



Have your cake and eat it too? Simultaneously pursuing the knowledge-sharing benefits of agile and traditional development approaches

W. Alec Cram*, Marco Marabelli

Bentley University, Waltham, MA, United States



ARTICLE INFO

Keywords:

Agile development
Traditional development
Knowledge sharing
Ambidexterity
Systems development

ABSTRACT

This research examines how organizations manage knowledge-sharing processes in systems development projects that employ both agile and traditional development techniques. Using a longitudinal case study, we draw on one company's experience with a system implementation that employed a traditional approach during its first phase and then a hybrid, agile-traditional approach in its second phase. By applying an ambidexterity theory lens, we find that the adoption of a hybrid approach allowed the project to continue to exploit the traditional techniques that were working well, abandon techniques that were underperforming, and explore the use of agile techniques in selected areas.

1. Introduction

The increasing popularity of the agile approach to systems development has significantly altered the activity of planning, designing, and implementing software within many organizations by shifting the focus away from a structured, traditional approach to information systems development (ISD) [1–4]. The attention being paid to the agile approach has contributed to growing scrutiny related to the continued use of traditional development approaches (e.g., waterfall), which advocate a more formal, linear development style with a focus on documentation and predefined stages [5–7].

However, it remains unclear if ISD projects perform better using a “pure” systems development approach that closely adheres to one set of principles (e.g., agile or traditional) versus the integration of a collection of diverse techniques together in a hybrid approach (e.g., agile and traditional). Although organizations routinely adapt their ISD approaches from a pure, “by-the-book” interpretation [8–13], projects that employ a mix of agile and traditional techniques are sometimes perceived as being inconsistent in their communication, control, and role assignment [2,14]. Despite this, such hybrid approaches are becoming common for companies with a history of traditional development, but that also have an interest in adopting a more modern, flexible approach [5,13]. Although hybrid approaches can be successful [12], it is a challenge to reach a point where a fusion of agile and traditional techniques can work together effectively [14,2,3,9]. This is particularly difficult when project teams are trained solely in conducting traditional software development, but intend to introduce agile components incrementally into their ongoing program of ISD.

Our research aims to address this important and practical challenge by examining *how* organizations with limited agile experience navigate the transition from a traditional ISD approach to a hybrid, agile-traditional approach. In particular, we focus on the implications for knowledge sharing because of its important role in driving project success [15–17] and because, as the literature clearly indicates, knowledge-sharing processes are conducted very differently on the basis of the ISD approach chosen [2,18–23]. In particular, we focus on knowledge-sharing-related factors that need to be considered by an organization in order to effectively shift from a pure agile approach to a hybrid approach, as these factors have not yet been identified in the literature. Therefore, we pose the research question: *How do knowledge-sharing processes associated with ISD projects change when agile techniques are increasingly used alongside traditional techniques?*

We address our question through the use of longitudinal, qualitative fieldwork at a company where staff had deep experience in traditional approaches, but minimal practical experience with agile. Following a first, unsuccessful implementation of a Customer Relationship Management (CRM) system, management initiated a second project phase (referred to as the “relaunch”), which incorporated selected agile techniques and resulted in a fusion of the traditional and agile approaches (i.e., hybrid). Therefore, it was meaningful to observe the challenges associated with the shift from a traditional to a hybrid approach in a context where the development team needed to learn *how* to manage knowledge-sharing processes during the relaunch. Through a rigorous analysis of our collected data, we were able to identify a series of *factors*, related to organizational mechanisms: *patterns*, associated with the evolution of the adoption of knowledge-sharing processes

* Corresponding author.

E-mail addresses: wgram@bentley.edu (W.A. Cram), mmarabelli@bentley.edu (M. Marabelli).

while transitioning from agile to hybrid, and organizational (internal) and environmental (external) *themes* that are relevant and useful to staff not specifically trained in agile approaches, but who are undertaking an ongoing, incremental shift from a traditional to a hybrid ISD approach.

While our paper provides meaningful relevant practical implications for managers, we also contribute to theory. Building on previous research (e.g., [11,24,3]), we suggest that the ability to manage the conflicting demand of efficiency (by adopting established best practices-related traditional approaches) and flexibility (by being able to manage fast-changing implementation issues typical of agile approaches) can be examined using an ambidexterity lens; our findings suggest that emerging ambidextrous capabilities, here defined as those abilities needed to face the ongoing (and often unpredictable) demand to explore and exploit knowledge, are essential for “learning by doing” in hybrid implementations. In addition, our identification of a series of factors, patterns, and themes related to the increasing use of agile techniques alongside traditional techniques and their impact on knowledge-sharing processes contributes to developing the important links that exist (but that have not yet been formally defined) between the ambidexterity and ISD literature.

Our paper is structured as follows: In the next section, we discuss the research background and theory base. Next, we outline our methodology, including details of the data collection and analysis. The results are then presented and our findings are discussed using an ambidexterity lens. We conclude by presenting a series of future research opportunities.

