ISSN: 2167-1435
v30 n2560

In Search of Design-Focus in IS Curricula

Jeffry S. Babb
jbabb@wtamu.edu
Computer Information and Decision Management
West Texas A&M University
Canyon, TX 79106 USA

Leslie J. Waguespack
lwaguespack@bentley.edu
Computer Information Systems Department
Bentley University
Waltham, Massachusetts 02452, USA

Abstract

Curricula in information systems embrace a broad range of topics that leave the identity of information systems as a discipline somewhat in flux. In the spirit of "the first among equals," we posit that design should have preeminence in the education of information systems professionals. Design frames problem understanding and defines what system's quality means. It behooves our profession to prepare designers who deliver systems that not only "work," but also deliver systems that society will recognize as "working well." The research community recognizes this as reflected in a renewed interest in design science research and in information systems design theory. While our discipline has been recently reshaped by offshoring, outsourcing, and service-oriented architectures, which provide myriad options for managing information in organizations, design persists as a central aspect of the discipline. This is so as information systems design remains close to stakeholders because design materializes an organization's core business model and strategy. This paper contemplates a design-focused IS curriculum and postulates a perspective on design that values the subjective sensibilities of stakeholders as well as an objective, algorithmic depiction of computing. The latter has shaped the classic education of a developer as a master of technology while the former nurtures an aesthetic awareness that captures nuances of stakeholder satisfaction and a more inclusive conception of system quality. The skillset of designers is a superset of that of developer and as such, a designer must be craftsman and more, a reflective practitioner skilled in the art of generative metaphor.

Keywords: Information Systems Curriculum, Information Systems Design, Thriving Systems Theory, Reflective Practice, Mastery learning.

1. INTRODUCTION

Design is not a core focus in contemporary information systems (IS) education (Waguespack, 2011). Yet it is a palpable force in the evolving role of computing in the everyday life of individuals, organizations, and business and, in many cases, has redefined normality as we know it (Christiansen, 1997). For example, as a company, Apple Inc. has been important not just as a technology leader, nor just as an innovative leader in the marketplace, but particularly for a marked, tenacious, and overt focus on the importance of design (Turner, 2007). Design is a central subject in the arts, and particularly in architecture (Alexander,

 ISSN: 2167-1435
v30 n2560

2002). Design delves into the human's capacity for subjectivity and aesthetic experience that does not succumb readily to the measuring tape or the algorithm. Despite its seeming absence from most programs in IS, design is what separates a system that "works" from a system that "works well!" – a sentiment perhaps most forcefully set forth by Fred Brooks:

Whereas the difference between poor conceptual designs and good ones may lie in the soundness of design-method, the difference between good designs and great ones surely does not. Great designs come from great designers. Software construction is a creative process. Sound methodology can empower and liberate the creative mind; it cannot inflame or inspire the drudge (Brooks, 1987).

In this light, of both design's emphatic impact on computing's role in everyday life and the challenge of developing great designers, this paper explores formulating a design-focused IS curriculum based upon a design perspective that values the subjective sensibilities of stakeholders as well as an objective, algorithmic depiction of computing. We assert the centrality of design even despite the changes wrought upon IS manifested in the outsourcing/offshoring of construction, the rapid emergence of the pervasive and ubiquitous computing brought by mobile computing, and a trend towards service-oriented architectures (Babb and Keith, 2012).

This paper proceeds as follows: First, we begin with a brief, selected review of relevant design research. We next argue the centrality of design in information systems to address the essential difficulties of the IS domain. We follow the influence of Christopher Alexander's *living structures* theory of design to introduce subjectivity as an integral aspect of design quality and use Thriving Systems Theory's (Waguespack, 2010) design quality clusters to illustrate. We then review perspectives on learning and action, for pedagogy and practice, utilizing guidance from Argyris and Schön (1974, 1978, 1996), describing how reflective practice illuminates the progression from student to master. Two extant college programs are used to illustrate a focused apprenticeship/craftsmanship model that may be better suited to developing pedagogy for design. We conclude considering next steps required to formulate IS education with design at its center.

