Investigating Determinants of Software Developers' Intentions to Follow Methodologies

BILL C. HARDGRAVE, FRED D. DAVIS, AND
CYNTHIA K. RIEMENSCHNEIDER

BILL C. HARDGRAVE is the Edwin & Karlee Bradberry Chair in Information Systems, and Executive Director of the Information Technology Research Center in the Walton College of Business at the University of Arkansas. His research on software development has appeared in *Journal of Management Information Systems, Communications of the ACM, IEEE Software, IEEE Transactions on Software Engineering, DATA BASE for Advances in Information Systems, Information & Management, Journal of Systems and Software,* and *Educational & Psychological Measurement,* among others.

FRED D. DAVIS is Professor, David D. Glass Chair, and Department Chair of Information Systems at the Walton College of Business, University of Arkansas. His research on user acceptance of information technology and computer-assisted decision-making has appeared in *MIS Quarterly, Management Science, Decision Sciences,* and *Organizational Behavior and Human Decision Processes.*

CYNTHIA K. RIEMENSCHNEIDER is Associate Professor of Information Systems in the Walton College of Business at the University of Arkansas. Her research has appeared in *Information Systems Research, Information and Management, IEEE Transactions on Software Engineering,* and others. She currently conducts research on IT and small businesses, IT adoption, and people in IT.

ABSTRACT: Seeking to improve software development, many organizations attempt to deploy formalized methodologies. This typically entails substantial behavioral change by software developers away from previous informal practices toward conformance with the methodology. Developers' resistance to such change often results in failure to fully deploy and realize the benefits of the methodology. The present research draws upon theories of intention formation and innovation diffusion to advance knowledge about why developers accept or resist following methodologies. Results from a field study within a large organization indicate that developers' intentions are directly influenced by their perceptions of usefulness, social pressure, compatibility, and organizational mandate. This pattern of intention determinants is quite different from that typically observed in studies of information technology tool adoption, revealing several key differences between the domains of tool versus methodology adoption. Specifically, although organizational mandate had a significant effect on intentions, the strength of its direct influence was the lowest among the four significant constructs, and usefulness, compatibility, and social pressure all influenced intentions directly, above and beyond the effects of organizational mandate. The findings suggest, contrary to popular belief, that an organizational mandate is not sufficient to guarantee use of the methodology in a sustained manner.

Journal of Management Information Systems / Summer 2003, Vol. 20, No. 1, pp. 123–151.

0742–1222 / 2003 $9.50 + 0.00.

KEY WORDS AND PHRASES: diffusion of innovations, innovation adoption, software development methodologies, technology acceptance model.

IMPROVING THE PROCESS OF SOFTWARE DEVELOPMENT is frequently identified as a top concern by information systems (IS) executives (e.g., [10]). Software development is not improving as it should: in addition to a growing backlog of IS, quality is declining [22], and only about 25 percent of all development efforts are considered successful [64]. Numerous software development innovations, such as methodologies and computer-aided software engineering (CASE) tools, have been created and promoted to assist firms in developing IS. However, research indicates that many of these innovations are slow to be accepted and have not become widely utilized [23, 24, 36].

Methodologies are formalized processes for developing systems that standardize steps of the development process, which can yield benefits such as increased productivity and higher quality [25, 28, 32, 52]. Converting from informal development practices to a methodology is often a long and arduous journey met with resistance by developers [40, 48]. As a result, "the expenditure of time and effort in implementing a software development methodology makes it one of the most serious areas of concern in IS" [54, p. 647]. Methodologies have not become widely embraced in practice, with only about half of all companies actually following any type of methodology [23]. The difficulty organizations face in successfully deploying methodologies is clearly a problem warranting continued research. To better understand how to deal with this resistance, the present research investigates the determinants of software developers' intentions to follow a methodology.

Previous Research on Methodology Use

LITTLE RESEARCH TO DATE HAS SPECIFICALLY ADDRESSED the determinants of individual software developers' intentions to follow methodologies. In the context of object-oriented (OO) development, Johnson et al. [33] used Ajzen's [4] belief elicitation pretesting procedure to identify specific salient beliefs underlying intention formation within the theory of planned behavior (TPB). The list of beliefs included process usefulness, product usefulness, communication usefulness, career usefulness, extra-organizational pressure, intra-organizational pressure, OO skill, general background, difficulty with OO, receptiveness to OO, complexity of OO, organizational support for OO, OO technical characteristics, and expectations of OO. Johnson et al. [33] did not empirically test the relationships between these elicited beliefs and intention to adopt or use OO development. Khalifa and Verner [37] studied several determinants of software developers' use of two specific methodologies—prototyping and waterfall—explaining 18 percent and 32 percent of the variance, respectively. They found that perceived quality of the development process and team size were significantly related to usage [37].

Several studies have examined individual developer acceptance of software development innovations such as programming languages (e.g., [3, 56]), analysis and design techniques (e.g., [40, 42]), and CASE tools (e.g., [11, 31]). It is important to distinguish between the adoption of specific software development tools and techniques versus the adoption of an entire methodology, since it is possible to use particular tools or techniques in the absence of a formal methodology [17, 62], and because adoption of a methodology represents a much more radical change than adopting tools or techniques [47, 54]. This distinction was demonstrated by Orlikowski [47], who examined two CASE tool adoption environments: the first one had a methodology in place and adopted a CASE tool to support the existing methodology (an incremental change); the second environment had no methodology in place and adopted the information engineering methodology and a CASE tool designed specifically to support it (a radical change). Reactions of various stakeholders differed between the two environments, clearly suggesting the importance of isolating the adoption of a methodology from the adoption of a CASE tool. This lack of isolation may explain why studies of CASE tool adoption have yielded mixed and inconclusive results (e.g., [11, 31]). It is likely that prior studies unintentionally confounded adoption of a CASE tool with adoption of the methodology it is used with. This becomes especially troublesome when the CASE tool under investigation promotes or even enforces a particular methodology that may or may not be aligned with the software practices currently followed by developers facing the new CASE tool. The present study avoids this possible confound by studying the adoption of a methodology in the absence of a CASE tool that may be used in conjunction with it.

Theory and Hypotheses

IT IS IMPORTANT TO DISTINGUISH THE QUESTION OF how *individual* software developers form intentions to follow a methodology from the decision of an *organization* to adopt a methodology across the board. There would naturally be some mutual influence between organizational and individual decisions to adopt a methodology. For example, interest in methodologies among a growing number of individuals within an organization could spur the organization to make a decision to standardize on the methodology in a bottom-up influence process. Conversely, an organizational decision to mandate use of a methodology should have a strong effect on individuals' likelihood of adopting it in a top-down influence process. It is specifically this latter situation that the present research is concerned with—how individual software developers within an organization that has mandated use of a specific methodology form intentions to follow it. Whereas the present study focuses on individual intentions to follow a mandated methodology, it does not ignore the external influences of an organizational mandate or influences from important referents, such as coworkers or consultants. In fact, constructs that represent these influences on individuals' intentions are explicitly included. Research on IS deployment outside the domain of software methodologies has shown that, despite the presence of a strong organizational mandate to use an information technology (IT) tool, resistance by individual users can

completely undermine its successful deployment (e.g., [34, 42]). As suggested by Fichman: "the relative lack of attention to individual adoption of technologies is unfortunate because, while the organization as a whole makes the initial adoption decision for such technologies, the actions of individual adopters (e.g., how enthusiastically they embrace the innovation) can be expected to have a large impact on the implementation process" [19, p. 203].

To understand the key determinants of an individual developer's intention to follow a new methodology being introduced to the organization, we draw upon established theories regarding the nature of intention formation and innovation diffusion in the context of IT tools, and tailor them to the context of following a software development methodology. Two major theoretical paradigms are used: the technology acceptance model (TAM) [15] and diffusion of innovations (DOI) [55]. Although these theories have been successfully used to explain intention to use IT tools, it is unknown how readily they can be adapted to encompass software methodology adoption. Because each of these theories was derived from more general theories of motivated human behavior across a wide range of domains, there is reason to hope that, with appropriate modifications, they may successfully generalize to the methodology context.

Methodologies are phenomenologically distinct from tools: whereas methodologies are specified comprehensive processes and patterns of behavior, tools are artifacts (usually software programs) that individuals may or may not use in performing tasks, independent of the processes within which such tasks may be embedded [17, 29, 41, 47, 62, 63]. Three key differences between the domain of individual IT tool acceptance and the present domain of methodology acceptance among software developers are (1) the degree to which adoption of the target innovation has been mandated by senior management of the organization, (2) the magnitude of behavioral change entailed by adopting the innovation [47], and (3) the greater emphasis on the use of project teams for organizing software development work that increases the relevance of peer influences compared to use of individual IT productivity tools. Therefore, relevant theory should include constructs reflecting the external influences of organizational policies and opinions of professional colleagues on individual intentions to use, as well as individual adopter concerns regarding the extensiveness of behavioral change required. In both domains (tools and methodologies), adopters are expected to be influenced by their own opinions about an innovation's potential effects on job performance and the difficulty in learning or using the innovation—these constructs should also be reflected. TAM and DOI provide suitable theoretical foundations for modeling these intention determinants.

