

MAKE IT INTUITIVE: AN EVALUATION PRACTICE EMERGENT FROM THE PLANS AND SCRIPTED BEHAVIOR OF THE COMPUTER COMMUNITY OF PRACTICE

Pat Lehane

*Defence and Industrial Systems
University of South Australia
Australia*

Abstract: *The catch phrase today for system designers is to “make it intuitive,” which begs the question, what is intuitive? The action research discussed in this article was the final stage of the application of grounded theory to user data that provided survey categories (criteria) for system acceptance. A theoretical rationale from the discipline of human–computer interaction was proposed to provide a consistent and repeatable interpretation of the users’ responses to the survey and directly align the responses to software design considerations. To put this work into context, I discuss in this article a case study on the use of the survey to monitor the user experience during the upgrade of an enterprise system and the subsequent implications and outcomes of applying the theoretical paradigm in practice. As such it may provide food for thought on survey design for elicitation of user requirements for information and communication technology systems.*

Keywords: *Survey, interaction design, community of practice, user experience, intuitive.*

INTRODUCTION

In this article I discuss a survey for assessing system acceptance and the user experience (UX) from the pragmatic perspective of a human–computer interaction (HCI) practitioner. There is a plethora of HCI-related surveys and, more often than not, the analysis of the participants’ responses to a survey results in a numerical index or a summarized qualitative description. It can be difficult to relate this type of analysis result back to the users’ perceptions of the system, or derive any useful design and development concepts or clear direction on how to proceed in response to the solicited user feedback. The survey discussed in this paper has 12 criteria that emerged from a grounded theory analysis (Dick, 2005; Glaser, 1994, 1998) of notes taken during interviews with users. The goal of the analysis was to make sense of the users’ responses to legacy surveys that had no assessment rationale.

The survey is the final stage of a grounded theory, data-driven investigation into the survey question grouping classifications that are the emergent theoretical constructs. An HCI theoretical

rationale (Lehane, 2010, 2012, in press) was developed to explain the emergence of these criteria, and it is used to interpret the users' responses to the survey in terms of HCI design concepts. Central to the theoretical rationale and the subsequent interpretation is the concept of leverage of prior knowledge, which is expressed as previously learned or scripted behavior patterns. The survey was used as an adjunct to industry-standard project management practice during the regular upgrade of an enterprise system. Its use to monitor and recommend interventions in the roll-out of the system was an action research project that was the basis for a doctoral study. The doctoral study was conducted independent from but in conjunction with the management practices of the system upgrade. The purpose of the survey was to provide a general indication of the users' system familiarity, based on their current usage and the leverage of prior knowledge and experience.

Activity theory has a three-level abstraction hierarchy to contextually describe human activity. Activities are long-term formulations with an objective that typically requires several actions or chains of actions to be achieved. Actions can be operationalized by habituation, wherein the action is a scripted and skilled behavior requiring minimal cognitive resources. Actions then become a series of operations (Bødker, 1991; Nardi, 1996a; Suchman, 1987; Vygotsky, 1978). Operations can be undertaken as conscious actions by conceptualization of the skilled behavior. This concept of levels can be applied recursively in a drill-down analysis of an activity. The example in Table 1 (Kuutti, 1996, p. 33) shows the hierarchical structure of an activity and its components.

Operations are the level at which intuitive use of the screen artifacts occurs. Thus, the concept of intuitive computer use by a community of practice (Vygotsky, 1978) provides the rationale for the interpretation of the responses to the survey (Lehane, 2010). When a survey criterion receives a low appraisal, the means to resolve that issue has already been identified. This is due to the concept that the interaction design is based on scripted behavior (Bødker, 1991; Suchman, 1987), which is viewed as the key concept for consideration of intuitive design and the subsequent resolution of arising use issues. When the survey responses are viewed as a time series comprising a benchmark, the transition(s), and familiar use, UX is presented in terms of HCI design concepts. The strength of the survey is that the time series of responses can show changes in high-level user perceptions; after a shortfall has been identified, the associated remedial program then can be implemented.

The enterprise systems discussed here were implemented with little or no customization: The functionality of the software was not modified to mirror the organization's existing business processes. In such situations, usability engineering (Nielsen, 1993) concepts are not applied at the activity level as a learned behavior; they are applied at the action level as the conscious operationalizations of the scripts, which establish the users' expectations for interacting with the system. This subtle variation, observed in the user interaction with large enterprise systems with no or limited customizations, is included in this investigation.

Table 1. The Structure of an Activity.

Term	Product of	Example
Activity	Motive	communicate with a friend
Action	Goal	to send an e-mail
Operation	Condition	produce the e-mail by using a computer's mouse and keyboard

The Context of the Upgrade Projects

In this article, *human–system integration* is used to bring together the workforce and all the other systems, including the information and communication technology (ICT) systems, that constitute the workplace, with the end result being a workplace with effective and efficient processes and procedures. *Enabling technology* is the term used to encompass this ensemble of software applications, supporting ICTs, people, cultures, and task-associated processes and procedures. To produce usable ICT systems, analysis and design now seek to employ the broader human–system integration perspective of enabling technology. In addition, the UX described here relates to the use of large enterprise systems with little or no customization. There is a sound business rationale for implementing systems with little or no customization: the very high cost of ongoing customization in subsequent version/upgrade releases. Because of this constraint, it is likely this practice will be the norm rather than the exception in the future. As a consequence, niche applications are often interfaced with the enterprise system as a specialized functionality to meet a required business capability, for example, e-mail.

Conceptually, system designers talk of usability and usability engineering concepts in response to a learned behavior (scripts), which is at the activity (process) level of an interaction. Scripts are conscious decisions based on the long-term memory of a familiar sequence of actions required to complete a process. Where this article differs from other research is that it considers usability and usability engineering concepts not only from the familiarity perspective, but also from the perspective of being introduced to new or unfamiliar software and learning how to use it.

In such situations, then, usability engineering concepts are not applied at the activity level as a reflection of a learned behavior. Rather, they are applied at the action level as the conscious operationalization of the scripts that establish the user's expectations for interacting with the system. At the operation level of an activity, behaviors are the unconscious operator executions that accompany skilled use of computer screen artifacts. This reflects the subtle differences between the user's and the developer's interpretation of the terminology. The developer thinks in terms of familiar use, while the UX is with new and unfamiliar software.

Similarly, the HCI professional perspective is to define a business capability as an activity, and then develop functionality with good usability to facilitate that requirement. Utility is a valued assessment of the functionality, and usefulness the fit-for-purpose assessment of that functionality. This perspective is readily applied to Web development or screens that are custom designed to support behavior that successfully fulfills a motive. However, when dealing with large enterprise systems with no or limited customizations, such as the project presented in this paper, the case studies present a subtle variation, which is the user perspective as interpreted by this author.

In such systems, a screen can be used to accomplish a number of functions, each of which requires the observable behavior of various cognitive activities. A single screen is not designed to accomplish one task as a business process. Rather, it presents a number of artifacts that may or may not relate to any number of the organization's business processes, as understood by the individual user. In other words, a number of people could access the same screen and, using different menus and menu items, undertake completely different business processes. Consequently, to meet the design concept of functionality, the user has to develop the interaction design for each of their organization's business processes.

For the production of the required work output, the user has to assemble the artifacts into an effective and efficient operational sequence, based on the usability of the artifacts' affordances. The derived sequence of actions across a number of screens then defines the activity as a business process. This activity then represents the functionality of the system that supports the process. Finally, the work is assessed as either satisfactory or not; a satisfactory assessment confirms that the system is useful in supporting the user.

Conceptually, usability for the user traditionally has focused on the individual screen artifacts and their affordances as usability indicators with efficiencies of use. The individual artifact interactions were sequenced by the affordances of the successive artifacts so as to build up an interaction design that provided the required work outcome as a process, which was then considered the functionality. For the user, artifact usability and utility came first, with the functionality based on the emergent interaction design.

The screen artifacts and their affordances shaped the learning activities, which later were integrated and sequenced into a process. Vygotsky's (1978) zone of proximal development best characterizes this learning situation, and Nielsen's (1993) usability, both the term and the criteria, best accommodate the users' descriptions of these interactions. Additionally, from Nielsen, utility is a value assessment and those values are the functions attributed to usability (such as easy to learn, use, remember, and recover from errors). Consequently, utility is a value assessment of the usability characteristics (affordances) of the screen artifacts.

In addition, the user interpretation of the HCI term *functionality* was best expressed in human factors terminology and concepts. The users described functionality as the computer software functions programmed into the software to support them in their work. By definition from human factors (Wilson & Corlett, 1995), function allocation is the division of labor between humans and machines. Humans are assigned tasks and machines allocated functions. For the users, functionality was the set of functions programmed into the software and subsequently accessed as a process by the users in the course of doing their work. Therefore, for the users, functionality did not necessarily represent a specific business capability or work outcome, as it did for the developers.

