Developer-Led Adoption of Open Source Software Libraries: A Conceptual Model

Russell Torres

Information Technology and Decision Sciences, University of North Texas, Denton, TX, United States, russell.torres@unt.edu

Follow this and additional works at: http://aisel.aisnet.org/amcis2012

Recommended Citation
http://aisel.aisnet.org/amcis2012/proceedings/AdoptionDiffusionIT/7

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2012 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
Developer-Led Adoption of Open Source Software Libraries: A Conceptual Model

Russell R. Torres
University of North Texas
russell.torres@unt.edu

ABSTRACT
Researchers have historically viewed Open Source Software (OSS) as a homogenous group of technologies and assumed adoption behaviors are uniform among OSS types. Focus on specific OSS types is rare and OSS libraries, in particular, have been underrepresented. This paper recognizes the unique characteristics of OSS libraries and notes that their adoption may receive less organizational scrutiny than other software packages. A conceptual model based on an augmented theory of Task Technology Fit (TTF) is offered to provide insight into OSS library adoption. If supported, this model represents not only a theoretical contribution to the OSS literature, but a source of actionable information for practitioners engaged in software development.

KEYWORDS
Open Source Software, Adoption, Software Development, Software Libraries, Task Technology Fit

INTRODUCTION
The emergence of Open Source Software (OSS) has created opportunities for businesses around the world. The Linux platform continues to gain strength as a competitive alternative to commercial operating systems (Lee and Cole, 2003). Similarly, a number of fully-functional open source applications are present in the market place and, in some cases, represent the market leader. For instance, a November 2011 Netcraft survey shows the Apache HTTP Server owns 65% of the web server market and surpasses its closest competitor, Microsoft Internet Information Server (IIS), by over 260 million servers (Web server survey, 2011).

For organizations seeking a pre-packaged solution, the low total cost of ownership and high quality of OSS applications makes a compelling case for their inclusion in the enterprise strategy (Watson, Boudreau, York, Greiner, and Wynn Jr, 2008). OSS libraries are also used in the enterprise setting and have many of the same benefits as their OSS application counterparts. Libraries, also known as components, packages, or modules, are bundles of code which encapsulate related functions. Examples include Hibernate, Spring, and the various sub-projects which comprise the Apache Commons. By utilizing a pre-developed, pre-tested software component, development effort is reduced. High quality components can be incorporated into existing development activities with minimal effort and active OSS communities minimize defect resolution times.

Despite the numerous benefits of OSS libraries, they also present challenges due, in large part, to the way in which they are introduced into the organization. The introduction of operating systems and enterprise-class applications is typically subject to some level of organizational approval (Benjamin, Dickinson Jr, and Rockart, 1985). Visibility is usually achieved through either purchasing or acquisition approval, enterprise architecture review, or approval by organizations responsible for support of the product. OSS libraries are unique in this respect because they may be introduced by individual developers without explicit notification to organizational leadership. Without a governance strategy capable of managing all aspects of the adoption process, businesses are fully exposed to well-documented issues associated with the use of OSS. In addition to the possible inclusion of software which is not in alignment with the organization’s technology vision, uncontrolled adoption can lead to both licensing and intellectual property issues (Kubelka and Fawcett, 2006). In this way, developer-led adoption of open source libraries introduces both technical and legal risks.
This paper contributes to open source adoption literature in two ways. First, it recognizes the unique nature of libraries in the OSS ecosystem and focuses attention on their idiosyncratic challenges. Second, it provides insight into the phenomenon of open source adoption at the, heretofore underrepresented, individual level.

The rest of this paper is organized as follows. First, a review of relevant prior work in information systems research on the topic of open source adoption is conducted. Second, theoretical foundations are established and a conceptual model is presented including arguments supporting hypothesized relationships. Finally, a conclusion is offered with implications for theory and practice as well as directions for future research.

LITERATURE REVIEW

Due in part to its low cost and dynamic distribution process, OSS has generated great excitement in the marketplace (Watson et al., 2008). Gartner predicts that by 2016, 99% of the Global 2000 enterprise mission-critical portfolios will contain OSS components (Natis, Weiss, Driver, Wurster, Prentice and Igou, 2010). In parallel with practitioner interest, academic interest has also risen steadily (Hauge, Ayala, and Conradi, 2010; Koch, 2005). As a result, the extant literature on open source software is voluminous (Aksulu and Wade, 2010).

