

# Chapter XIX

## Competing Commitments Theory

**John McAvoy**

*University College Cork, Ireland*

**Tom Butler**

*University College Cork, Ireland*

### ABSTRACT

*Information system development, like information systems adoption, can be considered to be a change process; yet problems arise when change is introduced. Resistance to the change can develop and the reasoning behind the resistance needs to be determined in order to address it. Resistance can be straightforward, where the change threatens a person's job or creates stress for individuals, yet resistance can also be hidden and complex. Individuals may describe themselves as supporting a change, yet they work against that change (even if they are unaware that they are doing so). When this is happening, competing commitments can be at play; a competing commitment is where an individual professes a commitment to a course of action yet works against that commitment in different, usual subconscious, ways. The competing commitments process is a means of identifying why resistance is occurring even though individuals profess support.*

### INTRODUCTION

Information system (IS) development has been conceptualized as a change process (Lyytinen, 1987); however, the change referred to by Lyytinen refers to the intervention into complex social webs (Kling & Scacchi, 1982) by a project team in the

design and implementation of software artifacts for use in organizational IS. Similarly, the seminal article by (Markus & Robey, 1988) looks at the relationship between information technology and change within organisations. The process by which software artifacts are developed is itself subject to change (Beck, 2000), with the behaviours of

project team members being one such object of the change. Significantly, Lafleur (1996) maintains that change is a constant in a software project, while in a broader context, several authors have described how projects are often used to bring about change: an example of which would be introducing a new ERP system, or a change of work practice (Alsene, 1999; Boody & Macbeth, 2000; Clarke, 1999; McElroy, 1996; Pellegrinelli, 1997; Turner & Muller, 2003).

Much has changed since the 1990s in terms of IS development practice; however, problems still persist. For example, the Standish Group's Chaos Report of 2004 found that over 53% of software projects were challenged, in that they were either over time, over budget and/or the software artifacts lacked critical features and requirements. Another 18% were failures, with the remaining 29% being deemed a success. In explaining the modest improvement since the 1994 survey, Standish Group Chairman Jim Johnson stated that: "People have become much more savvy in project management. When we first started the research, project management was a sort of black art. People have spent time trying to get it right and that has also been a major step forward" (SoftwareMag.com, 2004). The clear implications from this comprehensive industry-based study is that improvements in project management practice have not delivered the necessary improvements in the process and product of IS development. In addition, new design and development programming languages, methods, and techniques have been introduced since the original Chaos Report in 1994 to help solve the 'wicked problems' that plague software project teams. The obvious question is why haven't these improvements worked? The answer may be that all this change to software development processes and practice may not have been as beneficial as is believed. It is clear from extant research that problems arise when change is introduced to project teams. For example, systems developers endeavor to maintain stability and security in the face of change to design and

development processes and procedures (Nader, 1993). They do this because the imposition of change can result in stress and, accordingly, developers endeavor to avoid stressful situations by resisting change (Whitehead, 2001).

This chapter explores the phenomenon of change, and commitment to change, in IS development project teams and theorizes on the underlying factors that shape this complex phenomenon; it then proposes a research method with which to effectively investigate this phenomenon. The goal of this chapter is therefore to identify the difficulties resistance to change brings to many IS projects, and then to describe a method with which to identify the cause of the resistance.

## **BACKGROUND**

The nature of change in IS projects is complex viz. according to Beck (2000, p. 28): "The requirements change. The design changes. The team changes. The business changes. The team members change. The problem isn't change, per se, because change is going to happen; the problem, rather, is the inability to cope with change when it comes". It is clear from Beck that management of change in a software development project is vital. The problem is how is this achieved? Cushway and Lodge (1999) indicate that change is best managed by developing new strategies and structures; they make no mention of the teams and individuals who will effect, and be affected by, change to processes and activities. However, Zmud (1983) argues that trying to implement process change by changing people will lead to resistance: hence, Rainwater (2002) indicates that projects in which the impact of change is not assessed are in danger of running into problems. Clearly, successful change in software development processes and practices in teams will be dependent on several factors (Beck, 2000; Whitehead, 2001). However, if team members resist change, whatever their competences and abilities, then problems ensue.

10 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/competing-commitments-theory/35839](http://www.igi-global.com/chapter/competing-commitments-theory/35839)

## Related Content

---

### Formal Methods Overview

Ana Funes and Aristides Dasso (2015). *Encyclopedia of Information Science and Technology, Third Edition* (pp. 7152-7161).

[www.irma-international.org/chapter/formal-methods-overview/112413](http://www.irma-international.org/chapter/formal-methods-overview/112413)

### An Analytics Architecture for Procurement

Sherif Barrad, Stéphane Gagnon and Raul Valverde (2020). *International Journal of Information Technologies and Systems Approach* (pp. 73-98).

[www.irma-international.org/article/an-analytics-architecture-for-procurement/252829](http://www.irma-international.org/article/an-analytics-architecture-for-procurement/252829)

### From Linguistic Determinism to Technological Determinism

Russell H. Kaschula and Andre M. Mostert (2015). *Encyclopedia of Information Science and Technology, Third Edition* (pp. 4564-4574).

[www.irma-international.org/chapter/from-linguistic-determinism-to-technological-determinism/112898](http://www.irma-international.org/chapter/from-linguistic-determinism-to-technological-determinism/112898)

### Exploration on the Operation Status and Optimization Strategy of Networked Teaching of Physical Education Curriculum Based on AI Algorithm

Yujia Wang (2023). *International Journal of Information Technologies and Systems Approach* (pp. 1-15).

[www.irma-international.org/article/exploration-on-the-operation-status-and-optimization-strategy-of-networked-teaching-of-physical-education-curriculum-based-on-ai-algorithm/316892](http://www.irma-international.org/article/exploration-on-the-operation-status-and-optimization-strategy-of-networked-teaching-of-physical-education-curriculum-based-on-ai-algorithm/316892)

### Autonomic Execution of Web Service Composition Using AI Planning Method

Chao-Qun Yuan and Fang-Fang Chua (2015). *International Journal of Information Technologies and Systems Approach* (pp. 28-45).

[www.irma-international.org/article/autonomic-execution-of-web-service-composition-using-ai-planning-method/125627](http://www.irma-international.org/article/autonomic-execution-of-web-service-composition-using-ai-planning-method/125627)