From an interaction perspective, the users' descriptions of functionality, as a set of functions programmed into the software, were a utility assessment, a valued judgment on the characteristics of the usability of the artifacts (i.e., easy to learn, use, and remember). However, when the users described their work using the software, their descriptions aligned with the HCI concept of functionality, which is associated with the design perspective of the usability of the technology at the process level. Thus, it was an assessment of how well the software supported, as opposed to hindered, them in their work—an assessment on the software being fit-for-purpose. Whenever the system helped and the software functions supported them well, the users' descriptions aligned with the usability engineering concept of usefulness.

Consequently, based on the users' descriptions of their use experience, this project's grounded theory-derived abstraction hierarchy for usability engineering criteria was system acceptance, usability, utility, functionality, and usefulness. System acceptance is the least tangible concept and, consequently, the most abstract to assess; usability was the most physical and, thus, the least abstract concept. The survey questions regarding utility and usability reflected the physical function level of the system's abstraction hierarchy. These questions were used to confirm the users' responses to questions on the more abstract criteria of functionality and usefulness.

The theoretical concepts for system acceptance that aligned with the emergent construct were drawn from usability engineering (Nielsen, 1993) and activity theory (Hasan, Gould, Larkin, & Vrazalic, 2001; Leontiev, cited in Nardi, 1996b; Vygotsky, 1978). The activity theory elements, collectively, were called the use-community concepts. The theoretical rationale for the operationalization of activity theory (Lehane, in press) is summarized in Table 2.

The use-community criteria were grouped according to the use-community practice and the context of that practice. The use-community supporting use-practice criteria relate to the users' interactions with the ICT system, based on expertise gained from experiences using other systems, that is, the praxis of the use-community. Ordered from the physical form to the abstract, the use-community supporting use-practice criteria are the tools, distribution of community praxis, and the acquisition of praxis. Tools are the physical objects. The distribution of praxis involves the physical objects and their dissemination across people, time, and location. The acquisition of praxis is the high level internalization of concepts acquired through observation and physical activity in using the tangible objects.

The use-community ecological criteria are considered integration issues between the enabling technology and the other systems in the workplace. These other workplace systems contextually influence the affective, behavioral, and reflective responses of the persons using the software. The use-community ecological criteria, from the physical objects to most abstract, are hardware, human factors, and support for work-in-context. The hardware constitutes the physical object(s). Human factors revolve around the quality of the interactions with the physical objects. Support work-in-context is how well the outcomes produced by those interactions with the physical objects comply with the motive for the situated activities.

Use-community criteria are concepts fundamental to activity theory and relate to the users' whole-of-life computer experiences. As such, they provide information on how much of that experience is leveraged by the interaction design in making the system intuitive.

Table 2. Survey Criteria.

Criteria	Key Concepts
Usability Engineering Criteria	User Acceptability <ul style="list-style-type: none"> ▪ System Acceptance ▪ Usefulness ▪ Functionality ▪ Utility ▪ Usability
Use-Community Criteria	Support Use practice <ul style="list-style-type: none"> ▪ Change management and system upgrade issues (tools) ▪ Familiarization with the use-community ▪ Familiarization with the use-practice (specific aspects) ▪ Training system documentation (explicit practice) ▪ Training in the technology-situated work (tacit practice) Ecological Criteria <ul style="list-style-type: none"> ▪ Work in context: using application as one duty in many ▪ Hardware: the physical objects of the system (tools) ▪ Human factors: ergonomics, workplace health and safety, and emotional perceptions

Cognitive HCI provides case studies of these experiences but in a nonstructured way, which precludes analysis, design, and evaluation of the integrated system. I propose (Lehane, in press) that activity theory, by subsuming the appropriated HCI paradigms, theoretically provides an overarching theoretical rationale and, consequently, structures contextually those paradigms of cognitive HCI. I believe, as indicated within that paper, that subsuming cognitive HCI into activity theory is central to understanding the UX.

In the remainder of this article I discuss the survey, the rationale for using it, and the theoretical considerations this encompasses. To contextualize the use of the survey, the next section introduces the survey and its use in a case study. This is followed by an overview of the development of the survey as the final step in an action research project that utilized grounded theory to seek and confirm emergent theoretical constructs in the data.

The concept of scripted behavior has been well documented in the discipline of HCI. Suchman (1987) held that every course of action was dependent on its material and social circumstances. Scripts were used in the discussion of how experienced personnel, such as fire fighters, plant operators, and air controllers, analyze and respond to known and, in particular, unfamiliar situations (Jones, 1995; Kontogiannais, 1996; Pawlak & Vincente, 1996). The premise of scripts was an emergent concept from the research data and was fundamental to the development of the theoretical rationale used in the discussion to explain the observed user behavior as intuitive.

From this perspective, one of the objectives of analysis is to seek out as many as possible of the already established scripted behaviors required by the new system. An objective of design, then, is to re-establish on the computer screen the context that triggers the scripted behaviors and produces the expected outcomes based on the users' previous experience. On this basis, I call *intuitive use* successful user interaction with the new system by means of screens designed on the premise of prior knowledge and experience with the old system.

The Survey and Graphs

Prior to the case study, it is necessary to introduce the basic concepts behind the survey and the presentation of the users' responses in the graphs. The System Acceptance Indicator (SAI; Lehane & Huf, 2005, 2006) survey contained 25 questions about the positive and negative aspects of system use. Each question was assigned a value from 0 to 4. In odd-numbered questions, 4 represented *strongly agree* with a positive aspect of use, whereas 4 represented *strongly agree* with a negative aspect of use in even-numbered questions. Even-number questions require adjustment so that a score of 100, a perfect score, represents strongly agree with odd-numbered questions and strongly disagree with even-numbered questions (see Table 3). The global index for one survey is the summation of the values assigned to the response to each question; the SAI global index for a survey campaign is the average of the individual indices. This is similar to the way that the system usability scale (Brooke, 1986) works: The questions are grouped and responses averaged for graphs during the analysis.

In the case of the survey being computer generated with a numeric value consistently assigned to the Likert Scale from *strongly disagree* to *strongly agree*, the example user gave all questions on positive use a score of 4 and all questions on negative use a score of 0. After adjustment, as depicted in Table 3, the user rated the system highly and awarded a perfect score of 100 (25 x 4).

Table 3. Example Adjustment to an Individual Survey Response.

Question	Aspect of system use	User response	Adjustment of response for graph (in a spreadsheet)	Adjusted value used for calculations
1	Positive	4	Odd number no adjustment to user response	4
2	Negative	0	4 –user response	4
3	Positive	4	Odd number no adjustment to user response	4
4	Negative	0	4 –user response	4
5	Positive	4	Odd number no adjustment to user response	4
...
25	Positive	4	Odd number no adjustment to user response	4

SAI = 100

The SAI provides three measures. The first element is a global index as a number between 0 and 100. Fifty is the value of the global index indicating a neutral disposition towards the system. Zero indicates a system that is perceived unfavorably for all questions and 100 is the score for a system that received the maximum of favorable responses.

The second measure is the graph for the data determined by the technology acceptance model (TAM; Lehane, 2012; Lehane & Huf, 2005, 2006), which is a 12-element presentation of the users' perceptions of the system. An example of this graph follows in Figure 1. The criteria for this graph are expressed in analytical terms for technical consideration of the results by the system developers.

The five criteria from usability engineering describe immediate use:

- *System acceptance* is how well the users relate positively to the system.
- *Usefulness* is how well the overall system supports users in achieving their objective(s).
- *Functionality* is how well the system's functions support the designed activities.
- *Utility* is how efficient the system is in facilitating the actions.
- *Usability* is how effectively the actions can be operationalized.

The seven concepts identified as the use-community criteria compare the use of system against previous use knowledge and experience:

- *Support for work-in-context* (Support_WIC) is how well the system integrates into the extant workplace systems.
- *Active user* is the level of proactive interaction initiated by the user.
- *Distributed cognition* is how well the praxis of the domain's community of practice was transferred to the software (i.e., does it have a familiar look and feel?).
- *Affordance* is how well the context of that praxis was embedded in the artifacts (i.e., was use intuitive?).

