The Journal of Systems and Software 83 (2010) 574–584

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

An empirical examination of application frameworks success based on technology acceptance model

Gregor Polančič *, Marjan Heričko, Ivan Rozman

University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, 2000 Maribor, Slovenia

ARTICLE INFO

Article history:
Received 16 September 2008
Received in revised form 17 August 2009
Accepted 24 October 2009
Available online 29 October 2009

Keywords:
Application frameworks
Technology acceptance model
Post adoption behavior
Information system success
Survey
Structural equation models

ABSTRACT

Framework-based development is currently regarded as one of the most promising software development approaches when it comes to improvements in lead time, productivity and quality. However, many frameworks and projects based on frameworks still report failures, which indicate that there are problems related to both frameworks technology and frameworks usage. The objective of our research was to examine the major drivers that have an impact on a framework's acceptance and a framework's success. We used the technology acceptance model (TAM) and Seddon's information systems success model as our underlying theory. Data collected from an online survey of 389 active framework users was used to test hypothesized models. Data analysis was performed via structural equation modeling. Our findings support the post-adoption version of TAM and the relationship between continuous use and the successful use of systems, where more use also means more net benefits. We found that the successful use of frameworks is mainly dependent on two factors: continuous framework usage intention and the perceived usefulness of the framework. Several practical and theoretical implications can be foreseen including advances in framework development guidelines and insight into the relationship between the acceptance and success of frameworks.

1. Introduction

Application frameworks (or "frameworks" for short) are a mature technology for reusing software designs and implementations in order to reduce costs and improve the quality of developed software (Mamrak and Sinha, 1999; Morisio et al., 2002b). Frameworks are semi-completed systems that contain certain fixed aspects common to all applications in the problem domain, and certain variable aspects unique to each application made from it (also known as "framework instances") (Srinivasan, 1999). Object-oriented frameworks are the most prevalent. They are defined as "a set of classes that embodies an abstract design for a solution to a family of related problems" (Johnson and Foote, 1988).

Frameworks differ from other reuse techniques, such as components, libraries or design patterns, because they aim to reuse larger grained components and higher-level designs (Fig. 1). Because they define the flow of control, they act as the main program for instantiated applications.

Frameworks play a central role in the software development community (Manolescu et al., 2006), especially when it comes to instantiating software within product lines and product families (Batory et al., 2000; Cunningham et al., 2006). Frameworks also act as an extension of generic programming languages and allow developers to make gains from commonalities in the domain they act (using domain frameworks), development practices they implement (using support frameworks) or applications they develop (using application frameworks).

Besides the positive effects, developing, instantiating and maintaining frameworks continues to be a difficult endeavor (Bosch et al., 1999; Srinivasan, 1999; van Gurp and Bosch, 2001). Due to this, software developers may not decide to develop or use frameworks despite their availability. Or they might develop or use frameworks in an inappropriate way, which leads to project failures.

Because there are problems that make framework development and instantiation difficult, practitioners and researchers have proposed several improvements to them, ranging from documentation improvements (Johnson, 1992), technical improvements (van Gurp and Bosch, 2001) and general improvements for successful framework development and instantiation (Landin and Niklasson, 1995).

While these improvements do stimulate new ideas, most of them are based on personal experiences. Additionally, they do not include proven theoretical foundations and empirical research, which is regarded as one of the main problems of software engineering (Shaw, 1990). Measurement and experimentation for product line engineering has been identified by Frakes and Kang

* Corresponding author. Tel.: +386 2 220 7421.
E-mail address: gregor.polancic@uni-mb.si (G. Polančič).

0164-1212/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2009.10.036

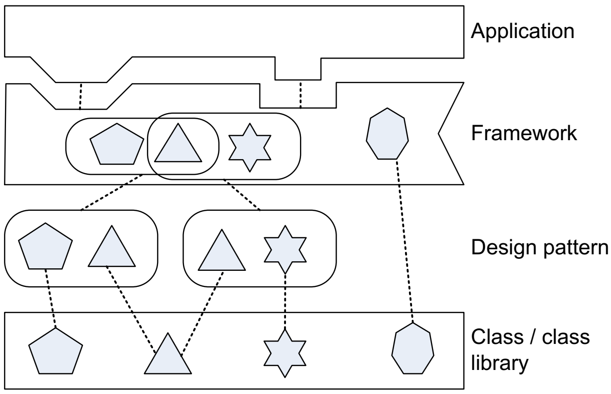

Fig. 1. Framework elements and their relationships (Sangdon et al., 1999).