Aksulu and Wade (2010) provide an excellent review of the current state of the field as well as offer a taxonomy of current work. Within this taxonomy, it is the OSS DIFFUSION pattern and specifically the OSS ADOPTION – GENERAL, OSS ADOPTION – BARRIERS, and OSS ADOPTION – DECISION FACTORS categories which are most relevant to the present paper. Articles coded by Aksulu et al. as well as subsequent works matching the characteristics of papers in these taxonomic classifications were reviewed for applicability.

Two dimensions of OSS adoption literature are of interest in the context of the current paper: Adopter level and OSS type.

Adopter Level

The existing research literature has examined the issue of adoption at multiple adopter levels. On the upper end of that spectrum, several studies have examined OSS adoption at the national or regional level. Ghosh (2003) and May (2006), for instance, examine the benefits of OSS adoption by developing nations, while others have examined the impact of national culture on the OSS adoption process (Pykalainen, 2008).

Despite this global interest, current research has tended to focus much more heavily on adoption at the industry or market level. Higher education (Glance, Kerr, and Reid, 2004; van Rooij, 2007, 2009) and Government (Applewhite, 2003; Cassell, 2008; Voth, 2003) adoption of OSS are common in this stream of research due, in part, to their common budget constraints and desire for freedom from vendor lock-in. The disruptive influence of OSS adoption (Brydon and Vining, 2008) and the alteration of the competitive landscape of the software development market (Lee and Mendelson, 2008; Ven and Mannaert, 2008) are also common themes within this research area.

While less represented than adoption at industry or market levels, organizational adoption of OSS is also well researched. Much of the focus within this research area has been aimed at OSS adoption drivers and considerations. Castelluccio (2008), for instance, posits six enterprise adoption drivers for open source: Agility and scale, quality and security, breaking vendor lock-in, cost, sovereignty, and innovation. Other authors have examined important decision factors for business consideration, recognizing that OSS adoption has both costs and benefits (Goldsborough, 2005). It is within this context that the significant issues of OSS licensing (Kubelka and Fawcett, 2006) and maintainability (Yu, Schach, Chen, Heller, and Offutt, 2006) are brought to light. Numerous case studies and other qualitative research approaches have also sought to explicate enterprise OSS adoption (Huysmans, Ven, and Verelst, 2008; Miralles, Sieber, and Valor, 2006; West and Dedrick, 2006).

The individual adopter level has received relatively little attention by the academic community; however, some examples exist. Levesque (2004) discusses several flaws in OSS development which are believed to have prevented user adoption, but no empirical basis for these observations is offered. Others have examined the role of managers in the OSS adoption process (Goode, 2005). While Goode does examine the individual decision-making process, the research aim is clearly more focused on enterprise adoption. As such, managerial adoption behaviors are used as proxies for organizational adoption. Dahlander and McKelvey (2005) examine individual OSS use but focus primarily on reasons users do not become active OSS contributors. Developer-led adoption of OSS has received some scrutiny (Sohn and Mok, 2008), but has not been viewed through the lens of the wider organizational context.
Jin, Robey, and Boudreau (2007) observe that open source use is underrepresented in the literature and call for research with a 4-part research agenda: Creation of OSS user communities, characteristics of OSS user communities, contributions by members of OSS user communities, and change and evolution of OSS user communities. This paper answers that call by establishing a theoretical model of developer-led open source adoption which is a precondition to the creation of OSS user communities.

**OSS Type**

Orlikowski and Iacono (2001) argue eloquently for the need to recognize that IT artifacts are made up of components and warn against “talk of IT artifacts as if they were of a piece—whole, uniform, and unified” (Orlikowski and Iacono, 2001, p. 131). Such recognition is clearly valuable from a research perspective as aggregation disguises nuances which may be overlooked without deeper scrutiny. Unfortunately, many authors in the OSS adoption space have fallen into this trap by either aggregating various OSS technologies or simply failing to specify which OSS technology is the focus of their research. In either case, the lack of focus implies homogeneity within OSS technologies and suggests the authors view OSS adoption as an all-or-nothing proposition. For instance, Choi, Kim, and Eldomiaty (2005) discuss the role of social conventions in the context of OSS diffusion and mention OSS ranging from operating systems and applications, to the open source tools used to support these products. While some level of commonality surely exists among these related OSS types, the opportunity to describe the type-specific, idiosyncratic aspects of adoption was missed. This lack of focus within OSS research has been noted previously and attributed to the immaturity of OSS research (Hauge et al., 2010).