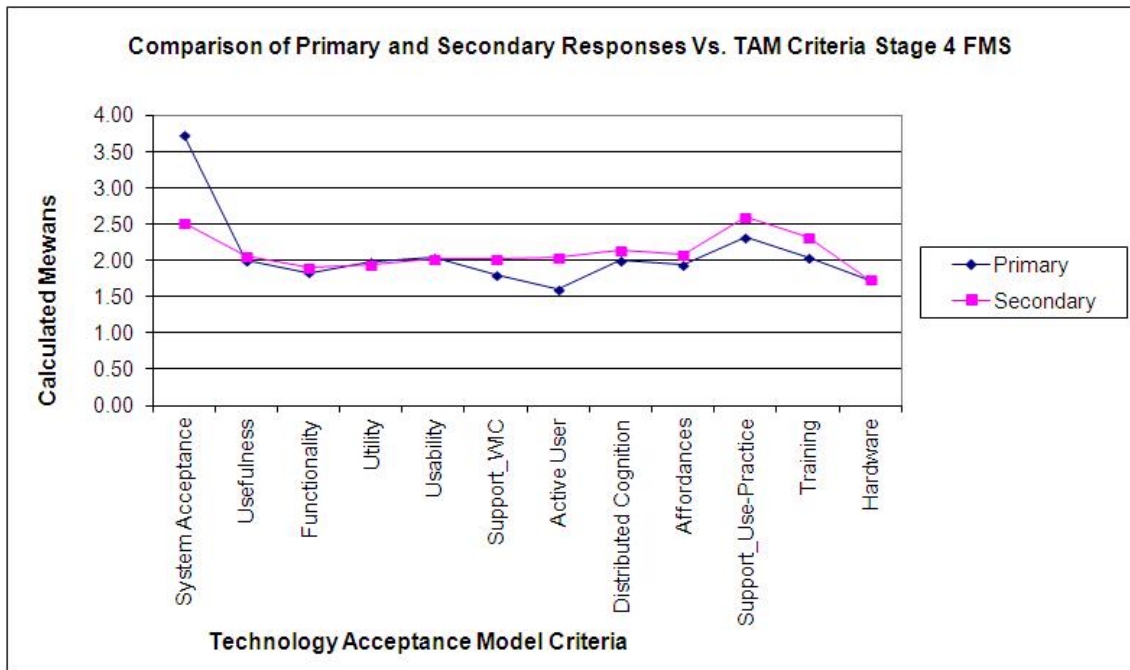


Figure 1. A graph of the technology acceptance model.

- *Support for use-practice* is the level of immersion of the user into the community of practice (e.g., an accounting background for a finance officer ensures comprehensive contextual knowledge).
- *Training* is the formal training and its cognitive and behavioral artifacts used to transfer use-practice from experts to novices.
- *Hardware* is concerned with issues related to the situated technology (i.e., computers and network).

The third measure is the SAI graph, wherein the survey responses are regrouped to a 10-element graphic presentation of the users' experience in nontechnical terms. The SAI graph is used as the basis for discussions with the business users of the system being surveyed. Again, an example from the case study of the Financial Management System (FMS) follows (see Figure 2). The TAM technical criteria of active user and distributed cognition are grouped in the SAI graph as EZ2Learn, while affordances and support for use-practice are combined as EZ2Use. EZ2Learn is an indication of the active user's ability to leverage prior knowledge through the use of distributed cognition. EZ2Use is an indication of the affordances and use-community praxis facilitating recall and operationalization of activities. Conceptually these two categories are associated with and provide an indication of the "look and feel" and how intuitive the software is to use.

The individual survey responses, after adjustment, are collated to compile the collective response to the UX. The survey ratings scale of 0 to 4 now covers the range of the collective response from strongly negative to strongly positive. The guide for interpreting the scale is

- 0 – total rejection
- 1 – poor response < 1.5 indicates a criterion to be looked at

- 2 – normal expectation, no significant influence
- 3 – good response > 2.5 indicates a criterion that was well received
- 4 – full acceptance.

The SAI was designed to provide a global indication of user satisfaction and identify the users' rationales for reaching that decision.

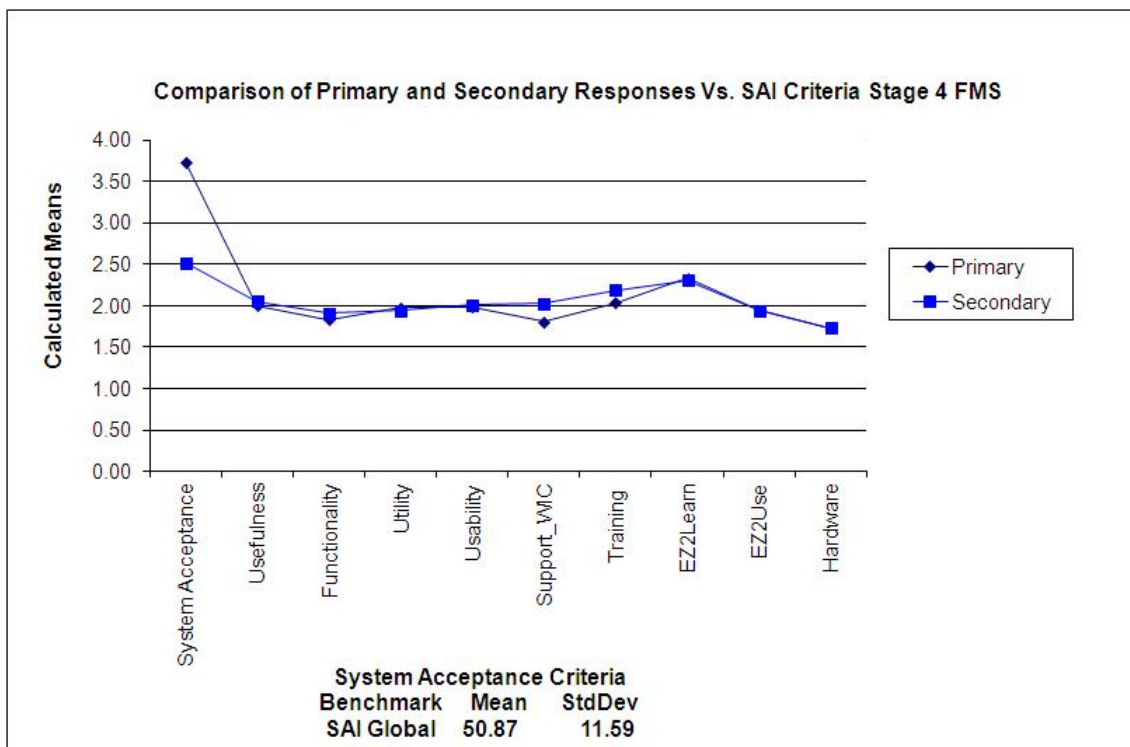


Figure 2. A graph of the System Acceptance Indicator.

A CASE STUDY

The case study was undertaken at a regional Australian university. Regional universities have their principal campus located outside the metropolitan areas of the states' capital cities. The upgrade of the enterprise system was a normal business requirement to keep the operating system current with vendor support. As an enterprise system, it is installed in many organizational units and is considered mission critical to the operations of the university.

Two focus groups were established for the research project: a managers' focus group and a day-to-day users' focus group (Bauer & Gaskell, 2000; Wilson & Corlett, 1995). Twelve managers, one from each of the university's faculties and other operational divisions, were invited to attend the meetings; all attended. One day-to-day operational user from each faculty and operational division was selected from the staff who volunteered to be on the focus group. The selection of members for this focus group was based on their experience (at least one year) and ability to articulate their experiences. Both focus groups reflected the university's employment policies; age groups and both genders were appropriately represented.

The research project was an action research project, and integral to action research are interventions. Interventions are recommendations that will improve some aspect of the system presented to and implemented by the owners of the system being investigated by the action researcher. Therefore, two stakeholder groups also were involved in the case study: the system upgrade project management board and the system users' management committee. Meetings were held with each stakeholder group separately at each stage of the research project. The purpose of these meetings was to present user issues as reported by the focus groups, the results of the SAI survey (which is fully described in Lehane, 2010), and recommendations to resolve arising problems. In this case study, the intervention of interest is reported in Stage 3 of the system upgrade.

As was stated in the Introduction, the SAI survey arose from previous work that identified the 12 criteria used to categorize the survey questions. On this basis, the survey was designed to assess the data-derived emergent system acceptance criteria in the users' terms of reference. The purpose of a survey is to elicit the users' assessments of a particular software application in meeting their use requirements within the system of systems that constitute the workplace. Each instance of use identified by a survey allows evaluation of the users' subjective assessments of the system as being fit for a purpose and the users' rationales for making that judgment. Over time and a number of survey campaigns, a picture emerges that presents, in HCI terms, the UX from first contact to habituated familiarity. The theoretical rationale for the interpretation of the participants' response to the survey and the emergent construct of HCI concepts used to structure the survey criteria are explained in Lehane (in press).

The assessment in this case study was conducted in four stages over a 14-month period, beginning in 2006. The first stage, prior to the introduction of the new system, benchmarked the existing version. The second stage, an appraisal after the transition period, was conducted as nearly as practical 2 weeks after the roll-out of the new system. The third stage, at the end of the consolidation period, was undertaken about 3 months after the roll-out. The final stage was a long-term assessment, undertaken after 12 months of use in the workplace.

The benchmark set the status quo for the system and was used as the reference against which the new version was compared. The second appraisal was used to evaluate the UX upon the release of the upgraded software. It assessed the change-management practices within the upgrade project, as well as evaluated the introduction of the actual technology and the extent to which this introduction was supported. Typically in these contexts, support leading to familiarization is a communal and collaborative activity, enhanced when an individual's learning experience is assisted by expert intervention. In activity theory, which draws on cultural-historical psychology, this transition period is called the zone of proximal development (Vygotsky, 1978, p. 86) and is identified as a critical element in securing a positive UX while acquiring knowledge or a practiced skill.