2. DESIGN IN INFORMATION SYSTEMS RESEARCH AND CURRICULA

IS, as a discipline, has been in flux for some years (Alter, 2008; Benbasat & Zmud, 2003; Walsham, 1993). If anything, design as a focus has diminished in IS curricula rather than grown. If we inspect IS model curricula as surrogates for defining the discipline it is clear that "… [the] distinction between design and implementation has faded from the structure of computing education. To ignore the conceptual distinction between the *design* and an *implementation* is tantamount to accepting any "solution" without even considering [quality]…" (Waguespack, 2011)

In IS research, however, there is a renewed interest in design; a recognition that design quality should not be an insignificant or accidental result of systems development. Design Science research has grown into a movement (Hevner & Chatterjee, 2010) and Information Systems Design Theory (ISDT) is finding shape as a means of promoting quality systems. (Walls, 2004, Gregor, 2007)

Design (as manifested in object-oriented programming) has drawn guidance from physical art and architecture in Christopher Alexander's pattern languages and the notion of design patterns (Alexander, 1977, 1979; Gamma et al., 1995). Alexander advocates, as a prime aim of design, to search for the "Quality without a Name," or perhaps, a "je ne sais quoi" which captures the essence of designing. That is, to speak of design is to speak of quality (Alexander, 1979). Alexander's theory of *living structure* underpins Thriving Systems Theory of design quality in information systems (Alexander, 2002; Waguespack, 2010; Waguespack & Schiano, 2012, 2013). We can draw an arc of design influence from Christopher Alexander, to the "Gang of Four," to Ward Cunningham and Kent Beck, as manifested in object-orientation, the Unified Modeling Language, design patterns, and agile methodologies. (Beck et al., 2001)

3. CENTRALITY OF DESIGN

Generally speaking, the predominant heritage of IS design closely aligns with the positivist philosophy of mechanistic or mathematical artifacts that is indifferent to any subjective or aesthetic qualities. This attitude proceeds from the natural sciences that focus on explaining

ISSN: 2167-1435
v30 n2560

extant physical and biological structures. At their core, the natural sciences are about determining "why" objects in nature exist as they do – basically taking intact, functioning "objects" apart to see what they are made of and how they work. For the most part these objects would exist with or without human attention. Information systems, however, are artificial, they manifest as "things" that exist beyond the "natural" world.

Information system artifacts do not exist independent of humans and human organization. They are human-made and reside in a sociological context where they evoke some degree of human satisfaction based on the value individuals or society perceives in them (the business moniker might be "cost/benefit"). The "value" of an object in the natural sciences view vests in its existence and/or survival with any human satisfaction based on "accident of nature." In contrast, the very existence of an information system (a human-made artifact) depends upon its value as perceived by a society of stakeholders (ostensibly that value is the reason the system was constructed). Therein lies the essential difficulty of IS design, meeting the human conception/perception of value and satisfaction: quality. In this sense, it is appropriate to say that design holds the central role in information system success.

Designing quality in IS artifacts entails: 1) a grasp of functional needs, 2) an aesthetic sensibility attuned to the stakeholder(s)' perception of quality and 3) the skill to engage technology that allows a formulation of (1) which allows (2) to resonate. Design in this formulation of quality is central to the entire IS discipline: technology, society, organization, management, and operation – every relevant aspect of IS.

Design and Subjective Resonance

Thriving Systems Theory (TST) is an emergent design theory that promotes specific emphasis on aesthetic sensibility that is attuned to the stakeholder(s)' perception of quality (Waguespack & Schiano, 2013). TST rests on three pillars of theory: Christopher Alexander's *living structure* in *The Nature of Order* (Alexander, 2003); Lakoff and Johnson's cognitive-linguistics and conceptual metaphor that explain human understanding and perception (Lakoff, 2008); and Fred Brooks' essence and accidents in systems development (Brooks, 1987). The full exposition of TST is found in (Waguespack, 2010).