(2005) as an area where considerably more work is necessary. Secondly, most of the framework improvements are only technologically oriented. However, while frameworks are intensively used by application developers, it is reasonable to incorporate framework developers and users into research on the subject. This supposition is consistent with the results of the Morisio et al. (2002a) study, in which the authors found that human factors have played a focal role in the success of software reuse. Finally, given the importance of frameworks and the extent of their impact on software development projects, there is still a lack of research that identifies and addresses fundamental issues, such as how different framework-related improvements influence frameworks and the perceptions of application developers who instantiate frameworks. However, without a clear understanding of the dynamics of a framework's success, those improvements can only be speculative.

Compared with existing framework research, our research does not propose concrete improvements to frameworks but investigates the major drivers that have an impact on a framework's acceptance and its successful use. We investigate framework acceptance as a special case of IT acceptance where a solid theory with reliable measurement scales has been developed and empirically validated over the past 15–20 years. This theory is commonly expressed in the "technology acceptance model" or TAM. Our research builds on TAM and adapts it to the context of frameworks. The reason for adapting TAM specifically to frameworks instead to all reusable assets is that researchers in previous studies reported problems with "not distinguishing among different types of reusable assets" (Mellarkod et al., 2007).

The original outcome of this article is an empirically validated causal model of factors that impact a framework's acceptance and its successful use. Based on the model, several theoretical and practical implications can be foreseen and are presented in the conclusion of this paper.

2. Theoretical background

2.1. Application frameworks

As presented in the introduction, a framework is a partial design and implementation for an application in a given problem domain. Frameworks are expressed in a programming language, so they commonly consist of a set of cooperating classes and libraries that make up a reusable design. Certain methods of these classes are left unspecified or abstract. In this way they expose details that vary among framework's implementations. A framework instance provides the missing details. It is a pairing of a concrete subclass with each abstract class of the framework to provide a complete implementation. These areas of variability within a framework

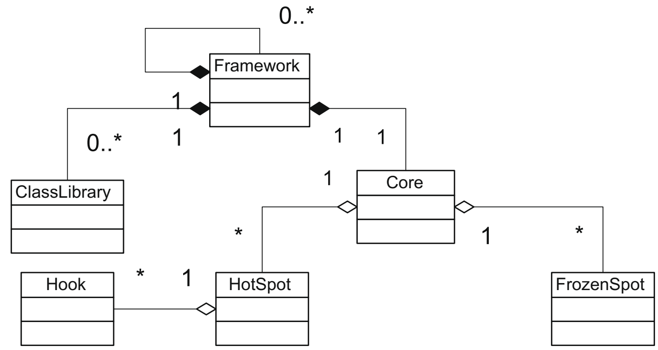

Fig. 2. Framework structure, presented in UML class diagram.

are called hot spots. A hot spot can contain several hooks, which represent actual places (methods) in the framework that can be adapted or extended in order to provide application specific functionality (Fig. 2).

In contrast, framework's frozen spots capture the commonalities across framework instances. These remain unchanged in any instantiation of the framework. In frozen spots, responsibility, collaboration and thread of control are defined. These are commonly expressed with design patterns (Froehlich et al., 1998).

Despite the problem domain knowledge and large-scale-reuse offered by framework-based development, application development based on frameworks continues to be difficult. The framework user must first understand the complex class hierarchies and object collaborations embodied in the framework to use the framework effectively. Besides, frameworks are particularly hard to document since they represent a reusable design at a high-level of abstraction implemented by the framework classes.

To improve the interaction between frameworks and their users, numerous framework development guidelines have been proposed (Mattsson, 2000). For example, a catalogue of 71 framework-related guidelines has been defined by Landin and Niklasson (1995). These guidelines impact design, implementation and use of frameworks. However, the questions remain how these guidelines impact frameworks, their users' perceptions and project outcomes.

2.2. Information technology acceptance

Understanding what influences users to use specific technology, is a major issue in the IT success area (Sharp, 2007). Within the theories which examine the acceptance and use of IT, Davis's (1989) Technology Acceptance Model (TAM), remains one of the most cited, validated and often used theoretical models (King and He, 2006). As demonstrated with solid arrows in Fig. 3, a key assumption of TAM is that external variables (EV) influence the decision to use particular IT only indirectly through their impact on users' beliefs: i.e. the perceived ease of use (PEOU) and perceived usefulness (PU). These two beliefs both influence users' attitude towards using IT (ATU). ATU sequentially has influence on which behavioral intention to use (BI), which is the key factor in determining IT use (U). And, as Tonella et al. (2007) stated, TAM factors must be fulfilled to efficiently exploit new tools.

The solid arrows in Fig. 3 show the initial TAM relationships as introduced by Davis. The dotted arrows show the relationships investigated by other researchers, whereas the values on the arrows show the results of TAM meta-analysis, which was performed by Legris et al. (2003). The values on arrows indicate following: (1) number of significant positive relations identified, (2) number of non-significant relations identified, (3) number of significant negative relations identified and (4) number of untested relations.