The third stage focused on the period of consolidation of use-practice, the period during which actions were operationalized. The literature review indicated that 3 months are required for an unskilled action to become familiar and to be acquired as a new skill (Venkatesh, Morris, Davis, & David, 2003). Unskilled action requires conscious thought to complete and reproduce whereas the skilled action does not require conscious thought (Nardi, 1996a; Preece, Rogers, & Sharp, 2007). The third stage, therefore, was to evaluate the UX at this point in time and to confirm or refute 3 months as the period of time required for the operationalization of actions.

Businesses inherently need operational cycles based on the week, the month, and the year. Long-term evaluation, the fourth stage, is undertaken after the participants are exposed to the full annual cycle of business activities. From a methodological perspective, this ensures that the final UX assessment is based on the same use exposure as the benchmark. The long-term evaluation of the system upgrade in this study was used also to determine if the 3-month appraisal is an accurate assessment of the re-established norms and that those norms withstood the rigors of a year's use experience.

The financial management system (FMS) presented in this case study was a mandatory-use application implemented in the late 1990s. Consequently, it had the dated look and feel of an early- to mid-1990s application. Hearsay noted the system as one the users loved to hate. The new implementation was Web-based with a modern and more familiar look and feel, which meant the user interactions with the application would be completely new after the upgrade. A précis incorporating an interpretation of the results is presented in this subsection.

The Survey Numerical Index

Considering the global indicator of the SAI over the four stages of the study, the ANOVA $F(3, 137) = 0.1224$, $p = 0.9467$ did not identify a statistical difference between the stages of the study. Table 4 provides indication of the minor variations in the SAI global indices. The response dispersion was less for Stages 1 and 4 than for Stages 2 and 3, which is indicative of a greater consensus among the participants after periods of extended use. I had anticipated a significant movement of the indicator, commensurate with changes in the user perspective. However, because of the limited increase in the dispersion of the user responses for the intermediate stages, the index, as an indicator, was considered unsatisfactory. An investigation into this unsatisfactory outcome has not been initiated at this time. However, observable variation in the graphed responses could be used as an indication of changes in the user perception.

An Interpretation of the Technology Acceptance Model

Stage 1: Establishing the Baseline

Figure 3 provides the TAM evaluation of the FMS for all four stages. A user evaluation of 2.0 is considered the response norm; an assessment of 1.5 or lower warrants further investigation; and an assessment greater than 2.5 is highly commended. The interpretation of the benchmarking data is that the original system was perceived to be useful with the necessary functionality to support

Table 4. SAI Global Evaluation for Case Study.

SAI Indicator	Mean	Standard Deviation	Study Stage	Test Date	Tally
Stage 1	50.03	11.19	Benchmark	September 2006	33
Stage 2	49.32	13.06	Transition	January 2007	31
Stage 3	49.26	13.35	Consolidation	May 2007	47
Stage 4	50.87	11.59	Long-term use	January 2008	30

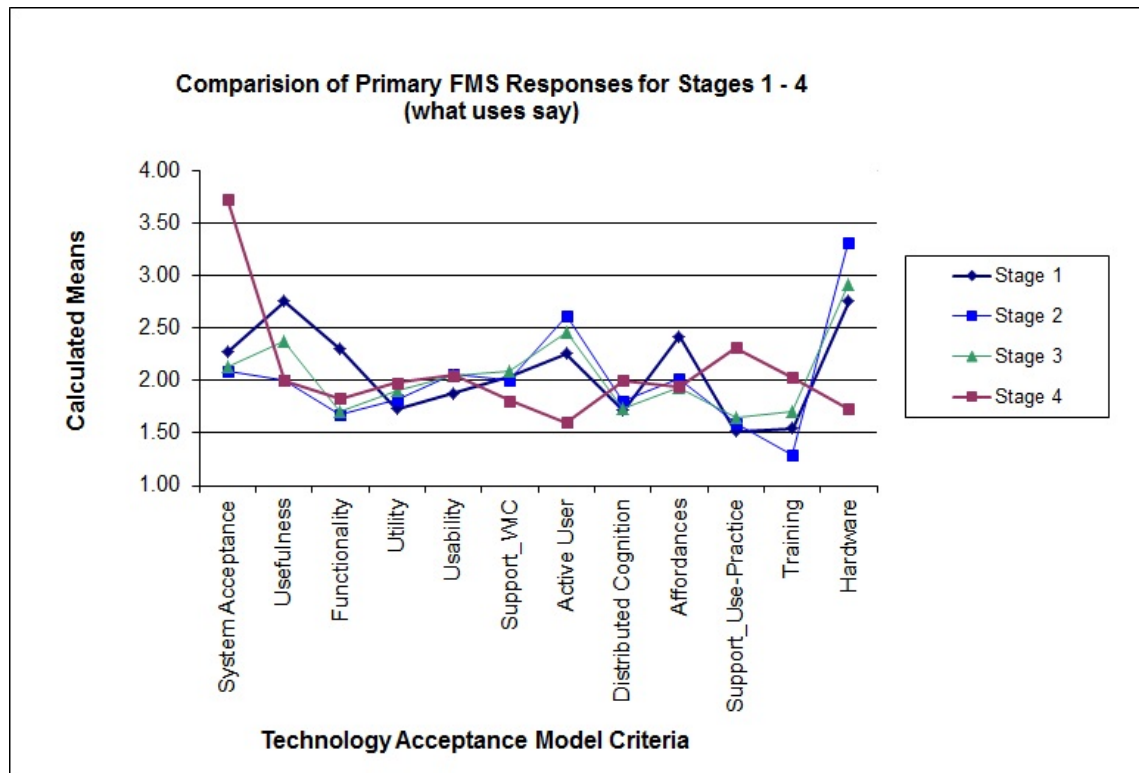


Figure 3. The case study TAM traces for Stages 1 to 4.

the users in their work. The focus groups confirmed that the system as such was accepted by the users. It was not as efficient as the users would have liked, and the issues around utility were investigated in the upgrade review. The utility issues were related to inefficiencies in the interaction design of the business processes and reported in detail in the case study. However, the users reported that they were comfortable with the usability and the integration of the FMS into their work responsibilities.

The assessment of the use perceptions of the active users indicated that the survey participants considered themselves proactively engaged in their self-directed learning to use the FMS. Similarly the focus groups confirmed that affordance-related responses were interpreted as being indicative of the user familiarity with the screen artifacts and layout and how to use those artifacts. However, the low user evaluations regarding the upgrade's use-practice support and training caused concern. The focus groups directed the investigation toward the support of use-practice, which revealed that very few users of the financial system had an accounting or financial background.

From the perspective of the theoretical rationale for the project, the workers were not members of the specific community of practice in which they were employed. They lacked the background in financial knowledge, experience, and skills directly applicable to this system's use. Training also had been poorly perceived. The focus groups reported that the users did not believe that there was a centralized or faculty-sponsored training strategy. They taught themselves by trial and error, an activity that helped to explain the high response to the active user criteria. Finally, the focus groups provided supporting feedback that the users were positively disposed towards the hardware because the application ran well on their machines and over the network.

Stage 2: Transition to the Upgraded System

The focus group discussions related to Stage 2 brought out a number of negative assessments on the upgraded system and the change management associated with the upgrade process itself. From the users' perspectives, the upgrade was not supported by a comprehensive training program. The training provided involved a seminar presentation to introduce the users to the Web interface and the new functionality. The users were not provided with hands-on training or even a step-by-step guide for the work processes affected by the upgrade. The financial reports provision, already an issue identified prior to the upgrade, was not adequate and, consequently, the upgrade did not meet the business requirement to monitor and control budgets. These deficiencies were reflected in the usability engineering criteria of the SAI.

Referring again to the survey results in Figure 3, this user perception of the loss of previously available functionality was interpreted as the reason for the users' poor evaluation of usefulness and functionality during the transition period. Unfamiliarity with the software, exacerbated by the absence of comprehensive training, was discussed extensively within the focus groups. The decrease in the users' evaluation of affordances and training was interpreted as a response to this user perception. In contrast, the survey participants estimated that their active user investigative interaction increased with the new software. Again the focus groups provided the basis for the interpretation presented to the stakeholders regarding training: the inadequate formal training and the proactive self-learning. The users' assessments of the remaining criteria remained at their benchmark levels, thus maintaining the status quo.

Stage 3: Consolidating Use Experience

After 3 months of use, the survey participants' assessments of the usability engineering criteria remained unchanged from the Stage 2 levels, except for Usefulness, which was assessed higher. The explanation for this increase, supported by the focus groups' comments, was the use familiarity gained as the users' learned about the available functionality and developed work-arounds. The focus groups also discussed how staff in their work areas independently sought to initiate some formal training that would help them learn more about the available functionality and how to use it. This discussion supported a trend that had been noted in the TAM graph of the SAI data: Support_Use-Practice was consistently given a low evaluation. This criterion assesses the survey participants' immersion in a work practice, in this case, the users' knowledge of accounting and finance.