"Everything that can be counted does not necessarily count; everything that counts cannot necessarily be counted." – Albert Einstein

TST's emphasis on subjectivity and aesthetics relies upon three concepts: 1) human perception is mediated by innate conceptual metaphors through which we recognize ordered-ness, 2) the transmission of ideas through any form of human communication is imperfect and therefore all communication is metaphorical, and 3) any conception of reality is incomplete therefore satisfactory communication relies on conscious and careful abstraction (Waguespack, 2010).

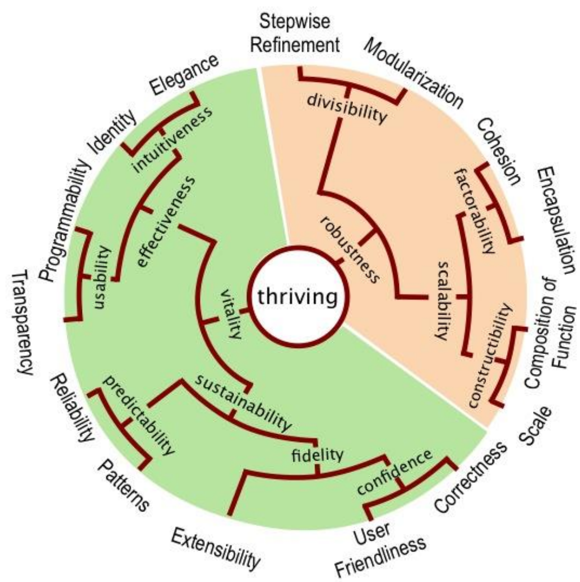

Figure 1. Choice Property Clusters

TST translates fifteen properties of design, identified by Alexander (1979), that convey a sense of *living structure* into the context of information systems. An analysis of the supporting relationships among the choice properties of TST exposes property clusters and weaving patterns of resonance that exhibit discernible design qualities. The clusters compose a hierarchical arrangement, a combining of resonance that converges to a comprehensive confluence of design affect. In Alexander's theory, it is in the ultimate confluence of these properties that an observer perceives a degree of *wholeness* in the design – a level of satisfaction that presages design

ISSN: 2167-1435
v30 n2560

quality. TST assigns names that denote the constituent properties. In Figure 1 the choice properties line the perimeter while the clusters converge toward the center of the diagram in four levels of confluence exposing two "families" of quality, robustness and vitality, which fuse into the quality that is the design theory's namesake, thriving.

A full exposition of the choice property clusters, the rationale for naming them, and the effect of their confluence of their constituent choice properties is found in Waguespack (2010).

4. DECOMPOSING DESIGN

We propose two dimensions of design competency in order to explore design-centric IS pedagogy: 1) the breadth of design quality addressed and 2) the depth of skill/expertise in realizing these qualities in artifact design. The skill/expertise dimension represents an accumulation of knowledge, but also implies an aspect of "absorption" to indicate what might be a reasonable expectation for a particular "learner."

Figure 2. Design Quality's Relationship With Skill/Expertise Level

Generally, only the most sophisticated of students would be able to address design in the abstract without a specific context or application domain. Accordingly, we stipulate that there must be some prerequisite domain knowledge to serve as a "sandbox" within which to demonstrate and practice design concepts where the student already recognizes domain objects and has some idea of their role and inter-relationships. Thus, design requires the acquisition and skillful utilization of domain knowledge. Within this context the scale of skill/experience might be described as in Table 1 intimating the individual's ability to understand and/or construct an artifact in a particular domain.

Table 1. Skill/Expertise Competency

Skill/Expertise Level	Competency
Master	Authoritatively knowledgeable
Professional	Trusted practitioner
Journeyman	Trained practitioner
Apprentice	Student in training
Novice	Beginning student
Consumer	A user of the product artifact

Among the design qualities of TST, novices will find the "robustness" design quality family easier to absorb because they express concepts of design that are more tractable. Indeed, many might argue that these are the very design "principles" that have traditionally shaped the curricula of systems development in computer science and information systems.