There are, however, many scholarly works which focus intently on particular types of OSS technology. Among these, authors have tended to focus on either operating systems (OSs) or applications. OS-centric studies have focused primarily on Linux. Such focus is likely due to the widespread success of the Linux platform among practitioners (Web server survey, 2011), which elevates its research importance among scholars. Examples of this research focus include Pykalainen’s (2008) study on the effect of culture on the adoption of Linux SSL servers, as well as West and Dedrick’s (2006) qualitative study on moderators of organizational adoption of Linux servers.

Other authors have examined the adoption of OSS applications. OSS applications vary widely in function and complexity, including everything from office suites aimed at individual productivity (e.g., OpenOffice.org, LibreOffice, etc.) to enterprise level content management solutions (e.g., Drupal, Joomla, Alfresco, etc.). Huysmans et al. (2008), for instance, review reasons for non-adoption of OpenOffice.org within the Belgian Public Service using a case study methodology. Goth (2005) reports the dramatic increase in adoption of the Eclipse Integrated Development Environment (IDE) in another example of works in this area of adoption study.

Despite it being among the most popular ways companies adopt OSS, empirical research on the use of open source libraries is sparse (Hauge et al., 2010). Ajila and Wu (2007) find that cost reduction, productivity gains, and quality improvements can be attained through the systematic integration of OSS components. In a descriptive study, Chen, Li, Ma, Conradi, Ji, and Liu (2007) examine the use of OSS components by Chinese software development companies and find that significant cost is involved in learning the details of included libraries. Li, Conradi, Sylngstad, Bunse, Torchiano, and Morisio (2006) discuss the decision making process employed by organizations as they evaluate OSS and commercial components for inclusion into their software products and find cost and quality concerns are significant concerns. While not exclusively adoption focused, Ven and Mannaert (2008) examine the use of OSS components by independent software vendors (ISVs) and conclude their best strategy after source code modification is to contribute back to the open source community.

**Dimensions of OSS Adoption**

The dimensions of OSS adoption described above provide a convenient method to examine the landscape of existing scholarly work. The matrix presented in Table 1, provides example papers categorized based on these dimensions and provides the context in which the present paper is provided.
<table>
<thead>
<tr>
<th>OSS Adoption Level</th>
<th>OSS Type</th>
<th>Multi-Type / Non-Specific</th>
<th>Operating Systems</th>
<th>Applications</th>
<th>Libraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Industry / Markets</td>
<td>Applewhite (2003); Bonaccorsi and Rossi (2003); Cassell (2008); Choi et al. (2005); Cromie and Ewing (2009); Glance et al. (2004); Hedgebeth (2007); Lee and Mendelson (2008); Lin (2008); Pfaffman (2008); Raymond (2001); Scheer (2007); Schultz (2002a, 2002b); van Rooij (2007, 2009); Voth (2003); Wang and Wang (2001);</td>
<td>Varian and Shapiro (2003)</td>
<td>Brydon and Vining (2008); Goth (2005)</td>
<td>Ven and Mannaert (2008);</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Dimensions of OSS Adoption
THEORETICAL DEVELOPMENT

In order to clearly explicate the phenomena of developer-led open source adoption, the present study draws upon the theory of Task Technology Fit (TTF) (Goodhue and Thompson, 1995) for its theoretical basis. It has been noted that TTF is, to some extent, still evolving (Dishaw and Strong, 1999). As such, Figure 1 below illustrates the relationships among concepts commonly held to exist within TTF.

While TTF is often envisioned as a theory designed primarily to describe performance benefits, its underlying assumption that performance is attained through use makes it effective for studying adoption phenomena as well. As described above, TTF suggests that adoption behaviors and performance impacts are influenced by the level of alignment which exists between the characteristics of the technology and those of the tasks for which the technology is intended to be used. It is the antecedents to utilization—task characteristics and technology characteristics as mediated by fit—which are of primary interest in the context of the present paper.