The focus groups confirmed that very few of the FMS users had a background in accounting or finance. As a result of these data interpreted within the activity theory precepts of induction into a community of practice, a research project intervention was recommended at Stage 3. It was proposed to the system users' management committee that staff from the finance department mentor the FMS users who did not have a financial background. The objective was to train them exclusively in the use of the FMS for their particular work requirements, based on the scientific management approach (Taylor, 1911), which recognizes the benefit of expert tuition in moving people through Vygotsky's (1978) zone of proximal development. This recommendation was accepted and implemented.

Stage 4: Long-term Use

The notable features of the Stage 4 trace in Figure 3 are twofold: the users' apparent positive response toward System Acceptance, Support of Use-Practice, and Training, and the decline in the Active User. During the extended period for system evaluation involved in Stage 4, the Functionality, Utility, and Usability of the system were not enhanced. While the system did not change between Stage 3 and Stage 4, the use of it did improve, as a result of the contextually focused training implemented at the end of Stage 3.

The rise in the survey participants' positive perceptions of training was confirmed by the focus groups, an indication that the training program provided the background knowledge and the use-practice necessary to complete the allocated work. Specifically, the focus groups confirmed that the improved response to the criterion Support_Use-Practice resulted from the training: the knowledge delivered to better understand the use-practice from a financial perspective. This assessment was supported by the fall in the Active User measure, indicating that the training regime was successful and the users were no longer individually proactive in trying to learn how to use the system.

Functionality was not as well perceived as the other usability engineering criteria. This had been confirmed by the members of the focus groups, who also substantiated the premise that work-arounds were established for the missing or inadequate functionality. Usefulness was not as well perceived at the end of Stage 4 as it was for the benchmark or for Stage 3. Comments made by the focus groups indicate that this perception was due to the pragmatic perspective acquired by the users as a result of their use-focused training and subsequent use of the system during Stage 4.

At the end of Stage 4, the survey participants' positively appraised the criterion System Acceptance. The improved user perspective of this high level usability engineering concept was attributed to the improved knowledge and expertise that the contextually-situated training provided and the frequently stated opinion that the upgraded system was preferred to the system it replaced, thus an implied sense of ownership.

The survey participants' assessments of the criterion Support_WIC was relatively stable for all four stages. There was a slight drop in Stage 4 that could be attributed to the mentoring and undertaking the ancillary work of documenting the work processes and training. The users' assessments of Active User fell with the use-practice-focused training. Criteria that indicate the intuitiveness of the system design—Distributed Cognition and Affordances, which help to make a system easy to learn and use—varied over the period of the investigation. Distributed Cognition, the familiarity with the context of the screen artifacts, rose marginally in Stage 4 with the training. Affordances, the presentation of artifact use options, remained below the Stage 1 assessment but at the normal expectation level.

This situation can be explained. The screens of the upgraded FMS presented familiar artifacts in familiar locations, but it was left to the users to establish a sequence of the screens, menus, and menu items that resulted in the required work output. In effect, they needed to produce the interaction design for each process from the myriad of options presented by the menus on the various screens available. Continued use established the process but, without careful attention to detail, errors would occur and the wrong menu or menu item would be selected. This was because a number of familiar screen artifacts looked

similar but would produce different results from that required for an individual process, hence the lower assessment of Affordances.

An Interpretation of the SAI Results

Figure 4 is the SAI graph for the study of the FMS. This graph was used to discuss the survey results with the system users' management committee. Obviously, the assessment is similar to that of the TAM. To recap, initially the survey participants perceived training poorly and the focus groups confirmed this. In response to this situation, the system users' management committee initiated a mentored training program following Stage 3 to improve the training content and delivery. By the end of Stage 4, the poor perception of training was no longer apparent in the response traces.

After the mentoring, the survey participants perceived the system to be easy use and marginally easier to learn, an expected outcome in response to the more familiar Web-based look and feel and the improved training. The move to a Web look and feel changed the layout of the screens to ones that the users encountered more frequently in their day-to-day interactions with other software in the office and at home. The training introduced the users to financial terminology and practice that they could use to better interpret the meaning of the words in the menus. Together these changes made the learning and use of the FMS more intuitive because the users did not have to learn terminology and practices as they attempted to learn how to use the system. They were able to recognize and use the screen artifacts rather than engage in investigative learning to identify the use and purpose of the screen artifacts.

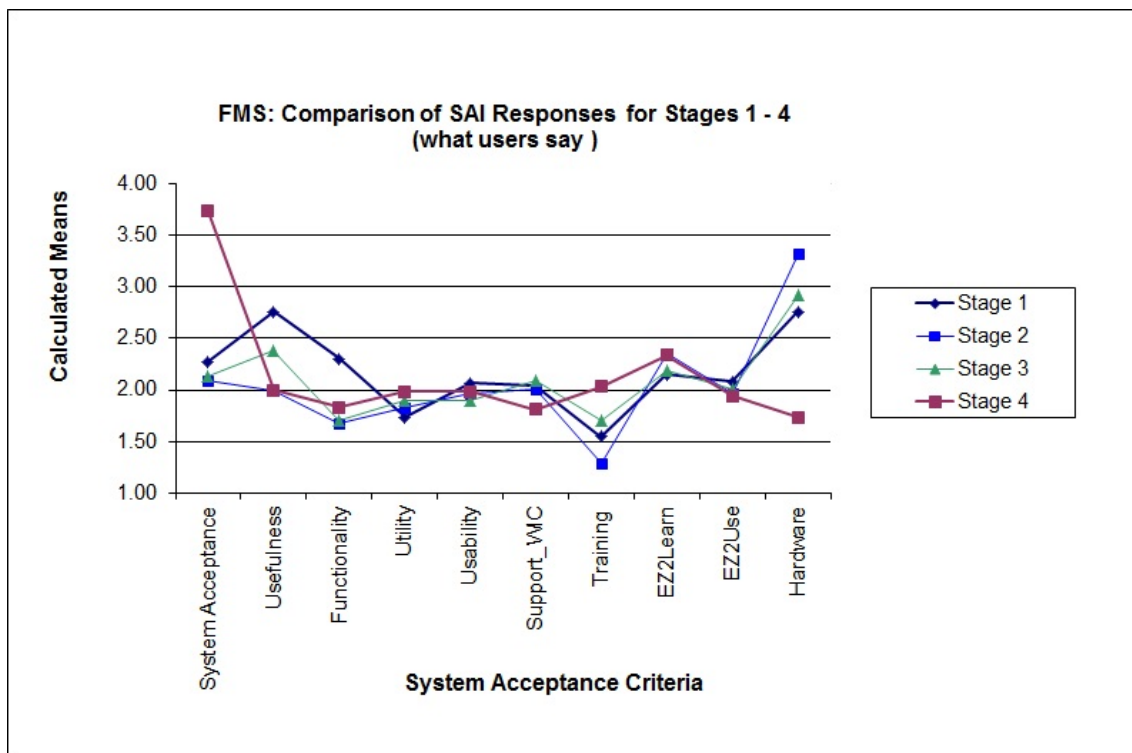


Figure 4. Case Study SAI Traces for Stages 1 to 4.

The issue of concern that arose with the long-term evaluation, however, was the drop in the assessment of the hardware. This criterion, reported through use issues, related to the time required for system network responses and the delays in operational processes. User feedback at the end of the consolidation period indicated that the users were concerned with systemic delays inherent in the Internet implementation. These delays involved queuing for server CPU processing time, queuing due to the maximum limit on concurrent users, and heavy network traffic.

Use Case Conclusions and Summary

The FMS was a mandatory-use system employed within the workplace for a decade prior to the commencement of this study. As such, I hypothesized that some statistical difference would be apparent in the data between the benchmark and the users' transition to the upgraded version. This proved not to be the case. The SAI global evaluation of user assessment remained relatively stable at the established benchmark. However, the theorized rise and fall sequence of user acceptance during the upgrade was observed: The predicted movement of the waveform around the usability engineering criteria, in particular Usefulness and Functionality, was apparent. For the upgrade to successfully leverage existing user practice (i.e., intuitive use), I hypothesized a little movement of the traces around the use-community criteria, and some movement was observed. The users' assessments of training and background knowledge (Support_Use-Practice) were higher after the mentor-based training, underscoring the need for context-specific training. The assessment of the Hardware declined because of the inherent Web technology issues of reduced server and network performances.

The expected fall in usability engineering criteria between Stages 1 and 2 was observed. The largest variation was for the criteria Usefulness and Functionality. The Stage 3 evaluation of Usefulness was higher than for Stage 2. This movement of the reflective assessment criterion Usefulness with confirming feedback from the focus groups was taken to indicate that the survey participants were comfortable with the new system but that new use norms were not established after 3 months use. This assessment was part of the rationale for the intervention.

