Int. J. Production Economics 114 (2008) 723–736

Contents lists available at ScienceDirect

Int. J. Production Economics

journal homepage: www.elsevier.com/locate/ijpe

Shared knowledge and product design glitches in integrated product development

Rauniar Rupak[a,*], Doll William[b], Rawski Greg[c], Hong Paul[d]

[a] Department of Management and Marketing, College of Business, University of St. Thomas, Houston, TX 77006, USA
[b] Department of Management, College of Business, University of Toledo, Toledo, OH 43606, USA
[c] Department of Management, College of Business, University of Evansville, Evansville, IN 47722, USA
[d] Information Operations Technology Management, College of Business, University of Toledo, Toledo, OH 43606, USA

ARTICLE INFO

Article history:
Received 21 April 2007
Accepted 20 March 2008
Available online 3 April 2008

Keywords:
Product glitches
IPD process
Shared knowledge
Project performance

ABSTRACT

Product development processes based on the joint collaboration of the cross-functional team, suppliers, and customers can minimize project glitches. Glitches in the product development project can cause project cost over-runs and delay a project past when first mover advantages are possible. While previous theoretical work has suggested a negative relationship between shared knowledge and product development glitches, empirical studies have not identified how different types of shared knowledge are associated with each other and the design glitches. This study proposes a model of the relationship between specific types of shared knowledge and design glitches in integrated product development (IPD) projects. We test our model using a sample of 191 projects from the automotive industry in the United States. The major findings were that: (1) shared knowledge of the development process can be built by improving a team's shared knowledge of customers, suppliers, and internal capabilities, (2) shared knowledge of the development process for a project reduces product design glitches, and (3) reduced product design glitches improve product development time, cost, and customer satisfaction.

1. Introduction

The competitive business environment requires the design and development of high-quality innovative products that are glitch-free and also mandates that the process of introducing new products to the market be structured and managed appropriately. *Glitches* in product development projects are the differences between the requirements of customers, suppliers, and manufacturing/assembly and the actual deliverables. Such differences in requirements on planned vs. realized means that the particular stage(s) in the product development process failed to deliver its target requirements. Glitches translate into the deficiency of product quality and can hamper both the project and product performance.

In order to improve the project performance, manufacturing firms are increasingly relying on integrated product development (IPD) processes for product development. IPD include some of the best practices such as concurrent engineering (Krishnan and Ulrich, 2001; Roemer et al., 2000), customer involvement (Griffin and Hauser, 1993), supplier involvement (Dowlatshahi, 1998), and the use of cross-functional team (Clark and Wheelwright, 1993). The strategic benefits of IPD have been found to include reducing time (Gupta and Wilemon, 1990; Blackburn, 1991), cutting costs (Hartley, 1990; Handfield, 1994), enhancing quality (Zairi, 1994), effective design of product and process (Rosenthal, 1992), and manufacturability (Swink, 1999).

The key to fast and effective product development is to learn quickly about and shift with uncertain environments

* Corresponding author.
E-mail address: rauniar@stthom.edu (R. Rauniar).

0925-5273/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ijpe.2008.03.005

and create structures accordingly (Eisenhardt and Tabrizi, 1995). In order for team to learn, knowledge that resides among its members need to be shared and then mapped into a shared knowledge of process to exploit this emergent knowledge base. Product development is a form of problem solving (Clark and Fujimoto, 1991; Thomke and Fujimoto, 2000) where experiments are conducted to determine an unknown solution space of process parameters, which optimize or satisfy a set of processing objectives (Pisano, 1996). This experiential learning process may avoid tradeoffs between cost, quality, and customer satisfaction.

The design and execution of highly inter-dependent concurrent tasks in IPD projects to simultaneously meet various goals and expectations can be challenging and the possibilities of glitches are therefore high. The learning challenges (Arlati et al., 1995; Zha et al., 2003) include: (1) poor and inadequate description of the parts, components, and the inter-dependence by the customers, (2) the lack of upfront representation of all the necessary information and knowledge, (3) the inclusion of new and emergent expectations, process improvements, and new technologies during project execution, and (4) lack of standardized, conventional, and unified approach for decision making and problem solving.