Adoption phenomena are often examined through the lens of either the Technology Acceptance Model (TAM) (Davis, 1989) or the Unified Theory of Acceptance and Use of Technology (UTAUT) (Venkatesh, Morris, Davis, and Davis, 2003). However, the author believes TTF provides a better theoretical basis for this study due to its explicit recognition of technology characteristics. Unlike many adoption scenarios studied in IS, the users of OSS libraries are software developers and thus highly technically skilled. This implies the decision-making process may be more technically focused, and while concepts such as perceived ease of use (in TAM) and effort expectancy (in UTAUT) may be factors, they are likely to be considered explicitly in the context of the technology characteristics of the OSS library in question.

Research Model

In order to provide a more comprehensive, richer view of adoption phenomena, TTF may be augmented to include additional antecedents to utilization. Figure 2 presents a research model based on an augmented theory of TTF which includes task characteristics, technology characteristics, individual characteristics, and context.
Figure 2. The Research Model

Task Characteristics
Tasks have been broadly defined as “the actions carried out by individuals in turning inputs into outputs” (Goodhue and Thompson, 1995, p. 216). In the context of the present study, the primary task is software development. As individual developers evaluate whether an OSS library should be employed or if they should develop code from scratch, the characteristics of the specific development task at hand is likely to be significant within that cognitive process.

Software development efforts are often marked by tight timelines. As one of the primary motivating factors for the reuse of code is reduction of development effort (Basili, Briand, and Melo, 1996), it is logical to conclude that as task complexity increases developers seek out pre-built components in order to improve productivity. Additionally, OSS libraries are often designed to address a specific operational goal (e.g. logging, error handling, etc.). Software development efforts that are unique in nature are less likely to be able to take advantage of OSS libraries. As such, the concept of task characteristics is operationalized as perceived task complexity and perceived task uniqueness and the following hypotheses are offered:

H1a: Perceived Task Complexity influences Fit
H1b: Perceived Task Uniqueness influences Fit
H2a: Perceived Support Availability influences Utilization
H2b: Perceived Quality influences Utilization
H2c: Perceived Ease of Use influences Utilization
H3: Fit influences Utilization
H4a: Individual Characteristics (Software Development Experience) influences Fit
H4b: Individual Characteristics (Organizational Experience) influences Fit
H4c: Individual Characteristics (OSS Familiarity) influences Fit
H4d: Individual Characteristics (Perceived Control) influences Fit
H5a: Context (OSS Policies/Governance) influences Fit
H5b: Context (Permissive OSS Climate) influences Fit
H1a: Increased perceived task complexity has a positive effect on fit.
H1b: Increased perceived task uniqueness has a negative effect on fit.

Technology Characteristics

As previously mentioned, the term OSS represents a diverse set of technologies, capabilities and standards. A review of the Apache Commons project, which serves as a repository of reusable Java components, reveals 42 active projects with another 16 in pre-production, developmental stages (Apache commons, 2011). Unlike Apache, Sourceforge doesn’t provide a separate project specifically for libraries, but a review of the site reveals they currently support over 300,000 projects. Many of these can be assumed to be open source libraries or components. When combined with the myriad other homes for OSS libraries on the web, the number of projects that provide reusable code components is staggering. Each project and, by extension, component varies based on a variety of characteristics which may influence the decision to adopt.

Several technology characteristics deserve consideration due to their prevalence in the OSS and adoption literature. For instance, the quality and availability of OSS support is a topic frequently addressed in the literature (e.g., Bezroukov, 1999). Similarly, authors have also studied the quality of OSS as compared with commercial offerings (e.g., Stamelos, Angelis, Oikonomou, and Bleris, 2002). Combined with the frequently used perceived ease of use (Jeyaraj, Rottman, and Lacity, 2006) technology characteristic, these elements are employed to operationalize the technology characteristic concept and the following hypotheses are offered:

H2a: Increased perceived support availability has a positive effect on fit.
H2b: Increased perceived quality has a positive effect on fit.
H2c: Increased perceived ease of use has a positive effect on fit.

Fit

Central to the theory of Task Technology Fit, the fit construct “is the degree to which a technology assists an individual in performing his or her portfolio of tasks” (Goodhue and Thompson, 1995, p. 216). While fit has been conceptualized in many ways (Dishaw and Strong, 1999), the present study adopts the view that fit represents the interplay or interaction between task characteristics and technology characteristics and offers the following hypothesis:

H3: Increased fit has a positive effect on utilization of OSS libraries.