Drawing heavily from theory of reasoned action (TRA) [5], Davis et al. [15] identified and measured a set of generic beliefs that apply across a range of IT tools. The resulting TAM is a parsimonious model with two primary direct determinants of intention: ease of use and usefulness. It should be acknowledged that TAM includes a causal relationship from ease of use to usefulness, so that ease of use has both a direct effect on intention and an indirect effect through usefulness. TAM2, a recent extension to TAM, added subjective norm and voluntariness [61]. TAM/TAM2 is used

extensively in the field of IS for explaining the acceptance of IT tools. The DOI theory [55] argues that "potential users make decisions to adopt or reject an innovation based on beliefs they form about the innovation" [1, p. 90]. This theory has been applied in many different fields (e.g., agriculture and marketing) and used to study a variety of innovations (e.g., spreadsheets, World Wide Web, teaching methods) [55]. In a meta-analysis of DOI, Tornatzky and Klein [59] identified three beliefs about innovations that are consistently relevant: relative advantage, complexity, and compatibility; these three will be used herein. Although DOI has been recommended as a theory to explain acceptance of methodologies and other software engineering innovations [50], to our knowledge no one has followed through with the suggestion. Although TAM and DOI have several conceptual similarities, they each provide distinct elements. Therefore, constructs from each of these models are used to form a unique combination of intention determinants (i.e., the combination does not appear in either of the individual models). The five direct determinants of a developer's intention to follow a methodology considered in this study are: usefulness, complexity, social pressure, compatibility, and organizational mandate.

Perceived Usefulness

Perceived usefulness refers to the degree to which a developer expects that following a methodology will improve his or her individual job performance. According to Davis et al., "within organizational settings, people form intentions toward behaviors they believe will increase their job performance. . . . This is because enhanced performance is instrumental to achieving various rewards that are extrinsic to the content of the work itself, such as pay increases and promotions" [15, p. 986]. In TAM, the usefulness construct captures this concept. Similarly, relative advantage, from DOI, is "the degree to which an innovation is perceived as being better than the idea it supercedes" [55, p. 212]. As a demonstration of the conceptual equivalence of usefulness and relative advantage, Moore and Benbasat [46], in their operationalization of DOI for the IS literature, measured relative advantage the same as TAM's usefulness. Historically, perceived usefulness has a strong positive influence on usage intentions in a tool context (for a recent review, see [60]). In a methodology context, Johnson et al. [33] found several usefulness elements (process usefulness, product usefulness, communication usefulness, career usefulness) when eliciting beliefs about OO development. Khalifa and Verner [37] examined two constructs similar to usefulness (process quality and product quality) as determinants of prototyping and waterfall usage; process quality was significant. Overall, theoretical foundations and prior research suggest that the more an innovation is perceived as enabling an increase in workplace performance, the more likely it is an intention will be formed to adopt it. Therefore, a developer's perceived usefulness of a methodology is expected to positively influence his or her intention to follow the methodology.

H1: Perceived usefulness will be positively related to a developer's intention to follow a methodology.

Perceived Complexity

Perceived complexity refers to the degree to which a software developer regards a methodology as difficult to follow or learn. The less complex a system is, "the greater should be the user's sense of efficacy and personal control regarding his or her ability to carry out the sequences of behavior needed" [15, p. 987], which, in turn, influences motivation. TAM and DOI posit that intention formation is partially determined by how difficult the innovation is to understand or use [15, 55]. All else being equal, the easier something is to use (i.e., the less complex it is), the more likely one is to accept it. The difficulty in learning a new development approach and the perceived complexity of the approach were identified by Johnson et al. [33] as salient beliefs that may influence the acceptance of OO development. In the present domain, a methodology that is overly complex or difficult to understand and follow may be less likely to be accepted.

H2: Perceived complexity will be negatively related to a developer's intention to follow a methodology.

Perceived Social Pressure

Perceived social pressure refers to the extent to which a developer experiences interpersonal influence (to follow a methodology) from important others within his or her social milieu. Although the original TAM did not find the influence of social pressure to be significant, TAM2 did find the construct (called subjective norm) to be a significant direct determinant of intentions when usage was organizationally mandated [61]. According to Venkatesh and Davis: "the rationale for a direct effect of subjective norm on intention is that people may choose to perform a behavior, even if they are not themselves favorable toward the behavior or its consequences, if they believe one or more important referents think they should" [61, p. 187]. Furthermore, as Agarwal suggests, "the attitudes and beliefs of others in groups to which an individual belongs help shape technology usage behavior through overt communication or more subtle forms of suggestion. Social interactions are instrumental in generating shared meaning and mutual understanding in an organization and thereby provide an important basis for subsequent patterns of behavior" [1, p. 96]. Furthermore, changes to development practices can "alter the political and social dynamics within the department and can cause resistance to change" [49, p. 110]. In the field of software development, where IT workers are expected to work closely with others on projects, social pressure may take on more importance than is the case for individual IT tool use. The opinions of coworkers, for example, have been suggested as important in forming one's beliefs about a new development process [33]. Given the emphasis on teamwork in software development and the likelihood that methodologies are organizationally mandated, a direct effect of social pressure on intentions is expected.

H3: Perceived social pressure will be positively related to a developer's intention to follow a methodology.

Perceived Compatibility

Perceived compatibility refers to the degree to which a software developer regards the practice of following a methodology as being consistent with his or her preexisting software development process. Tornatzky and Klein define compatibility as "congruence with the existing practices of the adopters" [59, p. 33]. Raghaven and Chand [50] suggest that methodologies are generally incompatible with the informal development processes commonly used because they require a more systematic process for developing systems. Johnson et al.'s [33] findings suggest that beliefs regarding the compatibility of one's existing skills and background in a particular development approach may influence intention formation toward the approach. In IT tool research, compatibility's role in intentions is inconclusive (e.g., [2, 13, 27, 31]), perhaps due to the lack of, or relatively minor, changes to one's existing work practices caused by the adoption of a tool. It is expected to be important here due to the more radical changes required by methodologies in the way developers perform their jobs.

H4: Compatibility will be positively related to a developer's intention to follow a methodology.

Perceived Organizational Mandate

Perceived organizational mandate refers to the degree to which a developer believes that following a methodology has been dictated by an official policy currently in effect within his or her organization. Organizational mandate (the opposite of voluntariness) was not included explicitly in either TAM or DOI since much of the intention formation and innovation adoption theory was developed in the context of voluntary usage decisions [19]. However, in recognizing that tool use is not always volitional, voluntariness was added to TAM2 to distinguish between mandatory and voluntary usage settings [61]. The omission of voluntariness from DOI has led to criticism for the unclear role of mandates within the theory [9]. To overcome this deficiency, Moore and Benbasat [46] added voluntariness as a key determinant in their perceived characteristics of innovating instrument. Although Moore and Benbasat [46] did not report the relationship between voluntariness and usage, Agarwal and Prasad [2] found a significant negative relationship between the two (i.e., all other things being equal, the more voluntary it is, the less likely it is to be used). In situations where methodologies are accompanied by an organizational mandate, developers are unlikely to regard usage as voluntary. We theorize that, all else being equal, those who regard the methodology as mandatory will be more likely to form intentions to follow it.

H5: Perceived organizational mandate will be positively related to a developer's intention to follow a methodology.

Intentions to Use

The dependent variable in this study is a developer's intention to follow a specified methodology in the future, whether or not he or she follows it currently [2]. Intention

is commonly used as an indicator of the acceptance stage of adoption in DOI [1]. Intention has become the de facto measure for gauging the acceptance of an innovation and has repeatedly proven to be a strong predictor of actual future use. According to Ajzen:

> Intentions are assumed to capture the motivational factors that have an impact on a behavior; they are indications of how hard people are willing to try, of how much of an effort they are planning to exert, in order to perform the behavior. These intentions remain behavioral dispositions until, at the appropriate time and opportunity, an attempt is made to translate the intention into action. [4, p. 113]

Method

Measures

A QUESTIONNAIRE WAS CONSTRUCTED USING measurement scales adapted from previous research (see Appendix A for scales). The behavioral intention items were adapted from standard scales such as those used by Ajzen and Fishbein [5], Venkatesh and Davis [61], and Taylor and Todd [57]. Usefulness and complexity items were adapted from the original Davis [15] instrument. Organizational mandate and compatibility scales were adapted from Moore and Benbasat [46]. Social pressure items came from Venkatesh and Davis [61] and Ajzen and Fishbein [5].

