Information and Software Technology 52 (2010) 680–696

Contents lists available at ScienceDirect

Information and Software Technology

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

Empirical validation of the Classic Change Curve on a software technology change project

Uolevi Nikula [a,*], Christian Jurvanen [b], Orlena Gotel [c], Donald C Gause [d]

[a] *Department of Information Technology, Lappeenranta University of Technology, P.O. Box 20, FI-53851 Lappeenranta, Finland*
[b] *Agricultural Data Processing Centre Ltd., P.O. Box 25, FI-01301 Vantaa, Finland*
[c] *Independent Researcher, New York City, NY 10014, USA*
[d] *Thomas J. Watson School of Engineering and Applied Science, Binghamton University, P.O. Box 6000, Binghamton, NY 13902-6000, USA*

ARTICLE INFO

Article history:
Received 30 October 2009
Received in revised form 5 February 2010
Accepted 9 February 2010
Available online 12 February 2010

Keywords:
Organisational change
Technology change
Classic Change Curve
Technology S-curve
Change resistance
Organisational performance

ABSTRACT

Context: New processes, tools, and practices are being introduced into software companies at an increasing rate. With each new advance in technology, software managers need to consider not only whether it is time to change the technologies currently used, but also whether an evolutionary change is sufficient or a revolutionary change is required.

Objective: In this paper, we approach this dilemma from the organizational and technology research points of view to see whether they can help software companies in initiating and managing technology change. In particular, we explore the fit of the technology S-curve, the Classic Change Curve, and a technological change framework to a software technology change project and examine the insights that such frameworks can bring.

Method: The descriptive case study described in this paper summarizes a software technology change project in which a 30-year old legacy information system running on a mainframe was replaced by a network server system at the same time as the individual-centric development practices were replaced with organization-centric ones. The study is based on a review of the company's annual reports, in conjunction with other archival documents, five interviews and collaboration with a key stakeholder in the company.

Results: Analyses of the collected data suggest that software technology change follows the general change research findings as characterized by the technology S-curve and the Classic Change Curve. Further, that such frameworks present critical questions for management to address when embarking on and then running such projects.

Conclusions: We describe how understanding why a software technology change project is started, the way in which it unfolds, and how different factors affect it, are essential tools for project leaders in preparing for change projects and for keeping them under control. Moreover, we show how it is equally important to understand how software technology change can work as a catalyst in revitalizing a stagnated organization, facilitating other changes and thereby helping an organization to redefine its role in the marketplace.

1. Introduction

Moore's Law predicts that the number of transistors on a chip will double about every 2 years [36]. The resulting improvement in computer performance has contributed to many changes in information systems, development tools, and practices, making software development managers wonder whether it is time to change their existing technology solution with every new advance that occurs. In the face of continuous technology change, decision makers are consequently faced with the age-old dilemma of either continuing to refine their existing technology at the evolutionary level or with adopting an entirely new technology at the revolutionary level. The former strategy introduces minimal organizational change and, as a consequence, appears to be an immediate and low risk approach. The latter strategy, by contrast, has the potential to introduce large changes into an existing organization and thus incurs more immediate risks, but it can potentially attain higher benefits in the longer-term.

Human response to change has been a formal topic of interest since the Second World War, when the need to send men to the battlefield required changes at home as women entered the workplace [26]. The seminal studies from this period defined change as a simplistic process consisting of unfreezing, moving, and freezing steps [44], and began to identify common change resistance

[*] Corresponding author. Tel.: +358 5 621 2839; fax: +358 5 621 2899.
E-mail addresses: uolevi.nikula@lut.fi (U. Nikula), christian.jurvanen@mloy.fi (C. Jurvanen), olly@gotel.net (O. Gotel), dgause@stny.rr.com (D.C Gause).

0950-5849/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2010.02.004

indicators in the workplace, including grievances, high turnover rates, low efficiency levels, and restriction of output [19]. More recently, the role of change has attracted increasing interest in software development circles and it has been studied, for example, from the change management [3,43,46] and motivational [63] viewpoints. However, in the software process improvement context, change has not been studied as extensively to date. Stelzer and Mellis [71] note that in their literature study of success factors of organizational change in software process improvement, unfreezing the organization was mentioned only in 24% of the ISO cases and in 52% of the CMM cases, quality initiatives from the International Standards Organization and Capability Maturity Model process framework respectively. Allison and Merali [2], on the other hand, report a structurational analysis of process improvement in a software package organization over a 10-year period focusing on the contextual and social factors of the changes.