Usefulness and Functionality were both evaluated lower in Stage 4 than in Stage 1. This result was anticipated because the focus groups were concerned with the limited gap and requirements analyses prior to the commencement of the upgrade project, as well as the number of work-arounds the users implemented. In addition, the user feedback confirmed that the training provided the financial background and use-practice that allowed for a more informed judgment of the overall usefulness and functionality of the system. Based on this assessment, the usefulness of the system within the constraints of the available functionality was improved by training the users in activities specific to their work. The training allowed them to consciously produce output tailored to meet their business process requirements as purposeful actions.

At the emotional level, the use-sensitive training and background knowledge facilitated optimization and confidence, and this was reflected in the high level of system approval at Stage 4. Statements presented at the stakeholders' meetings and comments solicited at the final system assessment by the focus groups supported these interpretations of the participants' responses in the final survey. This agreement was taken as additional endorsement of the validity of the theoretical model in which system acceptance was characterized, but not defined, by usefulness.

The précis of the case study of a system upgrade illustrates how the survey can elicit and monitor the UX across the stages of an upgrade. The UX can be expressed in terms of

immediate use by the usability engineering criteria and the leverage of prior knowledge to make the use intuitive by the use-community criteria. User concerns can be identified at a high conceptual level and investigated in detail to resolve issues. This is the strength of the SAI survey; it is strongly associated with the HCI design concepts that are assessed by the users in their response to the survey questions. The interpretation of the user response, based on previous UX, is intrinsic in the survey analysis methodology.

THE DEVELOPMENT OF THE SURVEY

The previous section was to present the usefulness of a specific type of survey in the context of resolving real-world business related issues. The strength of the SAI is twofold: (a) The criteria were developed from the application of grounded theory analysis (Dick, 2005; Glaser, 1994, 1998) of user-feedback obtained from interviews and surveys, and (b) The theoretical HCI rationale, developed to explain the emergent construct, provides the means to interpret the users' responses in concepts useful for system analysis and design. The following discussion will provide a brief overview of how this was accomplished.

The Survey Development Process

Systems analysts are required to talk to computer users about their computing problems and needs. Initially, it was difficult for me to rationalize and categorize these problems and needs because of the varied descriptions of the issues provided by the users. The process of understanding the users' interpretations of the issues they perceived, as well as categorizing them, led to the realization that a number of problem descriptions reflected the descriptions of issues I had read about in text books and research papers. The users were at times using HCI terminology but with an interpretation of the terms different from that formally used in the field. Consequently, it was necessary to learn to interpret their unique use of HCI concepts and terminology to fully understand the nature of their problems.

Systematic note taking was initiated, followed by a process of coding these notes as references to use again for consistent analysis results. A review of current HCI publications facilitated the coding. These books and papers documented the models, theories, and frameworks for HCI analysis and design. Concepts from a number of HCI paradigms could be applied to specific issues in the users' descriptions of their problems, as well as their requirements for system development. In addition, this review indicated that it might be possible to use the HCI concepts to develop a structured theoretical rationale, using the notes and coding system for analysis to support any recommendations made.

Concepts from a number of paradigms were brought together to form the contextual interpretation base. These concepts included but were not limited to activity theory (Bødker, 1991; Nardi, 1996a; Vygotsky, 1978), situated action (Suchman, 1987), distributed cognition (Hutchins, 1995, 2000), usability engineering (Nielsen, 1993), soft systems modeling (Checkland, 1999; Checkland & Holwell, 1998), cognitive systems engineering/cognitive work analysis (Rasmussen, 1994; Vincente, 1999), scientific management (Taylor, 1911), and the unified theory of acceptance and use of technology (Venkatesh et al., 2003). The 12 emergent

concepts were assembled to characterize the users' descriptions of their computer use experience. These concepts were used as the system acceptance criteria for the SAI.

The development methodology for the SAI followed best practice, as outlined in *Qualitative Data Analysis* (Miles & Huberman, 1994), *Evaluation of Human Work* (Wilson & Corlett, 1995), and the *Handbook for Evaluation of Knowledge-Based Systems* (Adelman & Riedel, 1997). The first stage was to define the objective and set the scope of the survey. The objective was to develop a survey instrument that evaluated the UX using the emergent data categories. In this format, the assessment would determine user satisfaction, as well as identify design and development issues requiring investigation in finer detail. The evaluation format had to be able to present the UX as a series of periodic analyses used to observe the evolution of the system; in other words, provide a numerical universal indicator for the level of user satisfaction and a pictorial presentation of the users' responses to the assessment criteria. This entire process involved five distinct stages.

In the initial stage, consideration was given to the rating method to be used. The Likert summated ratings method—as used by two survey instruments, the System Usability Scale (SUS; Brooke, 1986) and the System Usability Measurement Inventory (SUMI; HFRG, 1993)—was selected as the most appropriate. The reasons supporting this choice are that the Likert scale could be used to yield a single number as a universal indicator and the target audience was already familiar with this method of assessment.

The second stage was to verify and validate the conceptual foundation for the content of the survey that emerged from the data against current practice and research. The theoretical rationale for the grounded theory-determined classifications presented in this article was used as the basis for reverse engineering of SUS and SUMI questionnaires (Lehane, 2012) and the cross-validation by the unified theory of acceptance and use of technology (UTAUT; Venkatesh et al., 2003). During this stage, all of the questions in both surveys and in UTAUT could be classified against the proposed criteria for the SAI. The outcome of the reverse engineering process was early confirmation of the soundness of the emergent theoretical foundation for the SAI.

The third stage was to design the questionnaire content. The wording of the questions was critical to the success of the survey and the project. The questions were developed using simple words and simple sentence construction. The words had to convey a meaning that aligned with the theoretical concepts being assessed and those theoretical concepts had to be understood by the survey participants. The demographic questions of the introductory section were based on the work of Venkatesh et al. (2003) in developing the UTAUT.

The fourth stage was a pilot study using the draft questionnaire. Participants were invited to use the “think aloud” protocol while completing the questionnaire. Integral to their completion of the survey on their use of software were comments on the survey questions themselves and the intended meaning of the words and questions. The rationale for this was to assure that words with the widest shared understanding were used in the composition of the survey. Their comments and questions were considered and, where appropriate, the questionnaire was modified in light of the feedback.

The fifth stage was to implement the SAI in an action research project and assess the cognitive UX in workplace-situated instances of technology introduction. This was the stage during which the data presented in this paper were collected, analyzed, and interpreted. The outcomes of these processes were presented to the two stakeholder groups of the case presented in this article.

Primary and Qualifying Questions

The SAI analysis methodology assigns each question to a primary acceptance criterion and then re-assigns the same question into secondary groupings that qualify the various primary criteria. Thus, the survey responses are presented as two traces. In practice, the primary trace is derived from the high-level reflective assessment of the users' work with the system. By way of example, a primary question might ask: Was the system easy to use? The secondary trace is obtained by combining the response of this primary question with qualifying responses to other primary questions that describe how the user accomplished the task upon which the reflective assessment was made; in other words, how they accomplished what they said they did. If a system is easy to use, then the user should (a) know the actions the screen artifact affordances represent, (b) be in the flow using the system and not comment adversely about limitations or impediments to seamless work processes, and (c) not be confronted with unexpected outcomes from the system. These secondary questions confirm how easy the user actually found the system to use. The grouping of the questions for primary and secondary assessment is shown in Table 5.

To calculate the values plotted in the graphs, each criterion value in the primary trace is the average of the responses to the questions in that grouping (see Table 5, Column 2). The secondary trace is the average of the sum of both the primary and the secondary questions in that grouping, as in Table 5, Column 3. If the descriptive aspects of situated practice (i.e., the secondary questions) as modifiers of the primary question are rated lower, then the evaluation will be lower for the secondary question trace. The up-to-date grouping of questions as primary and secondary is presented in Table 5.

I present here the survey questions, with the rationale of each question as originally drafted. A detailed description of the use case for each question is available in Lehane (2012). A copy of the SAI as a form is available in the Appendix.

Table 5. The Grouping of Questions for Analysis.