Research Site

An appropriate sample for investigating the active formation of intentions to follow a methodology would consist of developers who are aware of the methodology, have the opportunity to use it, have tried it, and are still in the process of forming stable intentions to follow it or not [2]. As indicated by Fichman [19], individual adopter studies are usually confined to a single organization. Because the focus is on individual acceptance of a methodology, it was necessary to find an organization implementing a specific methodology so that the new methodology served as the common reference for the developers. For this study, a Fortune 1000 organization fitting the aforementioned sample criteria served as the research site.

The IT group in this organization consisted of 330 IT employees divided into two groups: services (145 employees) and applications development (185 employees). The methodology, based on the structured development paradigm, was custom-created for the company's internal use under the leadership of an external consultant. In this case, the organization was moving from an environment of no prescribed processes in place (i.e., no methodology) to an environment guided by an organization-wide methodology, which would be considered a radical change [46]. No other methodology was in use prior to, or during, the period of this study.

The methodology is comprehensive, providing process directions for the complete life cycle. It includes specifications for the creation of deliverables (such as a re-

quirements document) at various project stages by various parties, including users, and is adaptable to many different project types (e.g., small, large; new development, maintenance; legacy systems, enterprise resource planning systems). The exact portions of the methodology, such as producing entity-relationship diagrams or prototypes, to be used by various types of developers, such as programmers and systems analysts, are determined by project characteristics (i.e., the methodology established a contingency model for methodology adaptation). The methodology did not utilize a CASE tool.

To launch the methodology, all IT employees attended a presentation by the external consultant, who provided an overview of the methodology and the process for implementation. Over a six-week period beginning the day after the presentation, all employees were given training and experience on trial projects. Each Friday during this six-week period, review and feedback sessions were held to allow developers to ask questions, make suggestions for improvements, voice concerns, and so on. All employees were provided a hard copy of the methodology documentation (approximately 150 pages) and access to an online version with embedded links to templates (for deliverables). At the end of the six-week training period, all developers were instructed via a formal written policy statement from the chief information officer to begin using the methodology for all projects. Although the aforementioned written policy statement mandated the use of the methodology, no punishments (rewards) for nonuse (use) were expressed.

Only the applications development group was guided by the methodology; the services group was outside the scope of the methodology. Therefore, a questionnaire was distributed to all 185 IT employees reporting to the vice president for application development approximately six weeks after the issuance of the policy statement. The six-week time period was chosen to allow enough time for all developers to have the opportunity to gain experience following the methodology. This time frame also provided relatively quick, yet informed, feedback regarding developer acceptance of the new methodology.

One hundred twenty-eight developers completed the instrument for a response rate of 69 percent. As shown in Table 1, many different types of IT employees, representing the gamut of job functions in this organization, responded to the survey. The average age of the developers was 36 years, and they had a reported average of 10 years of development experience. The average number of years the developers had been with the company was five. Of those participating in the study, 38 percent were female and 62 percent were male. According to employee demographics, respondents were representative of the population and did not suggest a response bias.

Results

STRUCTURAL EQUATION MODELING[1] (SEM) provided a rigorous assessment of the psychometric properties of the measures within the context of a holistic test of the model linking the latent constructs to manifest indicators as well as to the criterion

Table 1. Respondent Demographics

Title	Percent
Programmer analyst	16
Systems analyst	15
Programmer	14
Programmer specialist	14
VP, director, manager	9
Senior programmer specialist	7
Coding specialist, contractor	2
Other	23
Gender	**Percent**
Female	38
Male	62
	Average
Age	36
Years of development experience	10
Years with company	5

(intentions). The psychometric properties are presented in Appendices B, C, and D. Supporting convergent validity, all factor loadings are sufficiently large [8, 65], and the *t*-value of each loading is significant and twice the standard error of the loading [6] (see Appendix B). Discriminant validity among the constructs was confirmed by using the chi-square difference test (Appendix C) [7]. Composite reliabilities range from 0.79 to 0.95, indicating acceptable internal consistency [20] (Appendix D). Standard deviations and intercorrelations of all constructs are also provided in Appendix D.

Table 2 and Figure 1 show model testing results. Of the five hypothesized intention determinants, four were found significant: usefulness ($\beta = 0.46$; $p < 0.001$), social pressure ($\beta = 0.20$; $p < 0.01$), compatibility ($\beta = 0.19$; $p < 0.05$), and organizational mandate ($\beta = 0.19$; $p < 0.01$). These four determinants explained 63 percent of the variance (i.e., R^2) in developer intentions to follow the methodology.

Because complexity was not a significant direct determinant of intentions as hypothesized, the model was respecified to examine the indirect (via usefulness) effect of complexity on intentions as found in many previous tool-oriented TAM studies (e.g., [45, 57, 61]). The relationship between complexity and usefulness was indeed significant ($\gamma = -0.67$). Unfortunately, modifying the model by adding the complexity-usefulness relationship resulted in poor model fit. Although some theorists regard goodness of fit to be overrated as evidence of a theory's validity [53], the presence of incorrect structural assumptions in a proposed model can distort estimates of those parameters within a model that are correctly specified [16]. One response to such trade-offs often employed is specification search [44], in which one or more parameters are added to a model, if theoretically plausible, to improve goodness of fit. In

Table 2. Results: Determinants of Intentions to Follow Methodologies

Variable	β	Significance	Hypothesis	Outcome
Usefulness	0.46	0.000	H1	Supported
Complexity	–0.04	0.605	H2	Not supported
Social pressure	0.20	0.006	H3	Supported
Compatibility	0.19	0.02	H4	Supported
Organizational mandate	0.19	0.002	H5	Supported
R^2	0.63			

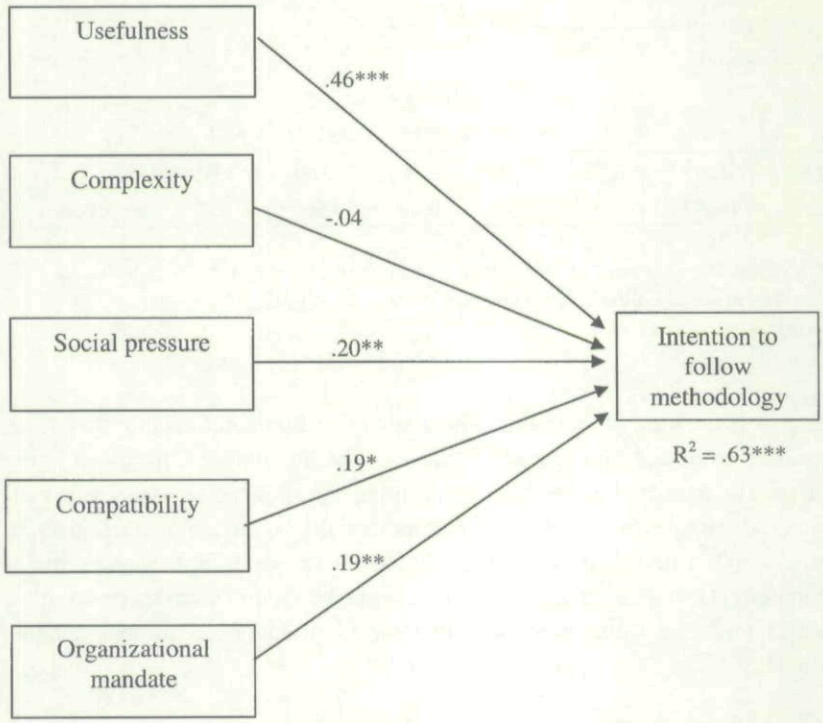
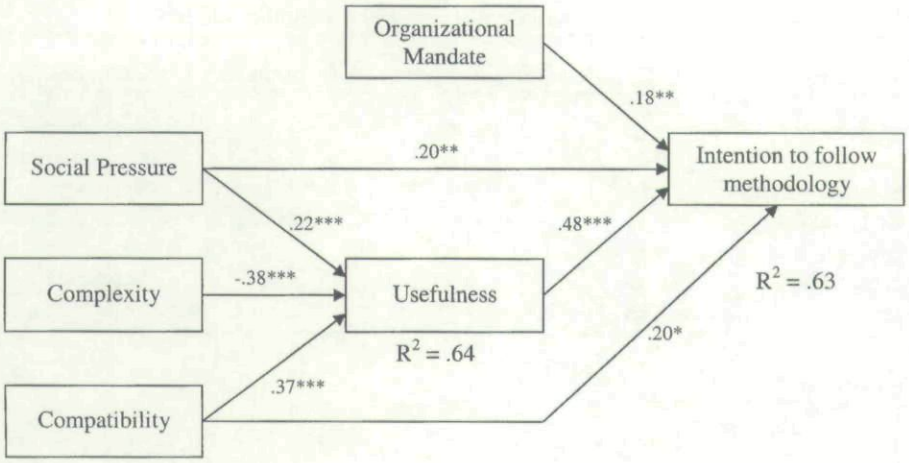