Given its intent to characterize the general change process, our goal in this present study is to validate the Classic Change Curve [66] empirically on a software technology change project. In particular, we are interested in exploring the reasons for initiating software technology change, understanding how this kind of change project unfolds, and determining those factors that affect the project unfolding. Since the present change project started as a technology change project but ended up dealing with organizational changes, we first present the technology and organizational change research frameworks used in the study (Section 2). The actual study started by reviewing the annual reports of the company, and the questions raised were further discussed in five interviews with company employees as well as with a key stakeholder of the company (Section 3). The case study approaches the research questions from the point of view of a legacy information system redevelopment project in which a 30-year old mainframe system was replaced with a network server system and where systematic software development practices were concurrently introduced into the company concerned (Section 4). We cover both the software technology change project and the sustained software process improvement phase that occurred thereafter to see how the company evolved over a 10-year period and emerged from the "Death Valley" of Change [26]. The empirical treatment of the research questions is followed by a discussion of each question based on related research findings (Section 5). We close the paper by exploring the implications of this case study and the generalizability of our findings (Section 6).

2. Related research

The related research focuses on defining the key concepts and context of the present study by studying what leads to a technology change, the manner in which a typical change project progresses, and how technology changes can be characterized. The technology S-curve, the Classic Change Curve, and the technological change framework provide simple communication tools for discussing these complex topics. For the purposes of this paper, we define technology in its broadest sense as the processes, tools, and practices that are used in software development.

2.1. Why change a technology?

The need to change a technology becomes apparent when the existing technology reaches a natural limit. Development efforts when undertaking a technology change tend to progress slowly in the beginning but, after all the essential knowledge has been learned, subsequent progress can be quick (Fig. 1). However, at some point in time the improvement rate slows down, making incremental improvements more difficult and more expensive. For example, the number of transistors that an integrated circuit can contain has a physical limit defined by the size of the components, the line width, the size of the chip [27], and the temperature buildup due to chip density. In software development, such physical limits are rare, but legacy information systems have been reported to exhibit aging symptoms such as increasing maintenance costs, limited performance, integration problems, restricted extension possibilities, and little availability of qualified maintenance and development personnel [1,8–10,69]. Since the aging symptoms, individually or jointly, can become the reason to change the system, they are potential limits of the technology currently in existence.

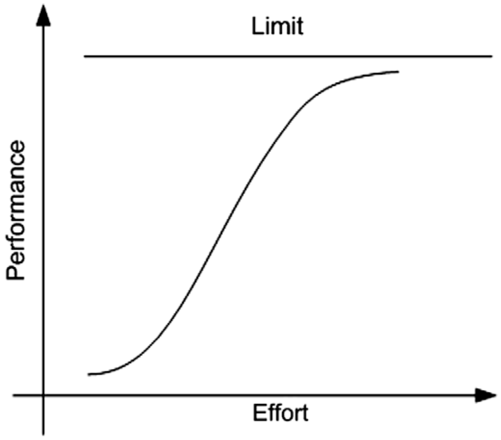

Fig. 1. The technology S-curve [27].

Another reason for a technology change is competition. Competitors using the same technologies should not pose an unmanageable threat, but disruptive innovations can provide entrant companies with an attacker's advantage [27] in a marketplace. A disruptive innovation introduces a new kind of product or service that mainstream customers seldom find interesting, often due to an inferior performance by traditional performance metrics [17]. However, since disruptive innovations are frequently less expensive than mainstream products, they can create new markets and attract mainstream customers as they mature. Examples of disruptive innovations include the introduction of personal computers, which has been claimed as the reason for Digital Equipment Corporation's "abrupt fall from grace" [17] and for none of the independent disk-drive companies of 1976 existing in 1995 [13]. As many incumbent companies have lost their leading market position to entrant companies due to disruptive innovations [13], keeping a close eye on emerging technologies is a key issue for company research and development. This also places an added burden of sponsoring active competitive analysis and market needs determination within those companies competing in rapidly changing technology markets.

A change from one technology to another, where a new technology is studied and adapted in a company context while an old technology remains in production use, typically results in a discontinuity and redundancy (Fig. 2). It has been estimated that leadership changes hands in approximately seven out of ten cases when such discontinuities strike [27] and that two thirds of major technological changes in organizations fail, mainly due to change resistance [47].

2.2. Personal and organizational response to change

Systematic study of change since the late 1940s [44] has resulted in numerous change models. Both personal and organizational level change models exist that explain, for example, the characteristic nature of personal grief [42], workplace morale