The "robustness" quality family can be demonstrated directly in the examination of entry-level software development coursework: programming, data structures, and computer organization, etc. This family gives the impression (at least to students) of rather static qualities and thus is often characterized as structural. There are convenient visual representations – static diagrams or charts – that allow students to learn to recognize design (noun) elements. When the time comes to ask the student to "create" a solution rather than understand an existing one – that is the point when the student engages "design" (verb). Up until that point we've only tasked them to "recognize" design (noun). The difference is not subtle. Every IS educator sees students who don't make the transition from recognition to performance readily and some fail to make this important transition at all.

Turning our attention to the "vitality" quality family, students must recognize a dynamic rather than static quality of "behaving" or "evolving" – as in Alexander's (1977, 1979) conception of *living structure*. These concepts are not so easily represented in diagrams or charts, as there is conceptual "movement" that needs to be "seen." There is some potential for "stop-motion" as in sequence diagrams in UML. But the full import of extensibility or reliability is a quality of "movement" or "evolution" requiring

 ISSN: 2167-1435
v30 n2560

imagination on the part of the student. They must "compose" a mental image of the concept that captures a variety of implications that stem from vitality design qualities and will impact the artifact's users and stakeholders.

Many of the TST/Alexander design qualities are too challenging for students at the lower skill/experience levels (novice or apprentice) to fully absorb, comprehend, or appreciate; since they probably don't have a broad or deep enough grasp of the application domain objects sufficient to recognize the nuances explained by the clusters. It is this grasp of the design qualities' impact on satisfaction that sets apart the upper skill levels (professional and master). That is, a designer generatively and iteratively evolves towards these higher orders through years of "conversing with" these materials of design (Schön, 1987).

As students and practitioners progress along the axis of professional maturation towards mastery, "imaginative visualization" becomes a key aspect of abstract thinking – a challenge for teachers as much as students when it comes to conveying abstract ideas. Moreover, much of this maturation will transpire *in situ,* in practice, as expertise continues to develop long after the classroom and laboratory experiences fade. An imagination is important as it is a "way of seeing" and, perhaps, "not seeing" that shapes design. That is, imagination is a means for matching experience to a new and/or wider context (Mills, 1959).

Curiously or coincidentally enough, this ability to wield imagination is itself a design challenge. This is "abstract thinking" as in "object think" or "relational think" where the mental image of the problem space provides the building blocks of the paradigm. This is, in another name, *metaphor-driven* where each individual has her own image of a concept seeking aspects of consistency that others will recognize and share the concept (Lakoff, 2008, Waguespack, 2010). This is a critical pedagogical challenge: how does a student's capacity for *abstract thinking* or *thinking metaphorically* develop? Reflective practice offers a promising protocol.

5. EDUCATING THE REFLECTIVE DESIGNER

Graduates of IS programs usually obtain employment and career-building experiences based upon the technical and construction skills they develop in (and out of) the classroom. But what are the seeds that should be planted and nurtured that precipitate higher orders of imagination, invention, problem solving – quality design?

Argyris and Schön (1974, 1978, 1996) and Schön (1983, 1987) research the individual and collective competencies that facilitate the generative process of learning that leads to mastery. They prescribe three specific competencies:

1) **Generative Metaphor** (Schön, 1983; 1987)
2) **Reflective Practice** (Schön, 1983; 1987)
3) **Double-Loop Learning** (Argyris and Schön, 1974, 1978, 1996)

While not exhaustive, these theoretical lenses pose a means to understand, in action, the various daily habits, norms, and competencies that can augment technical instruction to encourage and facilitate learning in IS students and practitioners in advancement toward mastery.