Figure 1. Determinants of Software Developers' Intentions to Follow Methodologies. *Notes:* * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$. Simple path models, such as this figure, are often just-identified (i.e., saturated), which means χ^2 equals zero with no degrees of freedom [43]. Thus, although unique parameter estimates and the model's explanatory power (i.e., R^2) can be produced and are meaningful, the model fit itself cannot be tested [14, 38].

the present context, two significant modification indices indicated that effects on usefulness from social pressure and compatibility would need to be included in the model in order to improve fit. Including these two paths and model parameters yielded a model with excellent fit, as shown in Figure 2. Although interpretation of the indirect effects of complexity, compatibility, and social pressure must be viewed tentatively



Fit measures	χ^2/df	*P*	RMSR	GFI	AGFI	NFI	Evaluation of fit
Thresholds	≤5.0	≥0.05	≤.10	≥.90	≥0.80	≥0.90	
Figure 2	0.683	0.51	0.03	0.99	0.96	0.99	Excellent

Figure 2. Methodology Intention Model—Direct and Indirect Determinants of Software Developers' Intention to Follow Methodologies. *Notes:* * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$.

given the post hoc nature of their assessment, meaningful theoretical interpretations and supportive empirical findings are available in the literature, as discussed in the next section. The modified model has the advantage of allowing us to assess the direct, indirect, and total effects of the variables, potentially yielding more insight about the relative overall influence of these determinants. As shown in Table 3, an examination of total effects suggests that usefulness, compatibility, and social pressure have the greatest total influence on intention with complexity and organizational mandate having the least.

Discussion

THE PRESENT STUDY TOOK A DEDUCTIVE, theory-driven approach utilizing extant theories of intention formation and innovation diffusion from the IT tool domain and adapted them to the context of methodology acceptance. Five potential determinants of intention were examined; four were found to have significant direct effects: usefulness, social pressure, compatibility, and organizational mandate (the fifth construct, complexity, was not significant). Compared to the tool domain from which the key determinants were adapted, the resulting pattern of intention determinants is quite different: (1) usefulness, although significant, was comparatively weaker; (2) complexity was not significant; and (3) social pressure, organizational mandate, and compatibility were all significant—a unique combination in explaining intentions. Overall,

Table 3. Direct, Indirect, and Total Effects of Determinants on Intention

Determinant	Direct effect on intention	Effect on usefulness	Indirect effect on intention	Total effect on intention
Usefulness	0.48	0.00	0.00	0.48
Compatibility	0.20	0.37	0.18	0.38
Social pressure	0.20	0.22	0.11	0.31
Complexity	0.00	–0.38	–0.18	–0.18
Organizational mandate	0.18	0.00	0.00	0.18

the results explained 63 percent of the developer's intention to follow the methodology. For comparison, Khalifa and Verner [37] produced R^2s of 0.18 and 0.32 (in explaining two different methodologies), most TAM studies produce an R^2 of about 0.40 [61] and a recent study using DOI reported an R^2 of 0.46 [2]. Thus, comparatively, the present study does a good job of explaining intention.

Very few studies have explicitly examined acceptance of methodologies by individual developers. Johnson et al. [33] solicited beliefs about a new development process (specifically, OO development), but did not examine links between the beliefs and intentions or behavior. Khalifa and Verner [37] studied usage determinants of two specific approaches to development, prototyping and waterfall, by individual developers. The present study differs from these studies in several ways. First, neither of the prior studies examined the direct determinants of intention. Johnson et al. [33] did not attempt to explain intention formation. Khalifa and Verner [37] looked directly at usage, bypassing intentions, which may be misleading in a mandatory environment. Second, Khalifa and Verner [37] primarily used single-item indicators, hindering the validity and reliability of their data. Third, both prior studies are largely inductive, exploratory studies, using a grounded theory approach.

Given the two different approaches taken by the present study and prior methodology studies (deductive versus inductive, respectively), it is interesting to observe similarities and convergence to a comparable set of determinants. Concepts similar to usefulness, social pressure, compatibility, organizational mandate, and complexity appear in at least one of the prior methodology studies (although the exact relationships between the factors and intentions are not always examined or clearly demonstrated in the prior studies). The convergence from different perspectives is encouraging and suggests progress beyond ideas uncovered in previous exploratory studies toward a better understanding of the factors influencing intentions to follow a methodology.

Theoretical Implications of Intention Determinants

Usefulness was a significant determinant of intentions to follow the methodology in the present study, consistent with prior methodology and tool studies. Johnson et al.'s [33] belief elicitation revealed several usefulness variables: process usefulness, product usefulness, communication usefulness, and career usefulness. Although Johnson

et al. [33] did not test the effects of these beliefs on intention, the exploratory nature of the elicitation process suggests that usefulness is a factor to be considered in methodology adoption by individual adopters. Khalifa and Verner [37] also included concepts similar to usefulness (product quality and process quality) in their study of methodology use. Process quality was significant for both methodologies examined in their study. Usefulness is also typically found in tool adoption studies [2, 15, 45, 57, 58]. However, usefulness generally has a beta of around 0.60 in TAM studies [61]. The influence of usefulness in the present study is comparatively small ($\beta = 0.46$; Figure 1), which signals diminished importance in this environment.

Two reasons may exist for this diminished importance. First, usefulness of a particular methodology may not be quickly realized. Thus, usefulness may not exhibit as strong an effect early in the adoption process as it might after direct experience has enabled developers to witness first-hand the benefits to their own performance. Second, the benefits of the methodology may primarily accrue to the organization rather than the individual. Some developers may not perceive methodologies to be useful, that is, beneficial to their own personal job performance, even if they regard following them as beneficial to their organization. TAM's usefulness construct and DOI's relative advantage construct are defined as one's perception of the extent to which performing a specified behavior will cause improved *individual* job performance. If individuals' perceptions of beneficial impacts of methodology adherence on team, department, or organizational performance are influential in determining their intentions, that fact would have been overlooked in the present study because such concerns are not part of TAM or DOI. The "communication usefulness" perception found salient in Johnson et al.'s [33] elicitation study is indicative of this broader conceptualization of usefulness. Another aspect of usefulness omitted by TAM and DOI pertains to the *skill acquisition* benefits anticipated to result from methodology adherence, which may benefit developers not just by enhancing their near-term job performance, but also by enhancing their market value as an IT professional. Research on the motivation of IT professionals suggests that skill acquisition is a key driver of motivation, satisfaction, and turnover (e.g., [18, 30]). Thus, gains in explanatory power may result from extending the present model to include constructs such as "perceived long-term consequences," which was not found to be significant in a tool usage context [58]. In general, broadening the conceptualization of usefulness to encompass organizational or career considerations represents a promising avenue of further investigation that would build upon and go beyond the current findings.

Social pressure, which has been found to be a significant direct determinant of intention in some tool studies (e.g., [57, 58, 61]; Studies 3 and 4]) and nonsignificant in others (e.g., [15, 45, 61]; Studies 1 and 2), was found significant in the present research. The direct effect on intention is based on the mechanism of compliance [35] which suggests that an individual perceives that an important referent wants him or her to perform the behavior [61]. In a methodology context, the extra-organizational pressure and intra-organizational pressure beliefs in Johnson et al. [33] included items referring to the opinions of important referents, such as coworkers and consultants. Also, due to the teamwork culture of software development, social pressure may be

more important in this environment than an environment of IT tool use. People adopting individually used IT tools may not be subject to the same social pressures as one engaged in a heavy teamwork environment. Thus, social pressure may be more important in the acceptance of methodologies, compared to IT tools.

Organizational mandate had a significant direct effect on intentions. In some tool studies, such as Venkatesh and Davis [61] and Hartwick and Barki [26], organizational mandate was found to moderate the effect of social pressure on intention, but not to act as a direct determinant. A supplemental analysis of the data in this study did not confirm this moderating relationship (β of OM*SP = –0.02; $p = 0.67$), indicating that social pressures from one's professional colleagues and the external pressure from the organizational mandate are not interacting as suggested in IT tool studies. Rather, each exerts significant but separate pressures on intentions. Whereas the moderating effect of voluntariness on the relationship between social influence and intention in TAM2 was inspired by Kelman's [35] distinction among the mechanisms of identification, internalization, and compliance (see Venkatesh and Davis [61]), the present findings are more aligned with the perspective of French and Raven [21], who theoretically distinguish between legitimate power (in which role relationships dictate a legitimate right to prescribe behavior, as is the case with organizational mandates) and referent power (in which the target of influence identifies strongly with the influencer, as is the case with social influence) as phenomenologically distinct bases for influence. Further insight into the nature and roles of the two distinct methodology adherence determinants of compliance with social pressure, on one hand, and acquiescence to organizational mandate, on the other hand, awaits further investigation.

Compatibility is a significant determinant of methodology use intentions in the present study. This finding is consistent with the preliminary work of Johnson et al. [33], whose study suggests that developers' beliefs about the compatibility between their existing skills and background and those needed in a new development environment may be an important consideration in intention formation. The exact role of compatibility in tool adoption studies is unclear—it was found nonsignificant as a direct determinant in some prior research [2] and significant in other studies (e.g., [13, 27, 31]). Because methodologies may change the way a developer works, compatibility may be a more important factor in this environment than in IT tool adoption, which may not require a change in the way someone performs their job. So far, compatibility has been an elusive construct in tool adoption research [51]. This finding perhaps sheds some light on this confusion. When the innovation radically changes the work practices of the adopter (in the case of a methodology), compatibility may play an important role. For less disruptive innovations (such as a word processor), compatibility may be unimportant as a direct determinant of intention. In the present study, the changes to one's work processes caused by methodologies are apparently large enough to warrant a direct influence on intention.

Complexity, identified as a salient belief in Johnson et al. [33] and often found significant in previous IT tool acceptance studies (e.g., [15, 45, 58, 61]), was found nonsignificant as a direct intention determinant in the present study. TAM theorizes two distinct mechanisms by which perceived ease of use (complexity) influences

intentions in the context of tool adoption: instrumentality and self-efficacy [15]. Instrumentality operates as an indirect effect (as discussed in the next section); the efficacy effect is a direct effect of complexity on intentions to carry out an act, and reflects individuals' confidence in their ability to successfully complete the act if attempted. If people doubt their ability to perform a difficult behavior, they may alter their intentions accordingly to avoid even intending to try. A closer look at the data suggests that most respondents did not regard methodology adherence behavior as being so complex as to cast doubt on their ability to follow through on the act if attempted (mean = 3.5 on a 1–7 scale; standard deviation = 1.35). To corroborate this interpretation, we divided the sample into two groups: those individuals for whom complexity was so high as to threaten the success of methodology adherence behavior given an attempt, and those individuals for whom complexity was not so high. Specifically, we compared the intentions of developers who rated complexity as 6 or 7 on the seven-point complexity scale ($n = 18$) to those who rated complexity less than 6 ($n = 100$). Mean of intentions for the high-complexity developers (3.25) was significantly lower than that of the mid- to low-complexity developers (5.60), $t = 6.56$, $p < 0.001$. This is consistent with the interpretation that an efficacy mechanism was at work for a relative minority of developers in the current sample. The generalizability of the lack of a direct complexity-intention effect to other methodologies and developer populations is a question for future research.

Consideration of Potential Indirect Effects

With the exception of complexity, all hypothesized direct effects on intention were significant, as discussed in the previous section. Because the direct effect of complexity was not significant and an indirect effect is known to exist in a tool context, the analysis was extended to investigate the indirect effect of complexity via usefulness. This analysis led to an examination of additional indirect effects (compatibility and social pressure via usefulness) as suggested by the modification indices of SEM (as explained in the Results section). Although tentative because of their post hoc discovery, the new relationships have plausible theoretical interpretations and provide additional insight into the role of these factors on intention formation.

Although complexity did not exert a direct (efficacy) effect on intentions, the results suggest an indirect (instrumentality) effect: complexity of the behavior influences usefulness to account for the perceived effect on ones' overall job performance of exerting the greater effort needed to perform more complex behaviors. Previous tool-oriented studies using TAM have found evidence of an indirect (via usefulness) effect of complexity (ease of use) on intentions (e.g., [45, 57, 61]). The indirect effect of complexity on intention via usefulness suggests that the more complex a methodology is perceived to be, the less useful it is considered, and therefore usefulness functions as a mediational mechanism linking complexity to intention.[2]

In addition to its direct effect on intentions, social pressure appears to have an indirect effect via usefulness. The influence of social pressure on usefulness is attributed to internalization, which is the process "in which people incorporate social influ-

ences into their own usefulness perceptions" [61, p. 198]. In the early stages of deployment, as in this study, developers may be more influenced by social pressures since they would have had little direct experience with the methodology [57], thus relying on the opinions of others to develop their perception of usefulness. Overall, social pressure appears to be an important determinant of acceptance of methodologies, exhibiting a total effect on intention of 0.31 (see Table 3).

Like social pressure, compatibility was found to indirectly influence intention through usefulness above and beyond its direct effect on intentions. Consistent with Chau and Hu [12], the present study indicates that compatibility may serve as an antecedent to usefulness, suggesting that inconsistencies between a new tool and existing work practices can influence one's perception of usefulness (i.e., a potential adopter may not find the tool useful if it does not support his or her work processes). Perhaps the more incremental change entailed with tool adoption in contrast to the more radical change involved with methodology adoption may account for the failure of Moore and Benbasat [46] to establish discriminant validity between compatibility and usefulness in a tool context, with items from both constructs loading on a single unidimensional factor. Overall, compatibility demonstrates a strong total effect on intentions (0.38), second only to usefulness (see Table 3).

In comparing the present study with prior research on tool use intentions (from which the constructs were adapted), the findings (summarized in Table 4) suggest that as the behavioral domain changes from tool use intention to methodology use intention, there is a reduction in the relevance of how easy or hard the behavior is to perform, and there is an increase in the relevance of social pressure to perform the behavior, the compatibility of the target behavior with individuals' current ways of performing their work, and the perception of a formal mandate. Interestingly, when the indirect effects are considered, organizational mandate has the weakest total effect, further reinforcing the implication that an organizational mandate, in this environment, is not sufficient to change one's behavior. Conversely, compatibility has a very large total effect, thus reinforcing the importance of compatibility in this environment compared to tool environments. Overall, whereas the constructs as a group were adapted from the domain of tool usage intentions, there was a different pattern of which determinants were significant. This is consistent with the observation that, compared to tool adoption, methodology adoption is a more fundamental and radical change to the behavioral processes in conducting one's work, and is more apt to be a mandatory than voluntary change.

Managerial Implications

This research provides important new insights for managing the introduction and deployment of a methodology within an organization. A common assumption when introducing a new technology is that issuing an official mandate will, by itself, be sufficient impetus for full adoption. Consistent with Kotter [39], who suggests that organizational mandates alone are of limited effectiveness in general, the findings from this study also indicate that individual developer acceptance is far from assured even in

Table 4. Summary and Comparison of Findings with Prior Methodology and Tool Studies

Construct	Prior methodology studies: Johnson et al. [33]	Prior methodology studies: Khalifa and Verner [37]	Typical relationship in tool studies	This study
Usefulness	Identified several usefulness beliefs, but did not examine relationship with intentions.	Concepts similar to usefulness (process and product quality) examined as determinants; process quality was significant.	Very strong direct determinant of intention (β of around 0.60 in many TAM studies).	Strong direct determinant of intention ($\beta = 0.46$), but weaker than most other tool studies.
Complexity	Contained elements of complexity beliefs (difficulty with learning OO development; perceived complexity of OO development); relationship with intentions not examined.	None	Direct determinant of intention; also operates through usefulness to influence intention.	Nonsignificant as direct determinant; significantly related to usefulness and indirectly influences intention. Thus, appears to be operating through usefulness sooner than most other tool studies; comparatively weaker importance in this environment (as direct determinant).

Social pressure	Contained elements of social pressure in the extra-organizational and intra-organizational belief constructs; relationship with intentions not examined.	Team size significantly related to usage.	Previous findings are mixed; interacts with organizational mandate (voluntariness) in TAM2.	Direct determinant of intention; serves as antecedent to usefulness (thus, also demonstrating an indirect effect on intention); does not interact with organizational mandate. Thus, increased importance in this environment compared to tools.
Compatibility	Beliefs about compatibility of existing skills and background with new domain identified; relationship with intentions not examined.	None	Previous findings are mixed.	Direct determinant of intention; serves as antecedent to usefulness (thus, also demonstrating an indirect effect on intention). Increased importance in this environment.
Organizational mandate	Contained reference to the pressure of supervisors/managers; relationship with intentions not examined.	None	Previous findings are mixed; interacts with social pressure (subjective norm) in TAM2.	Direct determinant of intention; does not interact with social pressure. Thus, increased importance in this environment.

the presence of an organizational mandate. Other important drivers of methodology acceptance exist beyond the target user's perception that the adoption decision is mandatory. Specifically, although organizational mandate had a significant effect on intentions, the strength of its direct influence was the lowest among the four significant constructs, and usefulness, compatibility, and social pressure all influenced intentions directly, above and beyond the effects of organizational mandate. In this particular research setting, as in many methodology deployments in practice, the organizational mandate was not accompanied by expressed rewards or punishments. This may have weakened the influence of organizational mandate relative to what it might have been if methodology adherence, per se, had been directly rewarded, or if severe penalties such as loss of employment could have resulted from non-adherence.

If the developers do not see the methodology as useful, its prospects for successful implementation will be seriously threatened. A methodology that does not enable developers to be more productive and enhance their overall performance likely will not be used in a sustained manner, even if organizationally mandated. Therefore, demonstrating the individual productivity benefits (i.e., usefulness) of the methodology may help shape a developer's intention to adopt. Unfortunately, in the early stages of adoption, immediate effectiveness gains due to a methodology for a given organization or developer may not be possible. In fact, individual programmers may actually lose productivity in the short term as they learn and routinize new development practices [42]. Even if long-term benefits accrue to both the organization and the individual developer, such adverse effects on developer productivity in the short term would tend to erode adoption intentions. This problem may be attenuated, for example, by altering the nature of the methodology to provide greater individual performance improvement, or altering the performance measurement system to reward individual developers directly for organizational-level performance improvements. Also, to persuade developers of the eventual benefits of methodologies, industry metrics from institutions such as the Software Engineering Institute may be valuable.

The more compatible a mandated new methodology is with how developers currently perform their work, the more likely they are to form intentions to follow it, all else being equal. Since a methodology is a comprehensive specification of the steps and procedures to be used throughout all phases of the software development process [29], the more it departs from the current work processes of any given developer, the longer and harder developers must strive to unlearn old routines and learn new ones. Software developers usually evolve work practices they find effective and have come to rely on to produce effective work output. There is significant effort and uncertainty involved in learning and routinizing new work patterns. Managers can do several things to overcome this barrier to successful deployment. Management would be wise to explicitly demonstrate instances of the new methodology's compatibility with existing work practices. If there are particular parts of a new methodology that especially depart from current practice, but which are not vital to the effectiveness of the methodology, these may be modified to be brought into greater alignment with current practice. Rather than attempting to convert from current practice to the new methodology in a single step (the "big bang" approach), it may be beneficial to seek

ways to design a migration path in which developers adopt parts of the methodology incrementally, in each step experiencing improved productivity. Training can be designed to undo existing work habits and routines while propelling developers along such a migration path. In any case, when selecting a methodology, managers should take into consideration how compatible the methodology is to preexisting work practices on the developer community. Overall, this finding suggests that off-the-shelf methodologies may be less appealing than those that are at least partially custom-tailored to provide greater compatibility with existing processes.

Social pressure exerts a distinct influence on intentions to adopt a methodology above and beyond usefulness, organizational mandate, and compatibility. This tells us that even if developers regard a methodology as useful, compatible with their current work practices, and organizationally mandated, they may still avoid using it because important referents (e.g., coworkers) think they should not be using it. For example, developers who believe that a mandated methodology would be useful and compatible for them personally may work among professional colleagues who actively oppose its use. In order to sustain a balance of equitable social exchange, or to avoid damaging key work relationships, such developers may avoid using the methodology despite its merits. Management may be able to devise persuasive communication strategies designed to reduce the developers' perceived likelihood and strength of professional colleagues' counterimplementation desires, or to attenuate their motivation to comply with such social pressure.

How should practitioners interpret the current findings for the purpose of prioritizing these various practical interventions for increasing developers' intentions to follow a new methodology? The coefficients in Tables 2 and 3 and Figures 1 and 2 indicate the predicted increase in intention resulting from a one unit increase of the associated determinant.[3] An examination of both direct and indirect effects (through usefulness) affords a meaningful practical interpretation of the actionable leverage points for increasing intentions. For example, an intervention that produces a one unit increase in compatibility would have a combined direct effect of increasing intentions by 0.20 units plus the indirect effect of increasing usefulness by 0.37 units, which in turn influences intentions by 0.48 units over and above compatibility's direct effect. The total effect of 0.38 (0.20 + (0.37 * 0.48)) is a more meaningful interpretation of the leveraging effect of increasing compatibility than would be obtained by focusing only on the direct effect of 0.20 (Table 3). Similarly, the total effect of social pressure on intention (0.31) is greater than what is suggested by its direct effect (0.20). In terms of total effects, usefulness is the most influential driver of intentions (0.48), followed by compatibility (0.38) and social pressure (0.31). Complexity and organizational mandate are tied for the least influential determinants (–0.18 and 0.18, respectively). From a managerial perspective, this tells us that while complexity and organizational mandate are effective levers, it would be a mistake to focus on interventions that target these two determinants to the exclusion of usefulness, compatibility, and social pressure, which could more than offset the beneficial effects due to reduced complexity and organizational mandate. Ideal interventions would target all five of these known intention determinants. When trade-offs are necessary in order to

design cost-effective interventions, the total effects depicted in Table 3 can be a useful guide to prioritization.

Future Research

The present research shows that mandating methodology use by software developers does not guarantee that the methodology actually will be followed. By demonstrating the significant roles of usefulness, social pressure, and compatibility above and beyond organizational mandate, several worthwhile avenues of future research are apparent. There is a need for research to confirm, extend, and refine the findings presented here. In particular, any additional determinants of methodology acceptance should be identified and incorporated into the research. Perhaps individual characteristics such as feelings of autonomy or power, job characteristics such as role ambiguity and work overload, or organizational characteristics such as organizational commitment and organizational goals influence a developer's intention.

In the research site for the present study, developers' job descriptions and reward structures were not modified when introducing the new methodology: What would be the effect of changing these? What are the causal antecedents of social pressure, compatibility, and organizational mandate? What is the comparative influence of alternative forms of training and persuasive communication on these determinants of adoption intentions? In the present research site, individual developers are rewarded based on individual performance. What, if any, modifications to the study would be needed to deal with the introduction of cross-functional self-managed development teams comprised of developers and users? Also, why did organizational mandate and social pressure act independently when other studies (e.g., [61]) found a moderating relationship? Is it because of the team environment in software development? If so, will this relationship extend to other team environments?

The TAM and DOI measures used in this study were previously developed and psychometrically validated specifically for the context of tool or artifact adoption, and therefore are worded to refer to the specific behavior of using a system. Because a methodology is not an artifact, but rather an approach to systems development, respondents may find it less natural to refer to its adoption as "using" a methodology. It may seem more natural to refer to the behavioral criterion as *following* or *adhering* to the prescribed steps and procedures of the methodology. Potential semantic distance between the behavioral referents of "using" versus "following" or "adhering to" a methodology may introduce systematic measurement error that could bias parameter estimates in ways that cannot be corrected by the use of structural equation models [16]. Future psychometric work comparing alternative item wording may prove valuable in tailoring the measurement items to the methodology context.[4] In addition, there may be perceptions that are salient in a methodology context that are not salient in TAM and DOI, which were developed in a tool context. For example, the belief elicitation study of Johnson et al. [33] suggests that process usefulness, communication usefulness, and career usefulness are salient for system developers contemplating adoption of a methodology, yet those beliefs are not present in TAM or

DOI. Whereas the present research found that TAM and DOI constructs were effective in explaining methodology adoption, a key next step in extending the boundary conditions of tool-oriented theories is to incorporate such additional determinants, if salient.

Conclusion

This study investigated the determinants of software developers' intentions to follow a methodology. Drawing from theories of intention formation and innovation diffusion, five potential direct determinants were identified and empirically tested. Four of the five determinants were found significant (usefulness, compatibility, social pressure, organizational mandate). Perceived complexity was not significant. The resulting pattern of intention determinants differs from that typically observed in studies of IT tool acceptance, reflecting the unique nature of process change and challenges faced when attempting to deploy a methodology. By identifying the determinants of intention, prescriptive models can be enhanced and contingency models of methodology use may be extended from a focus on effectiveness of a methodology, if used, to encompass the consideration of whether or not a methodology will be used in the first place. From a practical perspective, insights into intentions provided by the study suggest better strategies for introducing methodologies. By knowing the determinants of a developer's intention, management can take appropriate action to increase the likelihood of successfully deploying a new methodology.

Acknowledgments: The authors are grateful to the Information Technology Research Center at the University of Arkansas for financial support of this research and to Melinda Korzaan for her assistance with the analysis using AMOS.

Notes

1. The software package, AMOS 4.01, was used for SEM analysis.
2. Without usefulness in the model, complexity exerts a significant direct effect on intentions ($\beta = 0.216$; $p = 0.006$). With the complexity $\rightarrow$usefulness and usefulness $\rightarrow$intentions paths in the model (both significant), the direct effect of complexity on intentions falls to $0.041 (p = 0.578)$. Thus, a mediator effect is demonstrated.
3. A standardized beta coefficient such as reported in Tables 2 and 3 is a model's estimate of the number of standard deviations of increase in the dependent variable resulting from a single standard deviation increase in the associated independent variable, all else remaining constant.
4. We would like to thank an anonymous reviewer for this suggestion.

References

1. Agarwal, R. Individual acceptance of information technologies. In R. Zmud (ed.), *Framing the Domains of IT Management: Projecting the Future Through the Past.* Cincinnati: Pinnaflex Educational Resources, 2000, pp. 85–104.
2. Agarwal, R., and Prasad, J. The role of innovation characteristics and perceived voluntariness in the acceptance of information technologies. *Decision Sciences, 28,* 3 (1997), 557–582.

3. Agarwal, R., and Prasad, J. A field study of the adoption of software process innovations by information systems professionals. *IEEE Transactions on Engineering Management, 47,* 3 (2000), 295–308.

4. Ajzen, I. *Attitudes, Personality, and Behavior.* Chicago: Dorsey Press, 1988.

5. Ajzen, I., and Fishbein, M. *Understanding Attitudes and Predicting Social Behavior.* Englewood Cliffs, NJ: Prentice Hall, 1980.

6. Anderson, J.C., and Gerbing, D.W. Structural equation modeling in practice: A review and recommended two-step approach. *Psychological Bulletin, 103,* 3 (1988), 411–423.

7. Bagozzi, R.P., and Phillips, L.W. Representing and testing organizational theories: A holistic construal. *Administrative Science Quarterly, 27,* 3 (1982), 459–489.

8. Barki, H., and Hartwick, J. Interpersonal conflict and its management in information system development. *MIS Quarterly, 25,* 2 (2001), 195–228.

9. Bayer, J., and Melone, N. A critique of diffusion theory as a managerial framework for understanding adoption of software engineering innovations. *Journal of Systems and Software, 9,* 2 (1989), 161–166.

10. Brancheau, J.C.; Janz, B.D.; and Wetherbe, J.C. Key issues in information systems management: 1994–95 SIM Delphi results. *MIS Quarterly, 20,* 2 (1996), 225–242.

11. Chau, P.Y.K. An empirical investigation on factors affecting the acceptance of CASE by systems developers. *Information & Management, 30,* 6 (1996), 269–280.

12. Chau, P.Y.K., and Hu, P.J. Information technology acceptance by individual professionals: A model comparison approach. *Decision Sciences, 32,* 4 (2001), 699–719.

13. Chin, W.W., and Gopal, A. Adoption intention in GSS: Relative importance of beliefs. *DATA BASE for Advances in Information Systems, 26,* 2–3 (1995), 42–64.

14. Chou, C., and Bentler, P.M. Estimates and tests in structural equation modeling. In R.H. Hoyle (ed.), *Structural Equation Modeling: Concepts, Issues, and Applications.* Thousand Oaks, CA: Sage, 1995, pp. 37–55.

15. Davis, F.D.; Bagozzi, R.; and Warshaw, P. User acceptance of computer technology: A comparison of two theoretical models. *Management Science, 35,* 8 (1989), 982–1003.

16. DeShon, R.P. A cautionary note on measurement error corrections in structural equation models. *Psychological Methods, 3,* 4 (1998), 412–423.

17. Dietrich, G.B.; Walz, D.B.; and Wynekoop, J.L. The failure of SDT diffusion: A case for mass customization. *IEEE Transactions on Engineering Management, 44,* 4 (1997), 390–398.

18. Ferratt, T.W., and Short, L.E. Are information systems people different? An investigation of how they are and should be managed. *MIS Quarterly, 12,* 3 (1988), 427–443.

19. Fichman, R. Information technology diffusion: A review of empirical research. In J.I. DeGross, J.D. Becker, and J.J. Elam (eds.), *Proceedings of the Thirteenth International Conference on Information Systems.* Baltimore: ACM, 1992, pp. 195–206.

20. Fornell, C., and Larcker, D. Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research, 18,* 1 (1981), 39–50.

21. French, J.R.P., and Raven, B. The bases of social power. In D. Cartwright (ed.), *Studies in Social Power.* Ann Arbor, MI: Institute for Social Research, 1959, pp. 150–167.

22. Glass, R.L. The realities of software technology payoffs. *Communications of the ACM, 42,* 2 (1999), 74–79.

23. Glass, R.L. A snapshot of systems development practice. *IEEE Software, 16,* 3 (1999), 110–111.

24. Hardgrave, B.C. When to prototype: Decision variables used in industry. *Information & Software Technology, 37,* 2 (1995), 113–118.

25. Harter, D.E.; Krishnan, M.S.; and Slaughter, S.A. Effects of process maturity on quality, cycle time, and effort in software product development. *Management Science, 46,* 4 (2000), 451–466.

26. Hartwick, J., and Barki, H. Explaining the role of user participation in information system use. *Management Science, 40,* 4 (1994), 440–465.

27. Hebert, M., and Benbasat, I. Adopting information technology in hospitals: The relationships between attitudes/expectations and behavior. *Hospital and Health Services Administration, 39,* 3 (1994), 369–383.

28. Herbsleb, J.; Carleton, A.; Rozum, J.; Siegel, J.; and Zubrow, D. Benefits of CMM-Based Software Process Improvement: Initial Results (CMU/SEI-94-TR-013 and ESC-TR-

94-013). Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1994 (available at www.sei.cmu.edu/publications/documents/94.reports/94.tr.013.html).

29. Hoffer, J.A.; George, J.F.; and Valacich, J.S. *Modern Systems Analysis and Design,* 2d ed. Reading, MA: Addison-Wesley, 1999.

30. Igbaria, M.; Parasuraman, S.; and Badawy, M.K. Work experiences, job involvement, and quality of work life among information systems personnel. *MIS Quarterly, 18,* 2 (1994), 175–197.

31. Iivari, J. Why are CASE tools not used? *Communications of the ACM, 39,* 10 (1996), 94–103.

32. Iivari, J.; Hirschheim, R.; and Klein, H.K. A dynamic framework for classifying information systems development methodologies and approaches. *Journal of Management Information Systems, 17,* 3 (Winter 2000–2001), 179–218.

33. Johnson, R.; Hardgrave, B.C.; and Doke, E.R. An industry analysis of developer beliefs about object-oriented systems development. *DATA BASE for Advances in Information Systems, 30,* 1 (1999), 47–64.

34. Keil, M. Pulling the plug: Software project management and the problem of project escalation. *MIS Quarterly, 19,* 4 (1995), 421–443.

35. Kelman, H.C. Compliance, identification, and internalization: Three processes of attitude change. *Journal of Conflict Resolution, 2,* 1 (1958), 51–60.

36. Kemerer, C.F. How the learning curve affects case tool adoption. *IEEE Software, 9,* 3 (1992), 23–28.

37. Khalifa, M., and Verner, J. Drivers for software development method usage. *IEEE Transactions on Engineering Management, 47,* 30 (2000), 360–369.

38. Kline, R.B. *Principles and Practice of Structural Equation Modeling.* New York: Guilford Press, 1998.

39. Kotter, J.P. *Leading Change.* Boston: Harvard Business School Press, 1996.

40. Kozar, K.A. Adopting systems development methods: An exploratory study. *Journal of Management Information Systems, 5,* 4 (Spring 1989), 73–86.

41. Krishnan, M.S.; Mukhopadhyay, T.; and Zubrow, D. Software process models and project performance. *Information Systems Frontiers, 1,* 3 (1999), 267–277.

42. Leonard-Barton, D. Implementing structured software methodologies: A case of innovation in process technology. *Interfaces, 17,* 3 (1987), 6–17.

43. MacCallum, R.C. Model specification: Procedures, strategies, and related issues. In R.H. Hoyle (ed.), *Structural Equation Modeling: Concepts, Issues, and Applications.* Thousand Oaks, CA: Sage, 1995, pp. 16–36.

44. MacCallum, R.C.; Roznowski, M.; and Necowitz, L.B. Model modifications in covariance structure analysis: The problem of capitalization on chance. *Psychological Bulletin, 111,* 3 (1992), 490–504.

45. Mathieson, K. Predicting user intentions: Comparing the technology acceptance model with the theory of planned behavior. *Information Systems Research, 2,* 3 (1991), 173–191.

46. Moore, G., and Benbasat, I. Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information Systems Research, 2,* 3 (1991), 192–222.

47. Orlikowski, W.J. CASE tools as organizational change: Investigating incremental and radical changes in systems development. *MIS Quarterly, 17,* 3 (1993), 309–340.

48. Pfleeger, S.L. Understanding and improving technology transfer in software engineering. *Journal of Systems and Software, 47,* 2–3 (1999), 111–124.

49. Premkumar, G., and Potter, M. Adoption of computer aided software engineering (CASE) technology: An innovation adoption perspective. *DATA BASE for Advances in Information Systems, 26,* 2–3 (1995), 105–123.

50. Raghaven, S.A., and Chand, D.R. Diffusing software-engineering methods. *IEEE Software, 6,* 4 (1989), 81–90.

51. Ramiller, N.C. Perceived compatibility of information technology innovations among secondary adopters: Toward a reassessment. *Journal of Engineering and Technology Management, 11,* 1 (1994), 1–23.

52. Ravichandran, T., and Rai, A. Total quality management in information systems development. *Journal of Management Information Systems, 16,* 3 (Winter 1999–2000), 119–155.

53. Roberts, S., and Pashler, H. How persuasive is good fit? A comment on theory testing. *Psychological Review, 107,* 2 (2000), 358–367.

54. Roberts, T.L.; Gibson, M.L.; Fields, K.T.; and Rainer, R.K. Factors that impact implementing a system development methodology. *IEEE Transactions on Software Engineering, 24,* 8 (1998), 640–649.

55. Rogers, E. *Diffusion of Innovations,* 4th ed. New York: Free Press, 1995.

56. Sultan, F., and Chan, L. The adoption of new technology: The case of object-oriented computing in software companies. *IEEE Transactions on Engineering Management, 47,* 1 (2000), 106–126.

57. Taylor, S., and Todd, P. Understanding information technology usage: A test of competing models. *Information Systems Research, 6,* 2 (1995), 144–179.

58. Thompson, R.; Higgins, C.; and Howell, J. Personal computing: Toward a conceptual model of utilization. *MIS Quarterly, 15,* 1 (1991), 125–143.

59. Tornatzky, L.G., and Klein, K.J. Innovation characteristics and innovation adoption-implementation: A meta-analysis of findings. *IEEE Transactions on Engineering Management, EM-29,* 1 (1982), 28–45.

60. Venkatesh, V. Creation of favorable user perceptions: Exploring the role of intrinsic motivation. *MIS Quarterly, 23,* 2 (1999), 239–260.

61. Venkatesh, V., and Davis, F.D. A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science, 46,* 2 (2000), 186–204.

62. Vessey, I., and Glass, R.L. Strong vs. weak approaches to systems development. *Communications of the ACM, 41,* 4 (1998), 99–102.

63. Vessey, I.; Jarvenpaa, S.L.; and Tractinsky, N. Evaluation of vendor products: CASE tools as methodology companions. *Communications of the ACM, 35,* 4 (1992), 90–105.

64. Whiting, R. Development in disarray. *Software Magazine, 18,* 12 (1998), 20.

65. Yoo, Y., and Alavi, M. Media and group cohesion: Relative influences on social presence, task participation, and group consensus. *MIS Quarterly, 25,* 3 (2001), 371–390.

Appendix A. Measurement Scales

Behavioral Intention

1. I intend to use ADM* in the future for my work.
2. Given the opportunity, I would use ADM.

Usefulness

1. Using ADM improves my job performance.
2. Using ADM increases my productivity.
3. Using ADM enhances the quality of my work.
4. Using ADM makes it easier to do my job.
5. The advantages of using ADM outweigh the disadvantages.
6. ADM is useful in my job.

Complexity

1. Learning ADM was easy for me.
2. I think ADM is clear and understandable.
3. Using ADM does not require a lot of mental effort.
4. I find ADM easy to use.
5. ADM is not cumbersome to use.
6. Using ADM does not take too much time from my normal duties.

Social Pressure

1. People who influence my behavior think I should use ADM.
2. People who are important to me think I should use ADM.
3. Coworkers think I should use ADM.

Organizational mandate

1. Although it may be helpful, using ADM is certainly not compulsory in my job.
2. My supervisor does not require me to use ADM.
3. My use of ADM is voluntary.

Compatibility

1. ADM is compatible with the way I develop systems.
2. Using ADM is compatible with all aspects of my work.
3. Using ADM fits well with the way I work.

* ADM is an acronym for Application Development Methodology, a pseudonym for the true methodology name, which is disguised in order to mask the identity of the company where the data was collected.

Appendix B. Convergent Validity

Items	Loadings	Standard error	*t*-value	*P*
Usefulness-1	0.894	0.127	12.500	<0.001
Usefulness-2	0.787	0.139	10.232	<0.001
Usefulness-3	0.849	0.121	11.489	<0.001
Usefulness-4	0.804	0.139	10.563	<0.001
Usefulness-5	0.957	0.125	14.122	<0.001
Usefulness-6	0.944	0.119	13.768	<0.001
Complexity-1	0.796	0.128	10.304	<0.001
Complexity-2	0.899	0.115	12.519	<0.001
Complexity-3	0.839	0.115	11.179	<0.001
Complexity-4	0.908	0.108	12.711	<0.001
Complexity-5	0.786	0.115	10.121	<0.001
Complexity-6	0.715	0.134	8.851	<0.001
Social pressure-1	0.932	0.114	13.166	<0.001
Social pressure-2	0.973	0.105	14.196	<0.001
Social pressure-3	0.675	0.133	8.240	<0.001
Compatibility-1	0.843	0.108	11.356	<0.001
Compatibility-2	0.908	0.102	12.803	<0.001
Compatibility-3	0.990	0.104	14.953	<0.001
Organizational mandate-1	0.594	0.135	6.552	<0.001
Organizational mandate-2	0.893	0.126	10.484	<0.001
Organizational mandate-3	0.745	0.148	8.487	<0.001
Intent-1	0.767	0.123	9.515	<0.001
Intent-2	0.877	0.161	11.409	<0.001

Notes: To demonstrate convergent validity, the *t*-value of each loading should be significant (> 1.96 for 0.05 significance level) and should be twice the standard error of the loading [6]. In this case, *t*-values are all well above 1.96 and all standard errors are relatively low. All factor loadings are also sufficiently large (> 0.50) to demonstrate convergent validity [8, 65].

Appendix C. Discriminant Validity

Construct pair	ΔChi-square	ΔDegrees of freedom	Significance level
BI – USEF	15.96	1	$p < 0.0001$
BI – COMPLX	59.89	1	$p < 0.0001$
BI – SP	40.15	1	$p < 0.0001$
BI – COMP	40.35	1	$p < 0.0001$
BI – OM	75.65	1	$p < 0.0001$
USEF – COMPLX	279.34	1	$p < 0.0001$
USEF – SP	197.86	1	$p < 0.0001$
USEF – COMP	196.94	1	$p < 0.0001$
USEF – OM	109.84	1	$p < 0.0001$
COMPLX – SP	247.93	1	$p < 0.0001$
COMPLX – COMP	248.94	1	$p < 0.0001$
COMPLX – OM	103.04	1	$p < 0.0001$
SP – COMP	209.01	1	$p < 0.0001$
SP – OM	102.53	1	$p < 0.0001$
COMP – OM	108.51	1	$p < 0.0001$

Notes: For the chi-square test, an unconstrained measurement model estimating the correlation between a pair of constructs is compared to a constrained model with the correlation between that pair of constructs fixed to unity. A significant chi-square difference implies that the correlation between the pair of constructs (the unconstrained model) is significantly less than 1.0; thus, demonstrating discriminant validity [7].

Appendix D. Internal Consistency (Composite Reliabilities), Intercorrelations, Means, and Standard Deviations

	BI	USEF	CMPLX	SP	OM	COMP
Behavioral intention (BI)	1.00					
Usefulness (USEF)	0.88**	1.00				
Complexity (CMPLX)	–0.56**	–0.66**	1.00			
Social pressure (SP)	0.63**	0.60**	–0.32**	1.00		
Organizational mandate (OM)	–0.21**	0.02	–0.23*	0.22*	1.00	
Compatibility (COMP)	0.73**	0.77**	–0.62**	0.53**	–0.11	1.00
Mean	5.20	4.40	3.50	4.74	4.44	4.15
Standard deviation	1.62	1.58	1.35	1.45	1.28	1.42
Composite reliability	0.81	0.95	0.93	0.90	0.79	0.94

Notes: * $p < 0.05$, ** $p < 0.01$; all constructs measured on a seven-point scale; correlations were calculated using all factor items rather than averaged factor scores; threshold for composite reliability: ≥ 0.70 [20].