2. Background and theoretical base

2.1. The transition from traditional to agile ISD

Consistent with past research [25–27], we distinguish between the concept of a development approach (i.e., the high-level goals and principles of systems development, such as agile or traditional), development methodology (i.e., a grouping of guiding development concepts, such as eXtreme Programming or Scrum), and development technique (i.e., a lower-level activity conducted as part of a development project, such as pair programming). Traditional software development approaches are oriented around a predefined, incremental sequence of steps beginning with the analysis of system requirements, followed by the design, development, implementation, and maintenance of the system [6,7]. The benefits of the approach include its straightforward, linear design, and clear milestones that are helpful to manage and monitor the progress of a project [28]. Traditional approaches commonly rely on formal techniques including detailed procedures, output, and approvals, which are shared among team members through a variety of documents related to the project [29].

In comparison, an agile approach draws from a set of principles set forth in the Agile Manifesto [30], including advocating for face-to-face team interactions, collaboration, flexibility to respond to changes, and attention to excellence. Typical development techniques adopted by agile teams include pair programming (i.e., two developers working together on the development and refinement of a piece of code), standup meetings (i.e., short, daily meetings with project team members), story cards (i.e., short descriptions of desired system functionality), planning poker (i.e., a team exercise to arrive at consensus for the amount of effort required for a task), and sprints (i.e., iterative cycles of work typically lasting from 2 to 4 weeks) [31,1].

Although the proportion of organizations adopting a primarily agile approach has risen significantly in recent years, figures also suggest that agile techniques are regularly blended into existing traditional techniques to form a hybrid approach [5,32]. This allows for the customization of low-level development techniques to the needs of the company, while also smoothing the transition away from an institutionalized, traditional development approach. Although past studies have widely examined hybrid approaches [14,2,3,9], this research provides

somewhat “static” findings, which are helpful in understanding why hybrid works in practice, yet unveiling few details into *how* organizations navigate the transition from a traditional approach to a point where they are able to use (and hopefully benefit from) a mix of traditional and agile approaches, especially with respect to knowledge-sharing processes. In particular, it is relevant to understand how organizations with staff not specifically trained in agile are able to transition to such an approach using “learning by doing” techniques; namely, led by an agile savvy project manager, the team faces the challenge of adopting a collection of techniques that are typical of an agile approach. The extremely scant literature in this regard begs the question of *how* organizations transition away from traditional development to adopt a hybrid approach. We attempt to explore this issue by focusing on the dynamic unfolding of practices leading to a shift from the “status quo” (a predetermined, traditional approach) to a situation where a hybrid, agile-traditional approach is undertaken. To this end, we aim to uncover insights related to ongoing knowledge-sharing processes that are associated with such a combination of ISD approaches.

We suggest that it is important to examine this dynamic, focused perspective because organizations routinely experience practical challenges during a transition to hybrid. Particularly when past ISD projects have been rooted in their adherence to techniques such as written documentation (agile minimizes it), unchanging product requirements (agile encourages ongoing changes), and minimal direct contact with customers (agile encourages extensive face-to-face interactions), managers are faced with difficult decisions on what development techniques should remain traditional and what techniques should transition to agile. Companies adopting a hybrid approach are challenged to find a way to balance these seemingly conflicting areas without sacrificing the benefits associated with a “pure” methodology (i.e., where all techniques complement one another); however, by not adhering strictly to either traditional or agile, organizations attempt to gain benefits of both approaches, but may unintentionally make tradeoffs that erode the core principles of both approaches to the point that neither performs effectively. We address this practical challenge, which also reflects a theoretical gap, by examining one key aspect of software development, knowledge sharing, which we discuss next.

2.2. Knowledge sharing in systems development projects: principles and ambidextrous capabilities

Knowledge sharing is viewed as an important set of processes that can contribute to effective systems development projects. It is widely recognized that knowledge-sharing processes vary substantially under a traditional versus agile approach [2,20,33,34,11]. In an ISD context, knowledge sharing refers to the transfer of both tacit and explicit knowledge among project stakeholders. This includes project documentation, user requirements, training, developer interactions, and management guidance. Past research distinguishes between knowledge viewed as an “object” and can be exchanged in written form (i.e., consistent with traditional development approaches), as compared to knowledge that is seen as a “relationship” that is exchanged between project members through daily interactions (i.e., consistent with an agile development approach) [29,2,35–38]. However, the ISD literature points to the challenges associated with both perspectives [34,3]; on one hand, using written documentation represents a straightforward approach, where well-trained developers operate in an efficient manner by relying on consolidated best practices. However, this makes it difficult to manage unforeseen issues and generally does not allow for ongoing changes and departures from planned development strategies, therefore creating rigidity. On the other hand, focusing on informal communication, face-to-face meetings, and knowledge sharing through social practices can create a more flexible and unstructured environment – yet the awareness and ability to adapt to emerging circumstances [39] is required. In fact, while valuable insights have been

Download English Version:

<https://daneshyari.com/en/article/6948815>

Download Persian Version:

<https://daneshyari.com/article/6948815>

[Daneshyari.com](https://daneshyari.com)