Category	Questions, primary grouping	Questions, secondary grouping
Usability Engineering		
System acceptance	1	1,2,3
Usefulness	2	2,3,7
Functionality	3	3,10,11
Utility	4,6,20	4,6,20,10,24,25
Usability	5,24,25	5,24,25,4,6,22
Use-Community Criteria		
Support work-in-context	7,8,10,11	7,8,10,11,1,2,3,12
Active user	13,14	13,14,2,5
Distributed cognition	17,19	17,19,15,16
Affordances	16,18	16,18,15,25
Support use-practice	15,22,23,	15,22,23,5,12,25
Training	12,21	12,21,22,23
Hardware	9	9,20

1. *I would recommend this software to a colleague or friend.* System acceptance, a recommendation for future use by a member of the community of practice, indicates acceptance of the tool into the community by that community member.
2. *Using this software makes me feel bad, i.e., anger, frustration, stress, confusion, headache, or body pains.* System acceptance comes from a positive user experience of the usefulness. If the user feels anger, frustration, rage, tenseness, headaches, or illness, then the user will reject the system. If the user feels satisfaction, confidence, or mental stimulation, these feelings generate a positive user experience and system acceptance. For mandatory-use systems, the user rejection will not impact continued use. However for voluntary-use systems, a sense of ownership is implicit in system acceptance. Rejection of a voluntary-use system leads to discontinued use.
3. *This software does all the things that I need it to do.* Software functionality must mediate all of the activities of the practice, as well as support all of the actions in the activity. The application of the skills, rules, and knowledge paradigm (SRK; Vincete, 1999) apply here in the interface design.
4. *This software takes too many steps to get something done.* Efficiency is measured in the time of and/or the stages in a process (i.e., the number of actions to accomplish an activity). Design should streamline the process to minimize interaction.
5. *I thought this software was easy to learn.* The intent of this question is to verify the basic tenets of ease of learning and use. The assumption here is that elements from distributed cognition (Hutchins, 1995, 2000) and activity theory (Vygotsky, 1978) that make software easy to learn also make it easy to use.
6. *Using this software is slow because I have to make a lot of keystrokes or mouse clicks to do the work.* Efficiency and effectiveness in design-for-use should produce functionality tailored for the use context. The tailoring should produce functionality that is precise and of succinct operation. These two properties should enable actions to be readily operationalized. The artifact affordances needed to undertake the activity should be designed to support the workflow of the actions in the process and support operationalization of actions.
7. *This software makes my job easier.* The workflow of the software should support existing practice, sequence the workflow, clarify and/or affirm terminology, and integrate feed-in/feed-out systems.
8. *This software calls things I am familiar with by a different name.* The terminology used to describe the process and the elements associated with the process are not changed with a change of software, meaning the software does not make changes to existing practices. The use of the noncustomized American enterprise system changed the terminology used in Australian universities: For example, the Australian English term *unit* was replaced with American English term *course* where, previously in the Australian context, *course* was a set of units in a program.
9. *The computer (i.e., the hardware) is adequate; it does not need upgrading to run this software.* Adequate hardware is essential for the integration of new software into work processes. A mismatch between the computer specifications and software requirements impacts the operation of the software. The issue may be that the computer's central

processor is too slow, the computer needs more memory or faster peripherals, the network may be too slow, or the screen is the wrong size or resolution.

10. *Using this software, there are too many things to do to get the job done.* Additional work to complete a similar process negatively affects a user's perception of the upgrade: Changes in work practice could involve the user having to manage some form of housekeeping for the software, or an increased complexity due to numerous new stages in the process. Ideally, the process flow and housekeeping would be automated by the software.
11. *Because of this software, the job involves less follow-up work away from the computer.* Ancillary workload involves tasks away from the computer that support or result from use of a software program, such as feed-in/feed-out systems.
12. *The training to use this software was inadequate.* The successful introduction of software requires a suite of complementary activities. These include training in the use of the software, information to the users about the effects on the work of associated activities, and how the software benefits the user now and in the long term. If investments in learning or setting up the system for long-term benefits are needed, these have to be explained and have commensurate adjustments to workloads.
13. *When first introduced to this software, my main aim was to get an output as soon as possible.* This confirming question addresses lower level operational criteria. In the active user production paradox (Carroll & Rosson, 1987), the user is output driven, and once a solution is found, the solution becomes the method. Changing software may require extra work, but how the task is performed within the new software may or may not change. Thus, this question was directed toward determining whether the user is output driven or participates in exploratory behavior.
14. *When first introduced to this software, my main aim was to try as many things as possible.* In the active user assimilation paradox (Carroll & Rosson, 1987), the user tries familiar actions in unfamiliar situations. Therefore, this question is directed toward determining what composes that behavior pattern; the crux is whether or not the user is even interested in exploring other possibilities.
15. *There is always enough information on the screen when it's needed.* A tenet of distributed cognition (Hutchins, 1995, 2000) is that information is distributed between humans and the artifacts within the environment so that the use of the tool is intrinsic within the context of the activity. Distributed cognition implies the presence of the appropriate artifact to facilitate the behavioral task that supports the cognitive task (motive) driving the activity.
16. *Looking at the screen, sometimes I do not know what to do next.* The intent of this question is to consider the metaphor behind the application, that is, the metaphor for any particular screen or screen artifact. Holistically the metaphor should be congruent with the purpose of the activity. The artifacts should provide an individual context for the actions/operations, depending on the user's focus and familiarity. The affordances of the artifact imply what to do and how to do it.
17. *The screen for this software looks like other screens I have used.* The layout of the screen artifacts constituting the functionality of the system, or the composition of

the menus, are not the same on this screen as they are for other screens the user has experienced, or could be expected to come into contact with. The aim here is to check that there is visual consistency in layout, absolute and relative positioning, and the visual representation of artifacts is consistent for artifacts of similar functionality (i.e., consistency in icons or graphics for artifacts that relate to a common function). One of the objectives of design should be the leverage of prior knowledge in use-community praxis.

18. *Sometimes this software does things that I did not expect to happen.* Artifacts relative to the use context should be visible with visible affordances. The operability of the artifact depends on the visibility of the affordances or on the learned behavior of hidden affordances. Consequently, hidden affordances—a property (short form menu) or a method (right-click on mouse-over of artifact)—should be consistent across the occurrences of the artifact in the application and across occurrences in the practice of the use-community. False affordances should be modified or removed to minimize inappropriate expectations.
19. *When I look at the icons and menus, etc. on the screen. I know what to use and how to use it.* The screen artifacts have affordances: One aspect of affordances is the physical representation of the artifact. This physical representation has to incorporate cues to the operation of the artifact as a tool and the possible uses of the artifact as a tool.
20. *It takes too long for me to see things happening on the screen.* The directness and operational transparency of the artifacts relate to the response time of the direct manipulation interface to give feedback to the user.
21. *I found the help in the software to be very useful.* An integrated and contextually related help subsystem of the software is specified in checklists of system functionality.
22. *There is a lot to learn before using this software.* This question addresses the depth to which the user was immersed in the use-community and the amount of prior knowledge that is available for leverage from the user as a member of that community. It does not assess, however, the understanding of the artifacts used to leverage that prior knowledge.
23. *I think I will be able to use this software without asking for help from the experts who know how to use it.* This question is active-user based and directed towards output and training support systems. The user indicates that he/she has the knowledge and experience necessary to use the software. The affordances and metaphors are familiar to the user, who is able to recognize and associate actions with the on-screen display.
24. *This software is difficult to use because I have to work all over the place on different screens.* Functionality should support the activities in the work domain. The question is not about the efficiency of the functionality but rather about utility, the efficiency of using the functionality.
25. *For the things that I use, this software looks and works the same way, every time.* Building user confidence is important. This can be achieved early in the user experience by using meaningful artifacts from the distributed cognition of the use-community. This implies consistency of operational methods for activation of

command sets. The operability of artifacts should be consistent with the metaphor and consistent with the task–action grammar.

CONCLUSIONS

This article presented the SAI survey (Lehane & Huf, 2005, 2006) based on classification criteria that arose from a grounded theory analysis of users' descriptions of their computer use. The central theme for analysis interpretation and issue resolution is community-of-practice-scripted user behavior, based on the tool-associated practice of the use-community. The TAM graph of the SAI system acceptance criteria facilitates the presentation of the elements of scripted behavior (as interpreted by the survey), these being Active User, Distributed Cognition, Affordances and Support_Use-Practice.

In the case study, the SAI analysis identified the upgraded FMS as not being intuitive and proposed a means to resolve the issue. The users were deskilled and had to reskill on the use of the upgraded system, with training focused on background knowledge to make the context of the screen artifacts and their affordances familiar. The experience gained from the training and induction into the finance-community of practice enabled the users to understand what was presented and to navigate confidently the screens and menus. This circumstance illustrated that the difference between being intuitive and not being intuitive was their having to learn to use the system instead of just knowing how to use it by looking at it.

HCI has the design convention of leverage of prior knowledge that I implement as “give the users what they already know” and thus interpret as intuitive design. In the case study, it was necessary to train the users so that they had the knowledge they needed to make sense of the new FMS and use it confidently; the situation for the design convention was reversed. The design incorporated use-knowledge that was unknown and the integration between the users and the technology was to provide the background training assumed by the system designers. Once that experience was formally established, the HCI design convention of leverage of prior knowledge could be seen to be implemented in the new FMS.

The intent of this article is to illustrate the usefulness of the SAI as an analysis practice to elicit user requirements and relate that feedback to system design considerations for intuitive use. The foundation for the link between the users' input and design is the concept of scripted behavior from a community of practice. The theoretical considerations accompanying the development of the survey and the methodology to interpret the users' responses will be presented in Lehane (in press).

REFERENCES

- Adelman, L., & Riedel, S. (1997). *Handbook for evaluating knowledge-based systems*. Norwell, MA, USA: Kluwer Publishers.
- Bauer, W., & Gaskell, G. (Eds.). (2000). *Qualitative researching with text, image and sound: A practical handbook*. London: SAGE Publications.
- Bødker, S. (1991). *Through the interface: A human activity approach to user interface design*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates.

- Brooke, J. (1986). *System usability scale*. Reading, England: Digital Equipment Corporation.
- Carroll, J., & Rosson, M. (1987). Paradox of the active user. In J. M. Carroll (Ed.), *Interfacing thought: Cognitive aspects of human-computer interaction* (pp. 80–111). Cambridge, MA, USA: MIT Press.
- Checkland, P. (1999). *Systems thinking, systems practice: Includes a 30-year retrospective*. Chichester, England: John Wiley and Sons.
- Checkland, P., & Holwell, S. (1998). *Information, systems and information systems: Making sense of the field*. Chichester, England: John Wiley and Sons.
- Dick, R. (2005). *Grounded theory: A thumbnail sketch*. Retrieved June 10, 2005, from <http://www.scu.edu.au/schools/gcm/ar/arp/grounded.html>
- Glaser, B. (1994). *More grounded theory methodology: A reader*. Mill Valley, CA, USA: Sociology Press.
- Glaser, B. (1998). *Doing grounded theory: Issues and discussion*. Mill Valley, CA, USA: Sociology Press.
- Hasan, H., Gould, P., Larkin, P., & Vrazalic, L. (Eds.). (2001). *Information systems and activity theory: Vol. 2, Theory and practice*. Wollongong, Australia: University of Wollongong Press.
- Human Factors Research Group [HFRG]. (1993). *Software usability measurement scale*. Cork, Ireland: HFRG.
- Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA, USA: MIT Press.
- Hutchins, E. (2000). Distributed cognition. Retrieved April 18, 2000, from <http://eclectic.ss.uci.edu/~drwhite/Anthro179a/DistributedCognition.pdf>
- Jones, P. (1995). A methodology for human-machine systems research: Knowledge, engineering, modelling and simulation. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(7), 1025–1038.
- Kontogiannis, T. (1996). Stress and operator decision making in coping with emergencies. *International Journal of Human-Computer Studies*, 45, 75–104.
- Kuutti, K. (1996). Activity theory as a potential framework for human-computer interaction research. In B. A. Nardi (Ed.), *Context and consciousness: Activity theory and human-computer interaction* (pp. 17–44). Cambridge, MA, USA: Massachusetts Institute of Technology.
- Lehane, P. (2010). Designing interactions for the collective user experience. In *Proceedings of the 22nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction (OZCHI 2010)*; pp. 120–127. New York: ACM. doi: 10.1145/1952222.1952247
- Lehane, P. (2012). *The system acceptance indicator: An analysis practice emergent from the plans, scripts and intuitive interaction of the collective user community* (Unpublished doctoral dissertation). University of South Australia, Adelaide, Australia.
- Lehane, P. (in press). Make it intuitive: The theoretical constructs of an analysis practice emergent from the plans and scripts of the computer community of practice. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*.
- Lehane, P., & Huf, S. (2005). Towards understanding system acceptance: The development of an assessment instrument in practice. In *Proceedings of the 17th Australia Conference on Computer-Human Interaction (OZCHI '05)*; pp. 1–9. Narrabundah, Australia: Computer-Human Interaction Special Interest Group of Australia. Available at <http://dl.acm.org>
- Lehane, P., & Huf, S., (2006). Designing interaction: Reduction in activity time by the re-allocation of user tasks to machine functions. In *Proceedings of the 18th Australia Conference on Computer-Human Interaction (OZCHI '06)*; pp. 449–453. Available at <http://www.ozchi.org/proceedings/2006/sessions/industry/lehane-p449.pdf>
- Miles, M., & Huberman, A. (1994). *Qualitative data analysis*. London: SAGE.
- Nardi, B. A. (Ed.). (1996a). *Context and consciousness: Activity theory and human-computer interaction*. Cambridge, MA, USA: Massachusetts Institute of Technology.
- Nardi, B. A. (1996b). Studying context: A comparison of activity theory, situated action models, and distributed cognition. In B. A. Nardi (Ed.), *Context and consciousness: Activity theory and human-computer interaction* (pp. 69–102). Cambridge, MA, USA: Massachusetts Institute of Technology.

- Nielsen, J. (1993). *Usability engineering*. San Diego, CA, USA: Morgan Kaufman.
- Pawlak, W., & Vincente, K., (1996). Inducing effective operator control through ecological interface design. *International Journal of Human-Computer Studies*, 44, 653–688.
- Preece, J., Rogers, Y., & Sharp, H. (2007). *Interaction design: Beyond human-computer interaction* (2nd ed.). New York: Wiley.
- Rasmussen, J. (1994). *Cognitive systems engineering*. New York: John Wiley & Sons Inc.
- Suchman, L. (1987). *Plans and situated actions*. Cambridge, MA, USA: Cambridge University Press.
- Taylor, F. (1911). *The principles of scientific management*. New York: Harper Brothers.
- Venkatesh, V., Morris, M., Davis, G., & David, F. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3), 425–478.
- Vicente, K. (1999). *Cognitive work analysis*. Mahwah, NJ, USA: Lawrence Erlbaum Associates Inc.
- Vygotsky, L. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA, USA: Harvard University Press.
- Wilson, J. R., & Corlett, E. N. (Eds.). (1995). *Evaluation of human work: A practical ergonomics methodology* (2nd ed.). Philadelphia, PA, USA: Taylor & Francis.
-

Author's Note

The SAI and an Excel worksheet with notes on use are available for use by those assessing the introduction of technology into the workforce (<https://eportfolio.usq.edu.au/view/view.php?t=Zw1TAYBxKMOG5fUPunyi>). The only prerequisite for the use of the SAI or the inherent analysis practice is that any published report should acknowledge the source of the measure and practice.

All correspondence should be addressed to
Pat Lehane
P.O. Box 32
Darling Heights
Queensland Australia 4350
pat.lehane@usq.edu.au

Human Technology: An Interdisciplinary Journal on Humans in ICT Environments
ISSN 1795-6889
www.humantechnology.jyu.fi

APPENDIX

This provides a scan of the hardcopy version of the SAI. The same basic layout was used for the online versions. Responsibility, experience, and frequency of use may vary with the survey campaign.

In addition to the closed questions provided here there were four open questions worth considering, because the responses could negate the need for or complement the use of focus groups. These questions were along the lines of

1. For you, what was the best thing about using the software?
2. What annoyed you most about using the software?
3. What would you change?
4. As a user, what is the first thing you would put on the wish list?

System Acceptance Indicator

INSTRUCTIONS This questionnaire should be filled in, in little more than the time it takes to read it. There are no correct answers to the questions. Please place a cross in the box that best indicates how you first feel on reading the question. Give a single response to each question. Please fill in the reference information before responding to the questionnaire.

Reference Information:

Date..... Software[Test Identifier.....OfficeUseOnly]

Year of Birth:

Gender: female male

Conditions for use of software: mandatory voluntary

Responsibility level for software: manager user

Do you like using the software: no yes

Experience with software:

less than 1 month less than 3 months less than 1 year more than 1 year

Frequency of Use:

several times a week a month 3 - 6 months 6 - 9 months a year

	Strongly agree	Undecided N/A	Strongly disagree	
Q1. I would recommend this software to a colleague or friend.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q2. Using this software makes me feel bad ie anger, frustration, stress, confusion, headache or body pains	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q3. This software does all the things that I need it to do.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q4. This software takes too many steps to get something done.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q5. I thought this software was easy to learn.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q6. Using this software is slow because I have to make a lot of keystrokes or mouse clicks to do the work.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. This software makes my job easier.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q8. This software calls things I am familiar with by a different name.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q9. The computer is adequate, it does not need upgrading to run this software.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Strongly agree	Undecided N/A	Strongly disagree
Q10 Using this software, there are too many things to do to get the job done.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q11. Because of this software, the job involves less follow-up work away from the computer.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q12. The training to use this software was inadequate.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q13. When first introduced to this software, my main aim was to get an output as soon as possible.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q14. When first introduced to this software, my main aim was to try as many things as possible.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q15. There is always enough information on the screen when it's needed.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q16. Looking at the screen, sometimes I do not know what to do next.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q17. The screen for this software looks like other screens I have used.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q18. Sometimes this software does things that I did not expect to happen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q19. When I look at the icons and menus etc on the screen I know what to use and how to use it.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q20. It takes too long for me to see things happening on the screen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q21. I found the help in the software to be very useful.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q22. There is a lot to learn before using this software.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q23. I think I will be able to use this software without asking for help from the experts who know how to use it.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q24. This software is difficult to use because I have to work all over the place on different screens.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q25. For the things that I use, this software looks and works the same way, every time.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>