Against the challenges and difficulties of IPD projects, Bhuyian et al. (2006) highlight the importance of having a clearly identified IPD process. As a knowledge-intensive activity (Lang et al., 2002; Thomke and Fujimoto, 2000; Sureyskar and Ramesh, 2001), IPD process reflects what the team knows about the customers, products, past successes and failures, complex processes, and handoffs between the functions (Sureyskar and Ramesh, 2001). When the cross-functional team has a shared understanding about customers, suppliers, and their own cross-functional capabilities, project processes can be planned that effectively integrates the inter-dependent team knowledge. This should help the project in minimizing glitches and improve the project performance. Previous studies have focused on the importance of shared team knowledge of customers, internal capabilities, and supplier requirements (Hong et al., 2004). However, we do not know to what extent this shared knowledge of customers, suppliers, and internal capabilities help a team develop a shared knowledge of the product development process that improves multiple measures of project performance.

Although Hoopes and Postrel (1999) and Hoopes (2001) identify the importance of shared knowledge in projects to resolve design glitches, they do not suggest any relationships between types of shared knowledge and glitches. They do not examine whether specific types of shared knowledge (customers, internal capabilities, and supplier) work through a shared knowledge of process to impact project outcomes. Based on the knowledge-based view of product development, this paper examines the relationship between three types of shared knowledge (a shared knowledge of the customers, suppliers, internal capabilities) and a shared knowledge of the IPD processes that is essential to reducing design glitches and improving project performance.

2. Theory development

IPD project process reflects a comprehensive network of work breakdown structures or various stages and their inter-dependencies. Such inter-dependent relationship among various concurrent stages is defined in terms of task inter-dependencies and information inter-dependencies. At each stage of the IPD project, development teams are deeply involved in problem solving and decision making. Individual team members are assigned to the IPD projects because of the functional and technical knowledge, skills, and experiences that is relevant for project execution. Due to the reasons of bounded rationality, individuals in cross-functional IPD team have to rely heavily on the available information and knowledge of other team members. IPD projects are designed and implemented to exploit such individual assets by transforming it into collective team knowledge and strength. Knowledge sharing is therefore an important activity in IPD project environment.

In the current study, we refer shared knowledge as the shared, common understanding of the IPD team. Lang et al. (2002) indicated that successful collaboration in product development project requires cognitive synchronization/reconciliation, developing shared meaning, developing shared memories, negotiation, communication of data, knowledge, information, planning of activities, tasks, and methodologies, and management of tasks. Through shared knowledge, teams develop "transactive memory," in which members have knowledge about who knows what (Moreland, 1999). Shared team knowledge of the project process represents shared mental models of the task domain, procedures, and task inter-dependencies (Lang et al., 2002; Cooke et al., 2000) that are used in the collective problem solving and innovative solution finding (Davenport et al., 1996; Kogut and Zander, 1992). According to Cannon-Bowers et al. (1993), such shared knowledge helps the team to understand the expectations of one another and is directly related to the team performance.

Within the knowledge management literature, researchers (e.g., Alavi and Leidner, 2001; Kim, 1993; Polanyi, 1983) differentiate between *declarative knowledge* and *procedural knowledge*. Declarative knowledge or knowledge on something ("*know-what*") refers to the facts and objects. Within the context of IPD project, it refers to the fragmented, individual's, specialized functional knowledge that is used to complete various technical activities in the project. Procedural knowledge or process knowledge ("*know-how*") concerns the way cognitive process and actions are performed. It is through the improvisations, practices or "doing", that knowledge is best utilized for organizational advancements as in the case of new product development (Orlikowski, 2002). The "cognitive process" facilitates the cross-functional team to trigger different concurrent stages and activities based upon the availability and inter-dependencies of information and knowledge. In addition to know-what and know-how, researchers (for example, Alavi and Leidner, 2001; Zack, 1998) have also emphasized on *conditional knowledge* (or "*know-when*") which refers to the timing and