Individual Characteristics

While not present in Figure 1, others have noted that the inclusion of individual characteristics into TTF is common (Goodhue and Thompson, 1995) and consistent with the theories on which it is based (Dishaw and Strong, 1999). Goodhue and Thompson note that “characteristics of the individual (training, computer experience, motivation) could affect how easily and well he or she will utilize the technology” (Goodhue and Thompson, 1995, p. 216), and include individual characteristics in their Technology-to-Performance Chain. When viewed in the context of numerous adoption focused studies (e.g., Technology Acceptance Model (Davis, 1989)), it is logical to conclude that individual characteristics are an important contributor to adoption behaviors.

Developer experience has been previously found to influence the decision to reuse code (Hoadley, Linn, Mann, and Clancy, 1996). As code re-use is a key element of OSS libraries, experienced developers may favor their inclusion in software development projects. In addition to development experience, an individual’s organizational experience may also play a role in OSS adoption. Individuals with longer organizational tenure may be more willing to make OSS adoption decisions due to increased levels of social capital within the firm. Familiarity with OSS is also a likely determinant for adoption. Fitzgerald (2006) notes the transition of OSS from a host of fragmented offerings to more commercially viable forms. As such, developers may be aware of OSS either due to previous use or through brand awareness (e.g., Apache Software Foundation, Spring Source, etc.) as more organized OSS ventures have emerged. Another possible antecedent to adoption behavior is perceived control. Developers that feel empowered to make decisions regarding software, and specifically OSS, will be more...
likely to employ it as part of the development process. The influence of these individual characteristics on adoption behaviors is summarized in the following hypotheses:

H4a: Increased software development experience has a positive effect on utilization of OSS libraries.
H4b: Increased organizational experience has a positive effect on utilization of OSS libraries.
H4c: Increased OSS familiarity has a positive effect on utilization of OSS libraries.
H4d: Increased perceived control has a positive effect on utilization of OSS libraries.

Context

The context in which individuals operate is another important determinant of adoption behaviors. Again, while not depicted in Figure 1, Goodhue and Thompson (1995) do theorize additional precursors to utilization. These include, among other things, “facilitating conditions” and “social norms.” Elements of social and organizational context such as goals, norms, policies and procedures are all relevant to the adoption decision-making process. Social norms, for example, has been identified as one of the most frequently utilized independent variables in IS adoption research (Jeyaraj et al., 2006). The inclusion of context therefore is a natural extension of TTF in the study of adoption phenomena.

The presence or absence of OSS-specific policies or governance mechanisms may influence developer intention to use OSS. This may be through either outright prohibition of OSS or through perceptions that the “red-tape” is too burdensome to deal with. Similarly, the perceived OSS climate within the organization may exert normative pressure on developers either towards or away from the use of OSS technologies. Organizational context is therefore operationalized in the following hypotheses:

H5a: OSS policies/governance has a negative effect on utilization of OSS libraries.
H5b: A perceived permissive OSS climate has a positive effect on utilization of OSS libraries.

CONCLUSIONS

The goal of this paper is to improve understanding of the factors that drive developer-led adoption of OSS. This goal is achieved through the creation of a conceptual OSS adoption model based on TTF and augmented with constructs designed to capture task, technology, individual, and contextual characteristics.

In addition to extension of TTF, this study makes two important theoretical contributions. First, by focusing on OSS libraries, it addresses ambiguities that have been present in a number of other OSS adoption studies. This focus is justified given the unique nature of libraries within OSS and the idiosyncratic process associated with their adoption. Second, by focusing on individual adoption determinants, this study provides a foundation on which future empirical OSS adoption research may build.

Future research should include an empirical test of the research model in the context of organizations engaged in object-oriented software development. It may be interesting to examine the model’s performance in the context of several types of organizations. For instance, IT consulting or software companies may have different organizational objectives and may exhibit different patterns of developer-led adoption.

If supported, the model will provide increased understanding of OSS adoption phenomena and allow practice to improve governance strategies and management of OSS libraries. In addition to improving the effectiveness of policies aimed at managing the introduction of OSS into organizations, this work may also provide insights into the development of tools to improve organizational visibility and control of OSS adoption.
REFERENCES