a key area for improvement in overcoming software projects’ poor performance [14,22,67,75]. 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 [22] and business contexts [18], reduces projects’ inherent uncertainty, and provides a reliable basis for monitoring and controlling projects [73]. Planning involves comparing alternative courses of action [4,9], and establishing a pathway to accomplish the project’s goals [60]. 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-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 [74], 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 [31], or defining requirements beyond the actual needs of the customer or the market [62]. Moreover, managers can have a tendency to overestimate the quality of project plans or be too optimistic about future projects’ performance (“optimism bias”) [30, 70], hence approving project plans that are of questionable quality. Finally, an “ambiguity effect” supports a tendency to avoid decision making due to missing information [30], and managers can avoid making a decision on project termination when they should do so.

1“Planning quality” is equivalent to “quality of planning” and refers to a project’s planning stage, in 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.
These psychological biases have a negative impact on decision making [45], especially under high levels of uncertainty when information is incomplete or ambiguous [30,36,47]. These biases are intensified when decision makers rely on their personal knowledge, known as an "inside view" [30,70]. In an "inside view", decision makers build overly optimistic scenarios of events that may affect the project [18]. 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 [18,30].

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 [57]. 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 [30,44]. Systems 1 and 2 operate in parallel and interactively [57]. 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 [30], 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 [44]. 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 [17,43,51], and 2) evaluation requires measurement and learning [21]. Thus, we address the following research question: “How can the planning quality of software development projects be better evaluated and improved?”

2.2. Checklist approaches

Checklists are a common approach to assessing planning quality and risks [33]. Based on expert knowledge [25], software-development projects often use checklists to determine whether the planning phase has finished and the project can proceed with the next phase [33], 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) [63]; (2) Six Sigma, a set of techniques and tools for improving manufacturing [13] and software-development processes [40]; (3) ISO/IEC 15939, a measurement process applicable to system and software engineering and management fields [26]; (4) process improvement training and appraisal program for software development [63]; and (5) the SquaRE model, a general data-quality model for data retained in a structured format in a computer system [27]. Checklists can incorporate expert knowledge of a process, including lessons learnt from past projects [25].

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 [10,25].

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 of evaluatin...
دریافت فوری
متن کامل مقاله
امکان دانلود نسخه تمام متن مقالات انگلیسی
امکان دانلود نسخه ترجمه شده مقالات
پذیرش سفارش ترجمه تخصصی
امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
امکان دانلود رایگان ۲ صفحه اول هر مقاله
امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
دانلود فوری مقاله پس از پرداخت آنلاین
پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات