Toward Preprototype User Acceptance Testing of New Information Systems: Implications for Software Project Management

Fred D. Davis and Viswanath Venkatesh

***Abstract*—Errors in requirements specifications have been identified as a major contributor to costly software project failures. It would be highly beneficial if information systems developers could verify requirements by predicting workplace acceptance of a new system based on user evaluations of its specifications measured during the earliest stages of the development project, ideally before building a working prototype. However, conventional wisdom among system developers asserts that prospective users must have direct hands-on experience with at least a working prototype of a new system before they can provide assessments that accurately reflect future usage behavior after workplace implementation. The present research demonstrates that this assumption is only partially true. Specifically, it is true that stable and predictive assessments of a system's perceived *ease of use* should be based on direct behavioral experience using the system. However, stable and behaviorally predictive measures of perceived *usefulness* can be captured from target users who have received information about a system's functionality, but have not had direct hands-on usage experience. This distinction is key because, compared to ease of use, usefulness is generally much more strongly linked to future usage intentions and behaviors in the workplace. Two longitudinal field experiments show that preprototype usefulness measures can closely approximate hands-on based usefulness measures, and are significantly predictive of usage intentions and behavior up to six months after workplace implementation. The present findings open the door toward research on how user acceptance testing may be done much earlier in the system development process than has traditionally been the case. Such preprototype user acceptance tests have greater informational value than their postprototype counterparts because they are captured when only a relatively small proportion of project costs have been incurred and there is greater flexibility to modify a new system's design attributes. Implications are discussed for future research to confirm the robustness of the present findings and to better understand the practical potential and limitations of preprototype user acceptance testing.**

***Index Terms*—Early user acceptance testing, IT project failure, new technology implementation, technology acceptance model (TAM).**

Manuscript received April 30, 2001; revised October 16, 2002. Review of this manuscript was arranged by Department Editor V. Sambamurthy.

F. D. Davis is with the Sam M. Walton College of Business, University of Arkansas, Fayetteville, AR 72701 USA.

V. Venkatesh is with the Robert H. Smith School of Business, University of Maryland, College Park, MD 20742 USA (e-mail: vvenkate@rhsmith.umd.edu).

Digital Object Identifier 10.1109/TEM.2003.822468

I. Introduction

THE FAILURE rate for newly developed information systems remains unacceptably high, especially for large and complex systems. In a sample of 6700 projects across 500 enterprises tracked by Software Productivity Research, Inc. (SPR), 24% of the projects were cancelled and 17% of completed projects experienced cost overruns. This failure rate was amplified by project complexity: projects of 100 000 or more function points exhibited a cancellation rate of 65% and cost overruns of 35% [39]. Clearly, the failure rate for large and complex systems remains a significant concern. Although reasons for this high failure rate are still not well understood, recent evidence suggests that a major culprit is the improper management of user requirements. A study of 8000 projects reported by the Standish Group [58] found that the top three reasons projects were late, over budget, or failed to deliver desired functionality all had to do with requirements management practices: lack of user input, incomplete requirements, and changing requirements. According to McConnell (1996), "feature creep is the most common source of cost and schedule overruns," citing a Software Engineering Institute (SEI) survey estimating that half of the projects can be characterized as following inadequate requirements management practices.

It is useful to differentiate between two broad categories of software defects: 1) defects in implementing *specified* user requirements due to design or coding errors, and 2) defects in the correctness of requirements due to discrepancies between *specified* user requirements and *true* user requirements. Software development practices have made impressive advances in preventing, identifying, and eliminating the first type of defects, including techniques such as quality function deployment, computer-aided software engineering, extreme programming, rapid application development, joint application development, unified modeling, defect tracking, and clean room [6], [39]. New frameworks and standards have come into prominence, including the Capability Maturity Model [51] and ISO 9000–9004 [39], which feature comprehensive processes for minimizing defects in how well a working system implements specified requirements. In contrast, techniques for verifying the correctness of specified user requirements have lagged behind [47].

Removing defects of any kind is much more expensive and threatening to a project's ultimate success the later in the project they are detected and fixed [39]. This penalty for delayed correction is especially acute for errors in getting requirements right. Boehm and Papaccio [13] have estimated that it costs from 50 to 200 times more to correct a requirements error at the time of maintenance compared to getting the requirements right in the first place. Many observers claim that the art and science of information systems development is facing a crisis. As Mann [43] points out, "one of the great puzzles of contemporary technology" is that although software has become increasing critical to almost every facet modern life, many practicing developers believe that software quality is not improving, and may in fact may be getting worse. One of the root causes that Mann traces the problem to is the incredible fact that "the purpose of new software is often not spelled out before programmers begin writing it." The Capability Maturity Model (CMM) emphasizes the need for "requirements management," which refers to establishing a common understanding between the customer and the software project team of the customer's requirements to be addressed by the software project. However, there do not appear to be any metrics specified by CMM for determining the correctness of requirements [51]. These observations all point to the need for better requirements management techniques for preventing, detecting, and eliminating defective requirements specifications as early as possible in the project life cycle.

It is important to recognize that requirements creep is not fully under the control of the software develop team or its management. It is inevitable that many projects will face a change in true user requirements during the life of a project, and developers must be prepared to cope with such changing needs. In fact, studies of the function point metric have found that the average amount of growth in unplanned functionality is 1% per month from the end of the requirements phase until the beginning of the coding phase, which translates to about 25% for a two-year project [39], [47]. Nevertheless, a key success factor differentiating successful from unsuccessful projects is the stability of user requirements. Jones asserts that successful IS projects tend to have requirements that are stable to within 5%. It is important to manage the tradeoff between flexibility to accommodate changing requirements and the cost, time, complexity, and risk of a project. Apart from the issue of coping with requirements creep, however, it should be of great value to focus on the identification, correction, and prevention of requirements errors that have been introduced in the original specification of requirements. Such errors are committed in the earliest stages of a software project, and, if identified early, would be more readily fixed before moving on to design and coding.

What if it were possible to predict the degree of post-implementation user acceptance of a large and complex new information system before writing a single line of program code, possibly before even building a working prototype, by presenting design specifications to potential users and asking them to answer a few survey questions? Would such capability have value? We are among the first to admit that such capability is impossible given our current knowledge in the field of information systems development. However, considering the large proportion of complex systems projects that survive the often-grueling development process only to become user acceptance failures after workplace implementation, researchers should investigate better ways to perform such "preprototype user acceptance testing."

Measuring user evaluations of new information systems during their development process has long been advocated by practitioners and researchers in the usability engineering tradition ([2], [12], [31]–[35], [44], [54]). Hands-on usability testing of new information systems is traditionally done toward the end of the system development project, when there is a nearly-final working system. Conventional wisdom in the information systems field argues that potential users must have direct hands-on experience with at least a realistic working prototype of a new system before they can provide assessments that are predictive of future usage behavior after workplace implementation. Unfortunately, by the time a new system development project has reached the stage of prototype construction, significant amounts of time, effort, and money have typically been expended. Many key decisions about the functionality to be included in a new system are made early in the design process, and it is often impractical and cost-prohibitive to introduce major changes in functionality after design features have already been implemented in program code and a working version of the new system is available for hands-on user testing. Moreover, usability testing focuses primarily on a system's ease of use, and typically fails to evaluate the usefulness of its functionality, which minimizes that chances that errors in requirements would be detected even at this late stage in development.

The present research challenges the widely-held assumption that hands-on usage experience is a necessary prerequisite for capturing user test measures capable of predicting post-implementation user acceptance in the workplace. While we acknowledge that stable and representative measures of perceived ease of use do generally require hands-on interaction with a working system, we hypothesize that stable and representative measures of perceived usefulness only require that potential users be informed of what a system will be designed to do, i.e., its intended functionality, and do not require hands-on interaction with a working system. Moreover, substantial previous research indicates that usefulness is generally a stronger predictor of usage intentions and behavior than is ease of use (e.g., [71]). If it can be shown that usefulness measures based on noninteractive descriptions of functionality can approximate usefulness measures taken after users have had direct experience using a working system or prototype, it would open the possibility of conducting user acceptance testing much earlier in the system development process.

The potential importance of preprototype user acceptance testing is illustrated by Fig. 1. The time course of an information system development process is depicted on the horizontal axis, and the vertical access is used to represent the magnitude of development time, cost, and effort as they increase over the life of the project, and the amount of flexibility to alter a system's functionality as it decreases over the life of the project. T1 refers to a point in time where the functional requirements of a system have been determined and documented but before any program code has been written. At T1, sufficient information is available to describe to potential users what a system is being designed to do, even if the details of specific commands, menus, screens, and data elements have not been identified.

The central question to be asked at T1 is whether or not a system with the specified functionality, if it is eventually developed, will be accepted and used by target users. T2 represents the traditional point in a development project's life

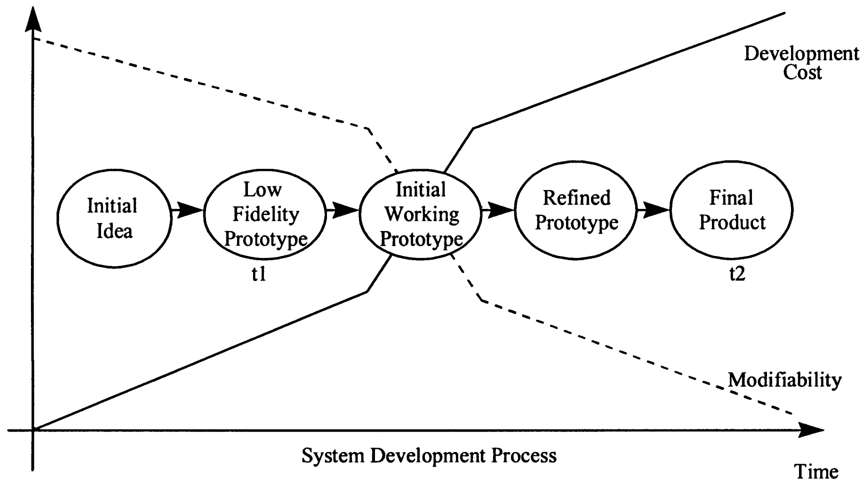


Fig. 1. Preprototype user acceptance testing.

when user acceptance testing is conducted using a realistic working prototype of the system. At T2, many detailed design decisions regarding what functionality to include have been made and implemented in working software, significant amounts of time, effort, and money have already been expended, and limited flexibility exists to introduce substantive changes to functionality. If preprototype usefulness measures taken at T1 are sufficiently consistent with postprototype usefulness measures that would ordinarily be captured at T2, early user acceptance testing might become viable. *If* preprototype user acceptance testing can provide valuable predictions postimplementation user acceptance, which we recognize is a very big "if" and the focus of the current research, then the following important practical advantages may be realized.

1) Preprototype user acceptance testing may provide valuable guidance on key project decisions such as: proceed with development of a system with the functionality currently specified, modify the target functionality prior to proceeding with development, or abandon the development project. This is a key benefit because less than 25% of the total cost has typically been incurred by the time functional specifications are defined (T1).
2) Even in the worst case, where the project is abandoned because preprototype measures reveal insufficient usefulness to motivate user acceptance of a future system having the specified functionality, as much as 75% of the total project cost may be saved. Many projects fail because the system's functionality does not sufficiently match users' workplace needs. Preprototype user acceptance testing holds the promise of being able to identify troubled projects that are headed for failure early enough to substantially cut losses.
3) In addition to the possibility of saving time and money, preprototype measures come at a time (T1 in Fig. 1) when there is high flexibility to modify the specifications of the potential system's functionality, since none of the specifications have yet been committed to working program code.
4) Given the relatively small amount of time, effort, and money (compared to building a working prototype) of presenting to users a potential system's functionality, multiple alternative system design concepts can be evaluated at the preprototype stage, each including differing features and functionalities. A comparison among competing alternatives could be used to prioritize features, guide the allocation of development resources, and identify designs with a positive prognosis for eventual workplace acceptance.

These potential benefits are premised on the as yet untested assumption that system developers can rely on preprototype usefulness measures as surrogates for postprototype measures captured from potential users after hands-on experience with a representative working system. The present research embarks on the investigation of how well such preprototype user acceptance tests predict actual acceptance and on-going use of a new system after it has been implemented in the workplace. The two studies reported in this paper seek to test a basic yet critical assumption: *Can predictively accurate assessments of the likelihood of workplace acceptance of a new system be made based on measures taken from prospective users who have been informed about the features and functionalities to be included in the new system, but have not yet had an opportunity to experience hands-on interaction with it?*

II. Theoretical Lens

The software project management literature has emphasized prototypes and measures of user reactions to prototypes as a key input to making changes to a system. In this research, we draw from user acceptance research to inform how software projects may be better managed through appropriate preprototype user acceptance tests. Specifically, the Technology Acceptance Model (TAM) [18], [19], *et al.* [21], one of the most widely researched models in the user acceptance literature, was employed to study the viability of preprototype user acceptance testing. TAM was specifically designed to represent the causal relationships linking a system's design characteristics to its acceptance and usage in the workplace. According to TAM, which is founded on intention models from psychology [4], [26], [67], users' intentions are the single best predictor of behavior (i.e.,

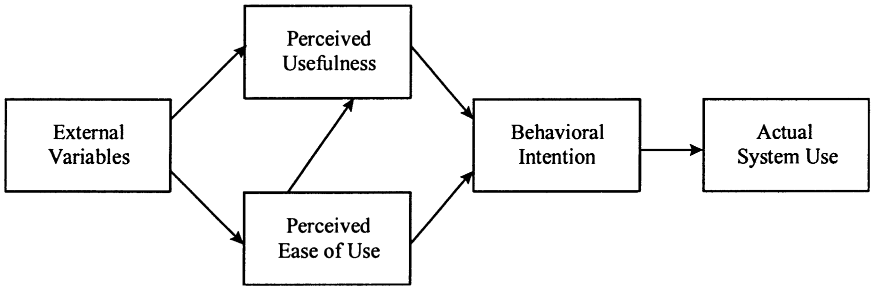


Fig. 2. TAM.

actual system usage). Research has supported the strong relationship between intention and behavior in a variety of domains (see [3] for a review and [56] for a meta-analysis), including information system use [21], [62], [64], [74]. TAM posits that behavioral intention to use a system (BI) is determined by perceived usefulness (U), the user's perception of the degree to which using the system will improve his or her performance in the workplace, and perceived ease of use (EOU), the user's perception of the amount of effort needed to use the system [18], [21]. Further, perceived usefulness and perceived ease of use are theorized to mediate the effect of external variables, such as system design characteristics and training, on behavioral intention to use a specific technology. In addition, TAM suggests that perceived usefulness is also determined by ease of use perceptions—the logic being that the easier a system is to use, the more useful it can be. TAM is particularly suited to the study of user reactions to preprototypes because of the beliefs in TAM, i.e., perceived usefulness and perceived ease of use. Perceived usefulness relates to user evaluations of functionality and perceived ease of use relates to user assessment of the interface. Thus, TAM presents an excellent fit to examine the viability of preprototype user acceptance testing by focusing on the critical predictors. Fig. 2 presents a pictorial representation of TAM.[1]

Subsequent to its development, the reliability and validity of TAM and its measurement scales have been demonstrated extensively in prior research across several applications and replications (e.g., [1], [14], [18], [19], [21], [22], [27], [36],[37], [46], [55], [60]–[62], [64], [71], [74]). While, as noted, there has been a significant body of work focused on TAM, little work has actually taken a "dynamic" view of user acceptance and examined the role of time and experience in a single model. When comparisons across levels of experience have existed, these have been predominantly cross-sectional assessments or between-subjects comparisons [40], [66]. Similarly, in other work, even though longitudinal data existed, only cross-sectional tests were conducted (see [72] and [74]). To address these limitations, we will develop and test a single, dynamic model that includes *preimplementation user reactions*, *postimplementation user reactions*, and *actual usage behavior*. We will first examine the TAM relationships based on preprototype user assessments; next, we will examine the stability of key constructs (perceived usefulness, perceived ease of use, and behavioral intention) over time; and finally, we will assess the predictive validity of early perceptions.

[1]Although, the original conceptualization of TAM included the attitude construct, the final model excluded the attitude construct because attitude did not fully mediate the effect of perceived usefulness and perceived ease of use on intention (see [21, pp. 995–996]); recent extensions to TAM also exclude the attitude construct [71], [72].

III. HYPOTHESIS DEVELOPMENT

Why do we hypothesize that preprototype measures of perceived usefulness will approximate usefulness measures taken after hands-on experience using a working system? Why do we accept the conventional wisdom that preprototype ease of use measures will not be good substitutes for ease of use measured after hands-on use of a working system? Why do we expect that preprototype measures of future usage intentions will be useful indicators of post-hands-on measures, especially given that we acknowledge that one of the known determinants of intentions is not stable across these two times of measurement? In this section, we discuss the psychological processes theorized to underlie the formation of perceived usefulness, perceived ease of use, and behavioral intentions to use a specific technology, and examine the role of direct behavioral system usage experience in the formation and change of these key predictors of ultimate system acceptance and use. We present a summary research model that captures the predictive validity of early measures of user acceptance of new technologies.

A. Perceived Usefulness

Perceived usefulness is defined as the extent to which the individual believes that using a system will enhance her/his job performance. We posit that people form perceived usefulness judgments by comparing what a software product is capable of doing with what they need to get done. We develop the theoretical justification for a matching account from different theoretical perspectives by primarily drawing from action identification theory [69]; we also justify the arguments by drawing from image theory [10], [11], motivation model of technology acceptance [20], and knowledge and learning theories [7]. Drawing from these theoretical underpinnings, we suggest that usefulness perceptions formed prior to hands-on experience will be predictive of subsequent usefulness perceptions (based on significant hands-on experience).

Any given behavior can be mentally represented in a "cognitive hierarchy ranging from detailed, mechanistic depictions of the action ('low-level identities') to comprehensive consequence-defined depictions ('high-level identities')" ([68]; see also [69]). Vallacher and Kaufman [68] further explain this hierarchy with an example of an individual traveling to Munich. The lowest level identities include sitting in a seat, looking out the window, dealing with toilet facilities; at the next higher level, the identities include getting to a conference, traveling; at the highest level, the identities include adding to one's life experience, straining international relations. A similar identity hierarchy in the case of software use could be striking keys at the lowest level, writing a report at a higher level, and achieving one's annual sales goals at the highest level. The level of identification is critical in understanding mental control of action because people prefer to think about action and consequences in comprehensive terms, i.e., people represent actions at the highest level of identification possible [68], [69]. In addition, higher levels of identification are related more closely to goals and plans.

Given that perceived usefulness is an assessment of the consequences of a use-performance contingency, action identification theory would suggest that individuals will use higher level of identities such as those related to their job in forming usefulness judgments. Kieras and Polson [41] suggest that users possess distinct knowledge about their job situation, which they use as a basis for determining what tasks can be performed with a given system [53, p. 186]. Within the cost-benefit paradigm of decision research, Creyer *et al.* [16, p. 2] propose a similar matching process in the formation of accuracy judgments, which are similar to usefulness. If people indeed use existing knowledge about their important job tasks as a backdrop against which to assess a system's usefulness, then in order to acquire well-formed and stable usefulness judgments, it should be adequate to explain what a system is capable of doing using noninteractive prototypes, without necessarily providing direct experience using the system. While performance feedback from direct experience may be an additional source of information affecting usefulness judgments, it may not be necessary given a matching-based process, especially since such performance feedback is generally ambiguous because one is unable to observe the comparative performance associated with actions not taken [25]. Further, given that usefulness perceptions are related to a use-performance contingency [18], [21] and are representations at higher levels of identification [69], they are driven more by ends or outcomes, thus reflecting the goal-directedness of the behavior of using technology.

The use of noninteractive preprototype mockups as a basis to help users establish well-formed usefulness judgments is also supported by image theory [10], [11], motivational models of technology acceptance [20], and knowledge/learning theories [7]. Perceived usefulness is in line with image theory's notion of a trajectory image that is a set of adopted goals that guide action at the highest level. If individuals indeed possess knowledge about their tasks and jobs [41], they will be able to create a fairly accurate trajectory image based on noninteractive prototypes. Using a motivational model of technology acceptance, Davis *et al.* [20] found that perceived usefulness was determined by task importance moderated by the effect of system attributes, suggesting that people do in fact judge the usefulness of a system's capabilities in terms of how well they fit important job tasks (see also, [32] and [66]). From the perspective of knowledge and learning theories (e.g., Anderson [7]), we expect declarative knowledge that outlines concepts to enable potential users to assess the cause-effect contingencies relating actions to outcomes and, thus, to provide the primary basis for the formation of usefulness perceptions. Such declarative knowledge in the case of a technology is closely related to the functionality of a system [7], [49]. Given that individuals use their job and tasks as a backdrop to assess the functionality (see [74]) and, thus, usefulness of a system, it can be expected that noninteractive prototypes that outline a system's features will allow users to form fairly accurate perceptions about the usefulness of a system. Thus, based on a triangulation of research from different areas, we expect that people do not require direct hands-on interaction with a system in order to establish well-formed and stable usefulness perceptions.

H1: Perceptions of usefulness based on user exposure to preprototype requirements specifications will be highly predictive of similar perceptions taken after significant hands-on experience with a working system having those specifications.

B. Perceived Ease of Use

Perceived ease of use is defined as the extent to which an individual believes using a system will be free of effort. Unlike perceived usefulness, it is expected that perceived ease of use will require direct experience to become well-formed, and will, therefore, not be stable over time if noninteractive mockups are used. Conventional wisdom suggests that actually performing (or attempting to perform) a behavior is a prerequisite for accurately judging how easy or difficult it is, and theory and research bear this out. The key paradigm that the ease of use construct is based on is self-efficacy theory [9]. Bandura defines self-efficacy as "…judgments of how well one can execute courses of action…" [9, p. 122], which provides a basis for the definition of perceived ease of use [18]. Bandura argues that the primary source of information used to form self-efficacy judgments is direct experience performing the target behavior, what he terms "enactive attainments" [9, p. 126]. Gist and Mitchell similarly stress the importance of exposure to a task via direct behavioral experience for forming "strongly held, stable, and accurate" self-efficacy beliefs [29, p. 192]. In the context of computer use, Gist *et al.* [30] explicitly define software self-efficacy as being based upon direct hands-on experience. There exists theoretical and empirical evidence suggesting that before hands-on experience, user perceptions about ease of use would be anchored to various general computer beliefs about computer use, and that after direct experience, ease of use perceptions would be adjusted to reflect various aspects of the experience (Venkatesh [71]).

The importance of direct hands-on experience in the formation of ease of use perceptions is also supported by the theoretical perspectives used to examine the role of direct experience in the formation of usefulness judgments. Ease of use perceptions are tied to identities at the lower level per action identification theory. Such low level identities are transient and unstable mental representations [68], [69] that can be expected to change with increasing behavioral experience and the passage of time. From an image theory perspective, perceived ease of use is closely related to what is termed a strategic image that focuses on means and tactics that relate to concrete behavioral actions such as hands-on system usage.

From a knowledge and learning perspective [7], perceived ease of use is based on procedural knowledge. Anderson [7] suggests that "…procedural learning occurs only in executing a skill; one learns by doing. This is one of the reasons why procedural learning is a much more gradual process than declarative learning" [7, p. 215]. Further, it has been argued that procedural knowledge is formed through a process called knowledge compilation that is a gradual accumulation of a set of task-specific productions (proceduralization) complemented by a process of combining a sequence of productions into a single production (composition) [7]. In other words, the process of knowledge compilation related to skill acquisition occurs as a consequence of practice. Therefore, perceptions of ease of use, which reflect the extent of difficulty/ease associated with technology use, will not be well-formed based on noninteractive mockups and will require hands-on experience. Thus, we do not expect perceived ease of use measured before hands-on use to remain stable.

H2: Perceptions of ease of use based on user exposure to preprototype requirements specifications will not be highly predictive of similar perceptions taken after significant hands-on experience with a working system having those specifications.

C. Behavioral Intention to Use

TAM posits that behavioral intention to use a technology is a function of perceived usefulness and perceived ease of use. Such intentions can be more specifically termed implementation intentions (i.e., intending to use a system on the job) that guide the achievement of goal intentions (i.e., intending to improve performance in the workplace). The notion of implementation intention being a function of perceived usefulness and perceived ease of use is also consistent with research in image theory that would suggest that intentions and behavior are guided by a trajectory image (perceived usefulness) that is linked to a strategic image (perceived ease of use). While we expect that perceived usefulness based on a noninteractive mockup (preprototype) will be well-formed and stable when compared to perceptions measured after significant hands-on experience, we also expect that perceived ease of use will not be stable prior to direct hands-on experience with the system. Thus, in order to understand the stability of intention over time with increasing experience, the relative influence of perceived usefulness and perceived ease of use in determining intention is critical.

Davis *et al.* [21] theorized perceived usefulness to be a more important determinant of intention when compared to perceived ease of use. They justified the argument on the basis that in a workplace setting, which typically emphasizes productivity, a rational factor that is an individual's assessment of the performance outcomes associated with technology use (i.e., perceived usefulness) will be the single most important determinant of usage intentions and usage behavior. Empirical studies, spanning a range of different systems and user populations, have found perceived usefulness to be a stronger determinant of intention/usage than perceived ease of use. Specifically, Davis [18] found the path coefficient for perceived usefulness to be greater than that of perceived ease of use for an electronic mail package (0.55 versus 0.01), a text editor (0.69 versus 0.02), and two different graphics packages (0.69 versus 0.08 and 0.76 versus 0.17). Davis *et al.* [21] also found perceived usefulness to be more important in a study of user reactions to a word-processing package both after a 1-h training session (0.62 versus 0.20), and 14 weeks after workplace implementation (0.71 versus −0.06). Similar results were reported in other prior research ([1], [22], [46], [61], [73]–[75]).

On both theoretical and empirical grounds, perceived usefulness is a stronger determinant of intention than is perceived ease of use. Further, as users gain significant hands-on experience with a system, the direct effect of perceived ease of use on intention often diminishes to the point of nonsignificance (e.g., Venkatesh, *et al.*, [75]), leaving perceived usefulness as the sole determinant of intention. Given that perceived usefulness is expected to be stable over time, and intention is most strongly determined by perceived usefulness, users are not expected to require direct hands-on interaction with a system to establish well-formed and stable intentions to use a technology. Thus, it is expected that behavioral intention to use a system, formed on the basis of a noninteractive preprototype, will be stable over time with increasing experience.

H3: Behavioral intention based on user exposure to preprototype requirements specifications will be highly predictive of similar perceptions taken after significant hands-on experience with a working system having those specifications.

D. Summary Research Model

Intention to use a technology has been consistently shown to be the strongest predictor of usage behavior in a wide variety of studies within user acceptance of technology (e.g., [72]) and other areas [1], [56]). Such a pattern of predictive validity was shown in the context of self-reported usage behavior [21] and actual usage behavior [74]. Thus, we expect that intention would continue to be a strong predictor in the context of this research as well. However, one of the underexplored areas in IT adoption research is the role of early usage behavior in predicting sustained use. In the current research, this is crucial as the predictive validity of early intentions determine the success of preprototype user acceptance testing. There exists significant evidence in psychology to suggest that previous behavior will be the most important determinant of future behavior [8], [15], [48]. A recent meta-analysis also supports this pattern [50]. Based on this, we theorize that prior usage behavior will be the only predictor of future usage behavior. The proposed research model is shown in Fig. 3. It is important to note that the stability of perceptions and intentions predicted by H1–H3 manifests itself in the research model in terms of early perceptions driving early usage and subsequent usage being driven by early usage. Alternatively, if key perceptions and intentions were expected to change over time, this would have manifested in an influence on later usage behavior. As such, key preprototype perceptions and intentions influence early usage behavior that in turn influences sustained usage behavior.

It should be noted that the research model here suggests that user reactions at each stage of experience indeed drive intention at that time which, in turn, *could* drive subsequent usage behavior. Such predictions are generally consistent with previous

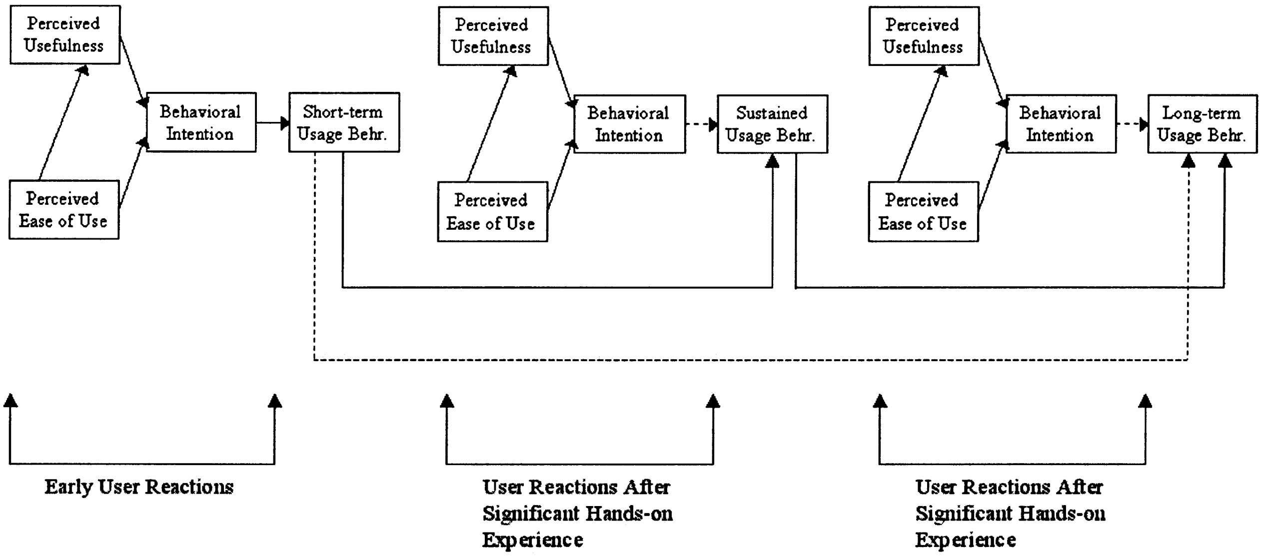


Fig. 3. Proposed research model.

user acceptance research (see, for example, [40], [64], and [72]). In other words, the basic TAM is expected to possess cross-sectional validity. However, the key strength of the new *dynamic model* proposed here is that not only are the early user reactions the key driver of short-term usage behavior but also after *controlling* for such short-term usage behavior, subsequent user reactions do *not* explain any additional variance in sustained usage behavior. Thus, in general, this renders cross-sectional model tests at later stages of user experience to be less pertinent.

IV. Method

A. Research Approach

In practice, user evaluations during early stages of system design and development would be used to modify/improve the system. However, for purposes of the current research, we are interested in how well these evaluations obtained at early stages would predict what would be obtained *if there were no changes made to these early system design ideas*, i.e., to determine whether changes are required. There are two approaches that can be used to study this issue.

1) One approach would be to use a system which is actually in its very early stage of design and development and get user reactions to the preprototype mockups, ensure that no changes are made to these early system ideas, and proceed with system development consistent with the initial design. For our purposes, changes in the design would need to be avoided because they would confound the assessment of representativeness of early measurements. Once the system is built, users can be asked to interact with and evaluate it in order to determine the representativeness (predictiveness) of early user assessments over time.
2) The second approach would be to use systems that already exist and recreate mockups that approximate those that would have been available during the early stages of the system development process. By exposing users to these recreated mockups, user reactions to system ideas during early stages of the design process can be obtained. The users can then interact with the actual system for up to several weeks and user reactions can be obtained again to assess the stability of user assessments over time. Thus, in effect, simulating the progression from a preprototype to a working version of the software product.

In this research, we employed the second approach. To employ the second approach, we created simple mockups, which could have been available at the early stages, and assessed user reactions at multiple points in time. The mockups correspond to a very early stage of the design process, before even a prototype has been built. We employed functionality descriptions, story boards, and screen designs from the early stages of system design. In this research, in addition to addressing the question of stability of user evaluations over time, we also attempted to address the question of how early can user assessments be obtained that would be stable over time by varying the detail and type of the mockups (from story boards to one page write-ups) and duration of the lecture (from no verbal instruction to a half-day training program). It would be normal for system ideas to be fairly fuzzy in the early stages of design and development. Therefore, it follows that mockups such as simple text descriptions, which do not delve into much detail as to the specifics of the software product, can be the simplest form of mockups produced very early in the process of design and development even before the system prototype has been built, with minimal investment of time, effort, and money. This is also a very stringent test for stability of user assessments over time given the limited amount of information about the software packages supplied to the users in the write-ups. In order to assess how well measurements taken after exposure to a preprototype mockup approximate what would have been obtained using traditional prototype testing and pilot programs, the participants then acquired extensive hands-on experience.

Typically, mockups such as functional descriptions, storyboards, and screen designs are used to gather user reactions and suggestions for modifications. Such a traditional approach is relying on an untested assumption that users can indeed form accurate judgments regarding a new system based on mockups.

This is a critical assumption since designers often make changes to the design based on user reactions. It is important to test this assumption because: 1) if users are unable to form stable reactions, placing faith in user judgments based on mockups is ill-advised or 2) if users are able to form stable judgments after being exposed to simple mockups, it creates the opportunity for designers to evaluate multiple design ideas and iterate through the design process in greater depth following up on several of the designs which are viewed favorably, change, and eventually move forward with development and implementation based on the most favorably perceived mockup.

Two longitudinal field experiments, spanning six months each, were conducted to assess the viability of preprototype user acceptance testing. Experiment 1 presented a within-subjects assessment of how well early measures predicted user acceptance three months into workplace implementation. Experiment 2 employed a within-subjects and between-subjects evaluation of preprototype user acceptance testing.

B. Measurement

For all three experiments, three constructs from the TAM were measured for each of the software packages: perceived usefulness, perceived ease of use, and behavioral intention. The measurement scales and the questions used were adapted from Davis [18], who developed, validated, and extensively pretested them, showing them to have high reliability and construct validity. Further replications and applications (listed earlier in the theory section) support the strong psychometric properties of these measures. Four questions each were used to measure U and EOU, and two questions were used to measure BI. Actual usage behavior was measured as duration of use via system logs.

V. Experiment 1

This experiment was designed to allow a within-subjects comparison to examine how early reactions of participants exposed to a preprototype will predict their later reactions formed on the basis of significant hands-on experience. Further, it will allow us to test the proposed research model that suggests that only the early user reactions will drive their short-term and sustained usage of the new software system.

A. Participants and System

One-hundred and six customer service representatives, including 16 women, in a small financial services firm participated in the within-subjects study, and 92 provided usable responses at all three points of measurement. The average age of the participants was 38.2 years with a range from 22 to 56 years. Individual experience with computers ranged from one year to 14.5 years, with an average of just under six years.

The new system being introduced was a mainframe-based system for customer account management. The proposed system was expected to serve as a possible replacement of the existing manual and computer-based systems (e.g., Excel) used by representatives. Based on interviews with two senior managers in the organization, it was determined that actual system use was voluntary, and the representatives could continue using their current method(s) of account management.

B. Procedure

The formal training session was designed to be based on a preprototype version of the system. The session lasted four hours, excluding a one half-hour break between each of the 2-h components. The two components were conceptual training and procedural training. Conceptual training was a 2-h session, and focused on the features of the system. Procedural training, which was also for two hours, was a walk-through of the specific steps required to use the several features of the system. The participants were acquainted with the procedure required to use the system with the aid of a hand-out of screen designs of the system, and a description of the commands required to access and use different features. The hand-out of the screen captures was built on the lines of a concise user manual that described the procedures to use the system. Following the training (T1), the participants filled out an online questionnaire. The system captured the participants' login ID and matched it with his or her responses, this information was used to generate a bar code to help track corresponding responses over time. Following the initial training, participants had an opportunity to use the system as part of their jobs for several weeks. User reactions were gathered [via a questionnaire similar to what was used at (T1)] after one month (T2) and three months (T3), respectively. Although the participants' identity was important in tracking responses on time, the confidentiality of participants was maintained with no one other than the researchers having access to the information. Actual usage (duration of use) was tracked using system logs for six months, with T4 representing the last three months of the study.

C. Results

Partial least squares (PLS) was employed to analyze the data. The preliminary analyzes included calculating the reliability and validity statistics. Table I reports the ICRs, AVEs, descriptive statistics, and correlations. All ICRs were greater than 0.85, thus supporting reliability. All AVEs were greater than the correlations, thus supporting convergent and discriminant validity. Finally, loadings on all constructs were greater than 0.80 and the cross-loadings were less than 0.30 (not shown here)—this is very consistent with previous TAM research that has established the validity of the TAM scales.

The predictions in this research pertain to the predictive validity of measures of perceived usefulness, perceived ease of use, and behavioral intention. Specifically, perceived usefulness and behavioral intention are expected to be stable but perceived ease of use is not expected to be stable. Table II presents the means, the results of paired mean difference tests, and paired correlations across measures taken at different points in time. In the case of perceived usefulness and intention, high correlations and nonsignificant mean differences reveal support for theorized pattern. In contrast, perceived ease of use was found to be different from T1 to T2 and T1 to T3, but not T2 and T3. An ANOVA followed by a Scheffe's test confirmed these results. This pattern of results suggests the possibility that users can form stable usefulness perceptions and intention without hands-on use, but require direct behavioral experience in order to form stable ease of use judgments.

TABLE I
STUDY 1-RELIABILITY AND VALIDITY ANALYSIS

	ICR	U1	EOU1	BI1	USE12	U2	EOU2	BI2	USE23	U3	EOU3	BI3	USE34
U1	.90	.87											
EOU1	.92	.24**	.88										
BI1	.90	.55***	.24*	.91									
USE12	N/A	.41***	.28***	.62***	1.00								
U2	.94	.61***	.12	.39***	.40***	.92							
EOU2	.88	.20*	.40***	.10	.29***	.26**	.87						
BI2	.91	.37***	.20*	.49***	.50***	.50***	.23**	.93					
USE23	N/A	.40***	.20*	.44***	.64***	.38***	.21*	.49***	1.00				
U3	.87	.60***	.12	.30***	.31***	.60***	.22*	.38***	.38***	.87			
EOU3	.95	.17*	.41***	.15	.13	.13	.38***	.19*	.21**	.25**	.92		
BI3	.96	.31**	.05	.53***	.30**	.30**	.16*	.51***	.40***	.51***	.25**	.92	
USE34	N/A	.32***	.09	.36**	.36***	.36***	.17*	.31**	.60***	.36***	.20*	.50***	1.00

Notes:
T1 measures: U1, EOU1, BI1
T2 measures: U2, EOU2, BI2
T3 measures: U3, EOU3, BI3
USE12: Usage between T1 and T2
USE23: Usage between T2 and T3
USE34: Usage between T3 and T4
ICR = Internal Consistency Reliability

Diagonal elements are the square root of the shared variance between the constructs and their measures. Off-diagonal elements are the correlations between the different constructs.

* p<.05; ** p<.01; *** p<.001

TABLE II
STUDY 1-MEAN DIFFERENCES AND CORRELATION ANALYSIS

	T1 M (SD)	T2 M (SD)	T3 M (SD)	Mean Differences			Correlation Analysis		
				T1-T2	T1-T3	T2-T3	T1-T2	T1-T3	T2-T3
U	4.6 (0.9)	4.5 (0.8)	5.0 (1.0)	ns	.ns	ns	.54***	.56***	.60***
EOU	4.0 (1.0)	4.9 (0.8)	5.0 (0.9)	p<.05	p<.05	ns	.16	.09	.38***
BI	4.5 (1.0)	4.6 (0.9)	4.5 (0.7)	ns	ns	ns	.49***	.53***	.51***

* p<.05; ** p<.01; *** p<.001

Given the support for the basic stability ideas, we tested the research model using PLS. Fig. 4 presents a pictorial summary of the pattern emerging from the distributed lag analysis. The results suggest that U and EOU measured after noninteractive, mockup-based training (T1) were predictive of usage between T1 and T2 (post-training to one month) post-implementation. In turn, the early usage predicted continued use between T2 and T3 (one month post-implementation to three months post-im-

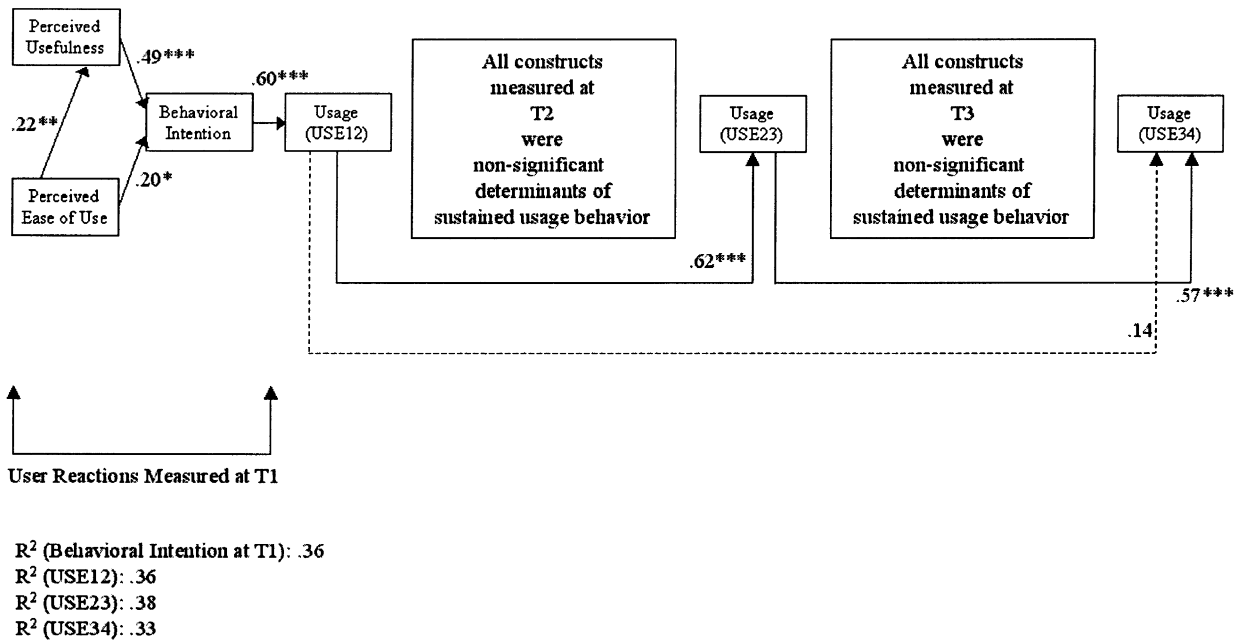


Fig. 4. Study 1: Distributed lag model.

plementation). Supporting the basic theorization that early measures possess predictive validity, U, EOU, and BI measured after significant direct experience (T2 and T3) did not add any explanatory power in usage behavior over and above previous usage. Similarly, usage behavior between T2 and T3 predicted usage behavior between T3 and T4 (three months to six months post-implementation). Thus, based on the predictive power of early intentions and the role played by U and EOU in determining these early intentions, it is possible to suggest that early user perceptions (measures taken even before hands-on experience—T1) can provide useful information regarding system success up to six months after workplace implementation.

VI. Experiment 2

Given the encouraging results from Experiment 1, we sought to replicate and extend the work. Experiment 2 was conducted in a different organization, thus creating the potential to assess and enhance external validity of the research findings from Experiment 1. While Experiment 1 was a within-subjects design, Experiment 2 was designed to include within-subjects and between-subjects comparisons to assess the viability of preprototype user acceptance testing. As noted in the description for Experiment 1, a within-subjects analysis allows us to compare early reactions based on users' exposure to a preprototype to reactions subsequent to significant hands-on experience. In order to complement such a study, we also study user reactions based on exposure to a prototype (rather than a preprototype) during the initial training. Such a group will allow us to directly compare mean differences between the preprototype group and the prototype group. Further, we can examine the stability of the two groups separately. Finally, the predictive validity of the research model can be examined for each of the two groups separately and for the data pooled across groups.

A. *Participants and System*

One-hundred and forty employees, including 36 women, from a small accounting firm participated in the study, with 116 providing usable responses at all three points of measurement. The average age of the participants was 33.4 with a range from 24 to 49. Average experience with computers ranged from 1 to 17 years, with an average just over five years. The new system being introduced was a Windows-based system to provide access to internal organizational information, ranging from policies/procedures to client information. Based on an interview of three members of the senior management, it was determined that the new system use was voluntary, and provided an alternative to existing manual systems.

B. *Procedure*

Experiment 2 was designed to assess the viability of preprototype user acceptance testing via a within-subjects and between-subjects study by comparing preprototype user assessments with user assessments based on a traditional lecture/hands-on training. The initial training included two different treatments: 1) noninteractive, mock-up based training ($n = 56$) and 2) traditional lecture/hands-on training ($n = 60$). Employees attended the four-hour training program scheduled in the morning or the afternoon of one of the two days when the training was conducted. Participants were randomly assigned to one of the two treatments, with each treatment in a session being capped at 30. Two teams conducted the training and the design was counterbalanced for team—for example, team 1 conducted the mockup-based training in the morning and the traditional training in the afternoon, and team 2 did the reverse.

The noninteractive, mock-up based training included a 2-h lecture on the concepts and features of the system. The second half of the training session was a 2-h lecture on the procedures

TABLE III
STUDY 2-RELIABILITY AND VALIDITY ANALYSIS

	ICR	U1	EOU1	BI1	USE12	U2	EOU2	BI2	USE23	U3	EOU3	BI3	USE34
U1	.91	.86											
EOU1	.90	.22*	.90										
BI1	.93	.53***	.21*	.88									
USE12	N/A	.44***	.23***	.60***	1.00								
U2	.91	.71***	.18*	.41***	.39***	.87							
EOU2	.95	.19*	.43***	.15*	.23***	.23**	.84						
BI2	.92	.33***	.18*	.59***	.51***	.51***	.21**	.94					
USE23	N/A	.43***	.22*	.41***	.64***	.35***	.21*	.45***	1.00				
U3	.88	.73***	.13	.28**	.30***	.70***	.20*	.36***	.35***	.90			
EOU3	.91	.15	.49***	.11	.18*	.19*	.48***	.10	.18*	.21*	.89		
BI3	.92	.28**	.10	.58***	.32***	.28**	.18*	.61***	.45***	.50***	.22*	.88	
USE34	N/A	.30***	.13	.37***	.37***	.35***	.15	.27**	.62***	.33***	.20*	.49***	1.00

T1 measures: U1, EOU1, BI1
T2 measures: U2, EOU2, BI2
T3 measures: U3, EOU3, BI3
USE12: Usage between T1 and T2
USE23: Usage between T2 and T3
USE34: Usage between T3 and T4
ICR = Internal Consistency Reliability

Diagonal elements are the square root of the shared variance between the constructs and their measures. Off-diagonal elements are the correlations between the different constructs.

* p<.05; ** p<.01; *** p<.001

of the system using storyboards and screen designs, and included a discussion of the procedures related to 20 different tasks. The participants followed along with the lecture on the procedures with hand-outs that corresponded to the storyboards and screen designs depicted on the overhead projector of the instructor. In the traditional training, participants received a 2-h lecture on the concepts (identical to participants in the preprototype training), and received hands-on experience by performing ten different tasks (a subset of the tasks explained to participants in the other treatment) on the system by following along with an instructor demonstration (approx. 1.5 h). Following this, the participants had the opportunity to practice and reinforce their learning on a different set of ten assigned tasks (these were the remaining tasks among the 20 discussed in the other treatment). The measurement process was similar to Experiment 1—participants filled out an online questionnaire immediately after training (T1). The system captured individuals' network login ID along with their responses. As in Experiment 1, this was used to create a bar code to help track respondents over time. Also, as in Experiment 1, participants' confidentiality was preserved. Participants had access to the systems soon after training. Reactions were gathered from users after 1 month (T2) and 3 months (T3) using a questionnaire similar to the one used at T1. Usage behavior was measured for six months post-implementation, with T4 representing the last three months of the study.

C. Results

The preliminary analyses establishing the reliability and validity of the scales revealed results consistent with Experiment 1. Table III reports the ICRs, AVEs, descriptive statistics, and correlations. Similar to the results for Experiment 1, the loadings and cross-loadings also revealed strong support for convergent and discriminant validity (details not shown here). The general TAM causal hypotheses (see Fig. 2) were supported in Experiment 2 at all points of measurement. This overall pattern of preliminary results is consistent with Experiment 1 and much prior TAM research. Once again, because of this consistency with previous research, the details of these analyses are not reported here in the interest of brevity. PLS was also used

TABLE IV
STUDY 2-MEAN DIFFERENCES AND CORRELATION ANALYSIS

		T1 M (SD)	T2 M (SD)	T3 M (SD)	Mean Differences			Correlation Analysis		
					T1-T2	T1-T3	T2-T3	T1-T2	T1-T3	T2-T3
Pre-prototype User Group	U	5.4 (0.9)	5.4 (0.8)	5.3 (0.9)	ns	ns	ns	.58***	.62***	.59***
	EOU	5.0 (1.0)	5.9 (0.9)	6.1 (0.8)	p<.05	p<.05	ns	.12	.13	.41***
	BI	5.2 (0.9)	5.5 (0.8)	5.5 (0.7)	ns	ns	ns	.51***	.54***	.58***
Hands-on User Group	U	5.5 (0.8)	5.4 (0.9)	5.5 (1.0)	ns	ns	ns	.57***	.59***	.59***
	EOU	4.7 (1.0)	5.9 (0.9)	5.9 (0.8)	p<.01	p<.01	ns	.18	.16	.37***
	BI	5.4 (1.1)	5.6 (0.8)	5.5 (1.1)	ns	ns	ns	.50***	.55***	.57***
Pre-prototype vs. Hands-on Groups Mean Differences	U	ns	ns	ns						
	EOU	ns	ns	ns						
	BI	ns	ns	ns						
Pre-prototype vs. Hands-on Groups Correlations	U	.73***	.70***	.78***						
	EOU	.43***	.53***	.58***						
	BI	.60***	.61***	.63***						

* p<.05; ** p<.01; *** p<.001

to further analyze the validity and reliability—Table III reports the results of this analysis.

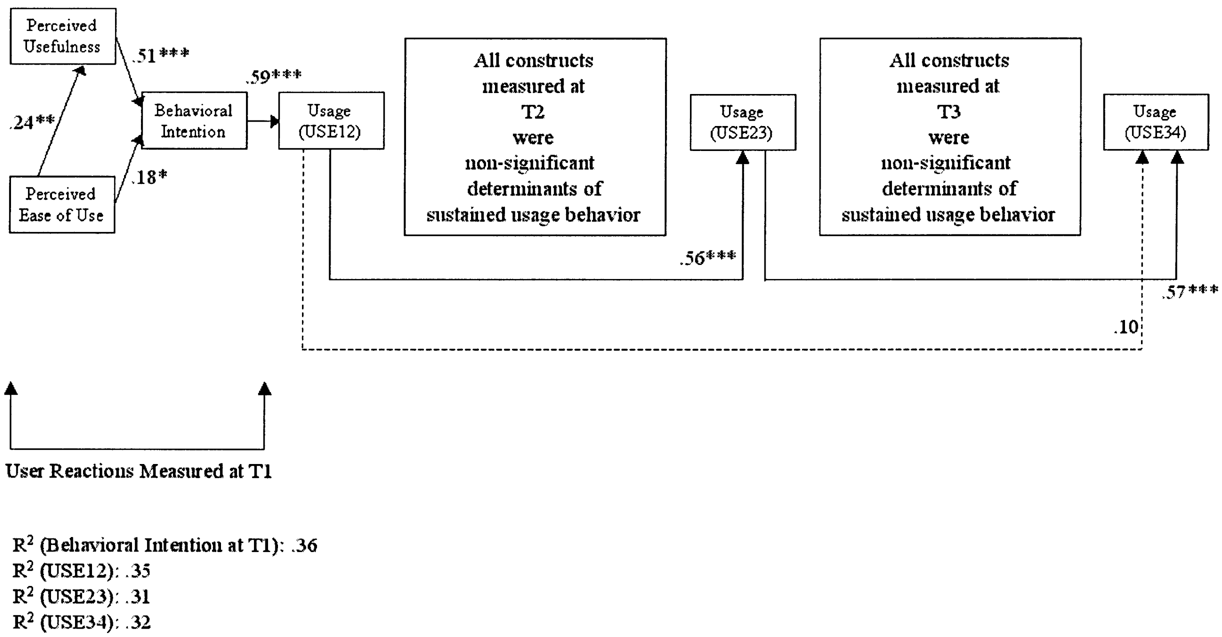
In order to examine the predictive validity of early measures of TAM constructs, we performed data analysis similar to the analysis in Experiment 1, and conducted some additional analysis. First, we conducted paired mean difference and correlation analyses. In this experiment, in addition to the analysis of users in the preprototype group (Table IV) similar to experiment 1, we also conducted an analysis of the users in the hands-on training group (also shown in Table IV).

For the preprototype group, Table IV indicates a high correlation between early measures (T1) of perceived usefulness and measures taken after several weeks (one month and three months-T2 and T3, respectively) after workplace implementation. Similarly, early measures of intention were also highly correlated with later measures. On the other hand, perceived ease of use was less stable. This pattern of stability of U and BI measures was also reflected by nonsignificant mean differences among measures taken at different points in time; EOU measures, in contrast, were significantly different over time in both interventions, confirmed by an ANOVA followed by a Scheffe's test ($p < 0.05$). This pattern of results was consistent with Experiment 1. Table IV also presents the analysis for the hands-on group at different points in time. The results were identical to the preprototype group. Further, an analysis of differences across treatments revealed no differences, suggesting that the various critical constructs behave similarly regardless of the type of early experience (preprototype versus hands-on).

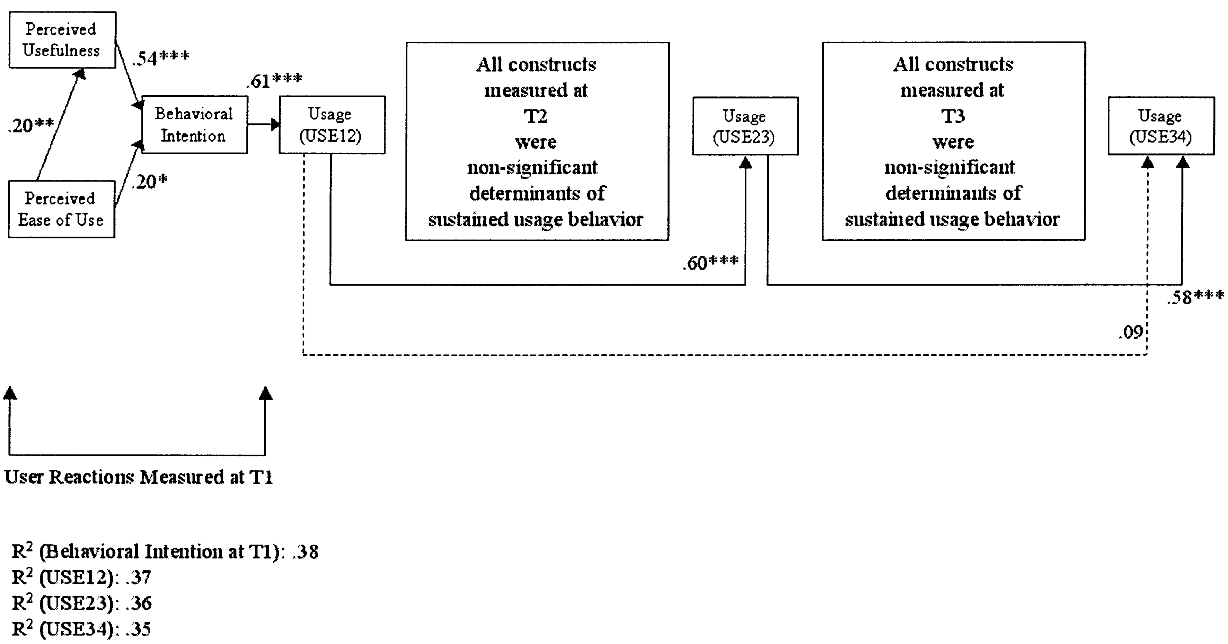
Similar to Experiment 1, we tested the research model using PLS. Fig. 5(a) and (b) present a more pictorial summary of the pattern emerging from the distributed lag analysis. As seen in Fig. 5(a), U and EOU measured after noninteractive, mockup-based training (T1) were predictive of intention measured in the same time period (T1), and these early intentions (T1), in turn, predicted usage behavior for the first month of implementation (i.e., use between T1 and T2), which, in turn, predicted use in the next two months, which, in turn, predicted use in the final three months of the study. This pattern is identical to experiment 1. The results from the hands-on user group, shown in Fig. 5(b) were consistent with the preprototype user groups from experiments 1 and 2. Including a dummy variable for treatment (preprototype versus hands-on) did not moderate any of the relationships in Experiment 2, thus suggesting the similarity of relationships among key constructs across these two training approaches.

VII. DISCUSSION

These findings are highly encouraging about the viability of capturing stable and predictive measures of user acceptance very early in the design process, even before users have had an opportunity to actually interact with a prototype software



(a)



(b)

Fig. 5. (a). Study 2: Distributed lag model (preprototype group). (b) Study 2: Distributed lag model (hands-on training group).

product. The findings revealed that behavioral intention and perceived usefulness measured before hands-on experience with a software product were highly correlated with, and not significantly different from the same measures taken after one month and three months of hands-on experience with the system. Much prior research in IT adoption has demonstrated the ability to successfully predict system success *only* after the users have acquired significant direct experience with the system. The present research presents the first efforts to systematically examine system introductions in real organizations and examine the viability of preprototype user acceptance testing longitudinally. Thus, the current research extends prior work by showing that usefulness and behavioral intention measures based on simple noninteractive mockups of a software product are predictive of those that would be obtained after hands-on use of a system as far as six months post-implementation.

Before discussing the other implications of the current work, some limitations should be noted. Using TAM as the theoretical lens has the potential for other predictors to be overlooked—especially those identified in other competing models of user acceptance. However, this limitation is alleviated to some extent since recent research on TAM has shown that many other predictors of user acceptance are, in fact, mediated by TAM constructs (see [71] and [72]). Another limitation relates to the type of expectations individuals may have formed in these two studies. Clearly, the pattern of results here indicates that the users were able to form realistic expectations about the system early in their learning/training process—this may not always be the case (see [63]). It is possible that such realistic expectations are more likely when software systems are *not* transformational of the business processes and/or jobs. Future research should examine the generalizability of these findings across a range of systems, particularly those that have major changes/implications in store for jobs and/or workflow processes.

The stability of key user reactions provided the primary ingredient for the viability of preprototype user acceptance testing. In other words, the stability of user reactions resulted in strong support for the proposed research model. Specifically, early user reactions drive short-term usage behavior, which, in turn, drives subsequent usage behavior. Given the stability of user reactions over time, later user reactions do not provide any additional explanatory power in later usage behavior, as predicted. Therefore, future research should approach cross-sectional analyses with a measure of caution-while they may a valid representation of the relationships among user reactions (i.e., perceptual constructs), they may not be predictive of key outcomes. This may, in fact, be what was observed by Straub *et al.* [59] when they found user reactions (to a system that users had significant experience with) predicted only self-reported usage behavior and did not predict actual behavior (we observe this latter pattern here when controlling for previous usage behavior).

The importance of early measures in predicting usage behavior presents exciting opportunities to study many competing designs and choose the right design plan prior to significant investment of time and money. However, the stability of early measures should caution system designers, developers, and trainers that early pitfalls can have a lasting influence on user perceptions that are very difficult to change. In any case, preprototype user acceptance testing appears to provide valuable insights into actual post-implementation user acceptance of software systems in the workplace. As project cycle times are crashing and there is an increasing emphasis on rapid turnaround times, development methodologies are starting to reflect that sentiment. The success of approaches such as extreme programming relies on our ability to accurately predict the success of specific system ideas prior to significant development effort and investment. The findings of the current research help identify the *most crucial perceptions* to measure that can have a lasting influence on system usage/success.

The findings suggest important implications for early user acceptance testing. Some previous work has presented a pessimistic view of the ability to predict user behavior based on subjective measures (e.g., [23], [28], and [57]). However, as Davis [18] points out, the ability to predict behavior based on subjective measures is highly dependent on the specific measurement instruments being used. The present work shows that expectations about a system captured using reliable and valid measures of key expectations, even before hands-on use of the system, are predictive of those that would have been obtained after brief use of a test prototype, as well as after several weeks of actual system use. Our findings show that stimuli in the form of noninteractive preprototypes can yield good approximations of user perceptions after several weeks of direct experience with the system. Given the stability of key expectations over time, the importance of the early stages of the system design and development process is further highlighted. Hence, it becomes even more critical to ensure that the early stages of design, which require less than a fourth of the total cost, be performed thoroughly since user expectations gauged at this early stage could provide valuable insights into subsequent acceptability of the software product to be developed.

Another area of future research is to examine the acceptability of software developed for general public consumption, rather than systems custom-built in organizational settings. There may be certain differences between organizational systems and commercial software products that merit attention. However, these findings even as they stand provide a very valuable development and marketing tool for system developers. Gauging user reactions using CD-ROM test drives, video mockups, graphic designs, and simple write-ups seem quite feasible in terms of providing valuable insights into the future acceptability of new systems.

Our results have other important practical implications as well. Hundreds of thousands, sometimes millions, of dollars are invested to develop a new software product. Such investments are inherently risky. Often the receptivity of the intended user population falls well below plans. With IT investment at unprecedented heights, our ability to successfully predict IT acceptance before significant investment is critical as it helps make decisions on “go-no go” decisions. Badly needed are any tools or techniques that would enable designers and managers to reduce the risk of less-than-successful software products. Using measures based on the TAM [21], the present research shows that meaningful estimates of the degree of user acceptance can be measured at a point in time before a test prototype of a new software product is even built. Based on such measurements, designers and managers can decide whether to: 1) go forward with the software product as planned; 2) modify the design to improve acceptability; or 3) abandon the design effort to avert major losses. Multiple candidate solutions to a business problem can be compared in order to focus in on the right functionality for a new software product. When managers face difficult resource allocation decisions among competing projects, preprototype testing could provide important input to the process of prioritizing the application portfolio. Given the large investments and high risks involved in new software product development, and the comparatively low cost of conducting preprototype testing, the economic implications of our findings are potentially very far-reaching.

APPENDIX
LIST OF ITEMS

A. *Intention to Use*

Assuming I had access to $\langle \mathrm{system} \rangle$, I intend to use it.

Given that I had access to $\langle \mathrm{system} \rangle$, I predict that I would use it.

B. *Perceived Usefulness*

Using $\langle \mathrm{system} \rangle$ will improve my performance on my job.

Using $\langle \mathrm{system} \rangle$ in my job will increase my productivity.

Using $\langle \mathrm{system} \rangle$ will enhance my effectiveness in my job.

I find $\langle \mathrm{system} \rangle$ would be useful in my job.

C. Perceived Ease of Use

My interaction with ⟨system⟩ will be clear and understandable.

Interacting with ⟨system⟩ will not require a lot of my mental effort.

I find ⟨system⟩ will be easy to use.

I will find it easy to get ⟨system⟩ to do what I want it to do.

Notes:

1) all items had a 7-point Likert scale;
2) the verb tenses were modified appropriately to reflect future/current interactions, depending on the point of measurement.

REFERENCES

[1] D. A. Adams, R. R. Nelson, and P. A. Todd, "Perceived usefulness, ease of use, and usage of information technology: A replication," *MIS Quart.*, vol. 16, pp. 227–250, 1992.

[2] R. Agarwal and V. Venkatesh, "Assessing a firm's web presence: A heuritic evaluation procedure for the measurement of usability," *Inform. Syst. Res.*, vol. 13, no. 2, pp. 168–186, June 2002.

[3] I. Ajzen, "The theory of planned behavior," *Org. Behavior Hum. Decision Process.*, vol. 50, pp. 179–211, 1991.

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

[5] M. Alavi, "An assessment of the prototyping approach to information systems development," *Commun. ACM*, vol. 27, pp. 556–563, 1984.

[6] S. W. Ambler, *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*. New York: Wiley, 2002.

[7] J. R. Anderson, *The Architecture of Cognition*. Cambridge, MA: Harvard Univ. Press, 1983.

[8] R. P. Bagozzi and S. K. Kimmel, "A comparison of leading theories for the prediction of goal-directed behaviors," *Br. J. Social Psych.*, vol. 34, pp. 437–461, 1995.

[9] A. Bandura, "Self-efficacy mechanism in human agency," *Amer. Psych.*, vol. 37, pp. 122–147, 1982.

[10] L. R. Beach and T. R. Mitchell, "The basics of image theory," in *Image Theory: Theoretical and Empirical Foundations*, L. R. Beach, Ed. Mahwah, NJ: Erlbaum, 1998, pp. 3–18.

[11] ——, "Image theory, the unifying perspective," in *Decision Making in the Workplace: A Unified Perspective*, L. R. Beach, Ed. Mahwah, NJ: Erlbaum, 1996, pp. 1–20.

[12] W. L. Bewley, T. L. Roberts, D. Schoit, and W. L. Verplank, "Human factors testing of xerox's 8010 'Star' office workstation," in *Proc. CHI'83 Human Factors in Computing Systems*, 1983, pp. 72–77.

[13] B. W. Boehm and P. N. Papaccio, "Understanding and controlling software costs," *IEEE Trans. Software Eng.*, vol. 14, pp. 1462–1477, Oct. 1988.

[14] W. W. Chin and P. A. Todd, "On the use, usefulness, and ease of use of structural equation modeling in MIS research: A note of caution," *MIS Quart.*, vol. 19, pp. 237–246, 1995.

[15] M. Conner and C. J. Armitage, "Extending the theory of planned behavior: A review and avenues for further research," *J. Appl. Social Psych.*, vol. 28, pp. 1429–1464, 1998.

[16] E. H. Creyer, J. R. Bettman, and J. W. Payne, "The impact of accuracy and effort feedback and goals on adaptive decision behavior," *J. Behav. Decision Making*, vol. 3, pp. 1–16, 1990.

[17] F. D. Davis, "A technology acceptance model for empirically testing new end-user information systems: Theory and results," Ph.D. dissertation, Sloan School of Management, Masschusetts Inst. Technology, Cambridge, MA, 1986.

[18] ——, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quart.*, vol. 13, pp. 319–340, 1989.

[19] ——, "User acceptance of information technology: System characteristics, user perceptions and behavioral impacts," *Int. J. Man-Machine Stud.*, vol. 38, pp. 982–1003, 1993.

[20] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw, "Extrinsic and intrinsic motivation to use computers in the workplace," *J. Appl. Social Psych.*, vol. 22, pp. 1111–1132, 1992.

[21] ——, "User acceptance of computer technology: A comparison of two theoretical models," *Manage. Sci.*, vol. 35, pp. 982–1003, 1989.

[22] F. D. Davis and V. Venkatesh, "A critical assessment of potential measurement biases in the technology acceptance model," *Int. J. Human-Comput. Stud.*, vol. 45, pp. 19–45, 1996.

[23] G. DeSanctis, "Expectancy theory as an explanation of voluntary use of a decision support system," *Psychol. Rep.*, vol. 52, pp. 247–260, 1983.

[24] A. Dix, J. Finlay, G. Abowd, and R. Beale, *Human-Computer Interaction*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[25] H. J. Einhorn and R. M. Hogarth, "Unit weighting schemes for decision making," *Org. Behav. Hum. Perform.*, vol. 13, pp. 171–192, 1978.

[26] M. Fishbein and I. Ajzen, *Belief, Attitude, Intention and Behavior: An Introduction to Theory and Research*. Reading, MA: Addison-Wesley, 1975.

[27] D. Gefen and D. W. Straub, "Gender differences in the perception and use of E-mail: An extension to the technology acceptance model," *MIS Quart.*, vol. 21, pp. 389–400, 1997.

[28] M. J. Ginzberg, "Early diagnosis of MIS implementation failure: Promising results and unanswered questions," *Manage. Sci.*, vol. 27, pp. 459–478, 1981.

[29] M. E. Gist and T. R. Mitchell, "Self-efficacy: A theoretical analysis of its determinants and malleability," *Acad. Manage. Rev.*, vol. 17, pp. 183–211, 1992.

[30] M. E. Gist, C. Schwoerer, and B. Rosen, "Effects of alternative training methods on self-efficacy and performance in computer software training," *J. Appl. Psych.*, vol. 74, pp. 884–891, 1989.

[31] M. Good, T. M. Spine, J. Whiteside, and P. George, "User-derived impact analysis as a tool for usability engineering," in *Proc. CHI'86 Human Factors in Computing Systems*, 1986, pp. 241–246.

[32] D. L. Goodhue and R. L. Thompson, "Task-technology fit and individual performance," *MIS Quart.*, vol. 19, no. 2, pp. 213–236, June 1995.

[33] J. D. Gould, S. J. Boies, and C. Lewis, "Making usable, useful, productivity-enhancing computer applications," *Commun. ACM*, vol. 34, pp. 74–85, 1991.

[34] J. D. Gould, J. Conti, and T. Hovanyecz, "Composing letters with a simulated listening typewriter," *Commun. ACM*, vol. 26, pp. 295–308, 1983.

[35] J. D. Gould and C. Lewis, "Designing for usability: Key principles and what designers think," *Commun. ACM*, vol. 28, pp. 300–311, 1985.

[36] A. R. Hendrickson, P. D. Massey, and T. P. Cronan, "On the test-retest reliability of perceived usefulness and perceived ease of use scales," *MIS Quart.*, vol. 17, pp. 227–230, 1993.

[37] M. Igbaria, N. Zinatelli, P. Cragg, and A. Cavaye, "Personal computer acceptance factors in small firms: A structural equation model," *MIS Quart.*, vol. 17, pp. 279–305, 1997.

[38] R. Johansen and E. Baker, "User needs workshops: A new approach to anticipating user needs for advanced office systems," *Office: Technol. People*, vol. 2, pp. 103–119, 1984.

[39] C. Jones, *Patterns of Software Systems Failure and Success*. London, U.K.: Thomson, 1996.

[40] E. Karahanna, D. W. Straub, and N. L. Chervany, "Information technology adoption across time: A cross-sectional comparison of pre-adoption and post-adoption beliefs," *MIS Quart.*, vol. 23, pp. 183–213, 1999.

[41] D. E. Kieras and P. G. Polson, "An approach to the formal analysis of user complexity," *Int. J. Man-Mach. Stud.*, vol. 22, pp. 365–394, 1985.

[42] M. A. Mahmood, L. Hall, and D. L. Swanberg, "Factors affecting information technology usage: A meta-analysis of the empirical literature," *J. Org. Comput. Electron. Commerce*, vol. 11, no. 2, pp. 107–130, 2001.

[43] C. C. Mann, "Why software is so bad," *Technol. Rev.*, pp. 33–38, July-Aug. 2002.

[44] M. M. Mantei and T. J. Teorey, "Cost/benefit analysis for incorporating human factors in the software lifecycle," *Commun. ACM*, vol. 31, pp. 428–439, 1988.

[45] M. L. Markus and M. Keil, "If we build it, they will come: Designing information systems that people want to use," *Sloan Manage. Rev.*, pp. 11–25, 1994.

[46] K. Mathieson, "Predicting user intentions: Comparing the technology acceptance model with the theory of planned behavior," *Inform. Syst. Res.*, vol. 2, pp. 173–191, 1991.

[47] S. McConnell, *Rapid Development: Taming Wild Software Schedules*. Redmond, WA: Microsoft Press, 1996.

[48] P. Norman and L. Smith, "The theory of planned behavior and exercise: An investigation into the role of prior behavior, behavioral intentions and attitude variability," *Eur. J. Social Psychol.*, vol. 25, pp. 403–415, 1995.

[49] L. Olfman and M. Mandviwalla, "Conceptual versus procedural software training for graphical user interface: A longitudinal field experiment," *MIS Quart.*, vol. 18, pp. 405–426, 1994.

[50] J. A. Ouellette and W. Wood, "Habit and intention in everyday life: The multiple processes by which past behavior predicts future behavior," *Psychol. Bull.*, vol. 124, pp. 54–74, 1998.

[51] M. C. Paulk, C. V. Weber, B. Curtis, and M. B. Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, MA: Addison-Wesley, 1994.
[52] S. Pfleeger, *Software Engineering: Theory and Practice*. Washington, DC: Rand, 1998.
[53] P. G. Polson, "A quantitative theory of human-computer interaction," in *Interfacing Thought*, J. M. Carroll, Ed. Cambridge, MA: MIT Press, 1987.
[54] M. B. Rosson, S. Maass, and W. A. Kellogg, "Designing for designers: An analysis of design practice in the real world," in *Proc. CHI + GI'87 Conf.*, 1987, pp. 137–141.
[55] A. H. Segars and V. Grover, "Re-examining perceived ease of use and usefulness: A confirmatory factor analysis," *MIS Quart.*, vol. 17, pp. 517–525, 1993.
[56] B. H. Sheppard, J. Hartwick, and P. R. Warshaw, "The theory of reasoned action: A meta-analysis of past research with recommendations for modifications and future research," *J. Consum. Res.*, vol. 15, pp. 325–343, 1988.
[57] A. Srinivasan, "Alternative measures of system effectiveness: Associations and implications," *MIS Quart.*, vol. 9, pp. 243–253, 1985.
[58] *Charting the Seas of Information Technology*, Standish Grp., Dennis, MA, 1994.
[59] D. W. Straub, M. Limayem, and E. Karahanna, "Measuring system usage: Implications for IS theory testing," *Manage. Sci.*, vol. 41, pp. 1328–1342, 1995.
[60] G. H. Subramanian, "A replication of perceived usefulness and perceived ease of use measurement," *Dec. Sci.*, vol. 25, pp. 863–874, 1994.
[61] B. Szajna, "Empirical evaluation of the revised technology acceptance model," *Manage. Sci.*, vol. 42, pp. 85–92, 1996.
[62] ——, "Software evaluation and choice: Predictive validation of the technology acceptance instrument," *MIS Quart.*, vol. 18, pp. 319–324, 1994.
[63] B. Szajna and R. Scamell, "The effects of information system user expectations on their performance and perceptions," *MIS Quart.*, vol. 17, pp. 493–516, 1993.
[64] S. Taylor and P. A. Todd, "Understanding information technology usage: A test of competing models," *Inform. Syst. Res.*, vol. 6, pp. 144–176, 1995a.
[65] ——, "Assessing IT usage: The role of prior experience," *MIS Quart.*, vol. 19, pp. 561–570, 1995.
[66] R. L. Thompson, C. A. Higgins, and J. M. Howell, "Personal computing: Toward a conceptual model of utilization," *MIS Quart.*, vol. 15, pp. 125–142, 1991.
[67] H. C. Triandis, *Interpersonal Behavior*. Monterey, CA: Brookes/Cole, 1977.
[68] R. R. Vallacher and J. Kaufman, "Dynamics of action identification: Volatility and structure in the mental representation of behavior," in *The Psychology of Action: Linking Cognition and Motivation to Behavior*, P. M. Gollwitzer and J. A. Bargh, Eds. New York: Guilford , 1996, pp. 260–282.
[69] R. R. Vallacher and D. M. Wegner, "What do people think they're doing? Action identification and human behavior," *Psychol. Rev.*, vol. 94, pp. 3–15, 1987.
[70] P. Van Schaik, "Involving users in the specification of functionality using scenarios and model-based evaluation," *Behav. Inform. Technol.*, vol. 18, pp. 455–466, 1999.
[71] V. Venkatesh, "Determinants of perceived ease of use: Integrating perceived behavioral control, computer anxiety and enjoyment into the technology acceptance model," *Inform. Syst. Res.*, vol. 11, pp. 342–365, 2000.
[72] V. Venkatesh and F. D. Davis, "A theoretical extension of the technology acceptance model: Four longitudinal field studies," *Manage. Sci.*, vol. 46, pp. 186–204, 2000.
[73] ——, "A model of the antecedents of perceived ease of use: Development and test," *Dec. Sci.*, vol. 27, pp. 451–481, 1996.
[74] V. Venkatesh and M. G. Morris, "Why don't men ever stop to ask for directions? Gender, social influence, and their role in technology acceptance and usage behavior," *MIS Quart.*, vol. 24, pp. 115–139, 2000.
[75] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User acceptance of information technology: Toward a unified view," *MIS Quart.*, vol. 27, 2003.
[76] J. C. Wetherbe and N. Vitalari, *Systems Analysis and Design Best Practices*, Fourth ed. St. Paul/Minneapolis, MN: West Publishing, 1994.

Fred D. Davis received the Ph.D. degree from the Sloan School of Management, Massachusetts Institute of Technology, Cambridge.

He is currently the David D. Glass Endowed Chair Professor in Information Systems and the Chair of the Information Systems Department, the Sam M. Walton College of Business, University of Arkansas, Fayetteville. He has served on the Business School Faculties at University of Michigan, Ann Arbor, University of Minnesota, Twin Cities, and University of Maryland, College Park. He has taught a wide range of information technology (IT) courses at the undergraduate, M.B.A., Ph.D., and executive levels. He has served as Associate Editor for the scholarly journals *Management Science, MIS Quarterly*, and *Information Systems Research*. He has published extensively on the subject of user acceptance of IT and IT-assisted decision making. His research has appeared in such journals as *MIS Quarterly, Information Systems Research, Management Science, Journal of Experimental Social Psychology, Decision Sciences, Organizational Behavior and Human Decision Processes, Communications of the AIS*, and *Journal of MIS*. His current research interests include IT training and skill acquisition, management of emerging IT, and the dynamics of organizational change.

Viswanath Venkatesh received the Ph.D. degree from the Carlson School of Management, the University of Minnesota, Twin Cities.

He is currently a Tyser Fellow, Associate Professor, and Director of MBA Consulting with the Robert H. Smith School of Business, University of Maryland, College Park, where he has been on the faculty since 1997. His research has been published in many leading journals including *MIS Quarterly, Information Systems Research, Management Science, Communications of the ACM, Decision Sciences, Organizational Behavior and Human Decision Processes, Personnel Psychology*. He is currently an Associate Editor at *MIS Quarterly* and *Information Systems Research*. His research interests include implementation and use of technologies in organizations and homes.

Dr. Venkatesh received *MIS Quarterly's* "Reviewer of the Year" Award in 1999.