Generative Metaphor

Design quality relies heavily on the role of metaphor to achieve stakeholder satisfaction. Thriving Systems Theory asserts that a thriving system is the result of models (metaphors) that capture and reflect the stakeholders' intentions along with careful choices of applying technology that resonate in the design (Waguespack, 2010).

Schön offers insight into these relationships with his concept of "generative metaphor" (Schön, 1993). A generative metaphor sets the problem in context with a "naming and framing" process. Metaphor, in this case, is used in a manner very similar to Lakoff's (2008) conception, as a projection of the problem in terms of a familiar surrogate. The metaphor names and frames the problem, proposing a set of potential solutions and priming a series of attempts to map these known solutions onto this problem. For instance, it is possible to characterize the run-down nature of a neighborhood by describing the neighborhood as "blighted." Since blight is typically used as a term to describe disease in plants (and other organisms), setting this problem, metaphorically, reveals "treatment" approaches to that problem. The designer's choice of metaphor maps her past solution experience onto this problem and adjusts the solution's treatment to the differential.

ISSN: 2167-1435
v30 n2560

Reflective Practice

Schön introduces *Theory of Action* (Argyris and Schön, 1974, 1978, and 1996) concepts of thinking in and thinking on action to develop a model of a reflective practitioner (Schön, 1983, 1987). Reflective practice is about building professional repertoire, particularly for those whose professional activities involve design. Both imagination and intuition develop through daily experimentation and reflection, reflecting *while doing* and *after doing* in a cycle that allows for error detection and correction. Reflective practice is a generative loop of discovery, classification, and application.

The practitioner allows himself to experience surprise, puzzlement, or confusion in a situation which he finds uncertain or unique. He reflects on the phenomenon before him, and on the prior understandings which have been implicit in his behaviour. He carries out an experiment which serves to generate both a new understanding of the phenomenon and a change in the situation. (Schön 1983: 68)

In reflective practice, the designer builds repertoire – a collection of concepts, ideas, visuals, examples, mistakes and actions – to draw upon in subsequent decision-making and designing.

When a practitioner makes sense of a situation he perceives to be unique, he sees it as something already present in his repertoire. To see this site as that one is not to subsume the first under a familiar category or rule. It is, rather, to see the unfamiliar, unique situation as both similar to and different from the familiar one, without at first being able to say similar or different with respect to what. The familiar situation functions as a precedent, or a metaphor, or… an exemplar for the unfamiliar one. (Schön 1983: 138)

Double-Loop Learning

Argyris and Schön's (1974, 1978, 1996) double-loop learning addresses the problem of ill-suited frames, metaphors that lead to habits and perspective that overlook key design quality aspects. Schön's collaboration with Argyris also centers on professional effectiveness. Professionals (e.g. designers) have mental maps governing their actions in situations. From these maps they (overtly and tacitly), plan, implement, and review their actions. These maps guide actions through intuition rather than any explicitly espoused theories explaining their actions. This often results in a split between theory and action – what people "say" they do and what they actually do. Argyris and Schön (1974, 1978, 1996) characterize two theories of action: one in which personal theories of action are implicit in daily practice (a *theory in use*), and another that is used when our actions are described to others (*espoused theory*). Correctly aligning these two theories in the student learner helps to build a useful repertoire for designing.

Smith (2011) describes the elements driving the development, utilization, and perpetuation of theories of action:

Governing variables: these are dimensions, such as the design qualities, that an individual (such as a designer) is trying to balance and harbor within acceptable and desirable limits. Actions taken are likely to impact these variables in a manner where the designer engages in trade-off behaviors to manage and balance impacts to governing variables.

Action strategies: The general patterns of behavior and action used to maintain acceptable balance among their governing variables.

Consequences: The outcomes, intended and unintended, associated with action.

These theories of action are instructive for educating and developing designers. According to Argyris and Schön (1978), learning is based upon the detection and correction of error. Upon failure (error detection), a designer seeks new action strategies to maintain presumed balances among governing variables. That is, the designer may not challenge given or chosen goals, values, plans and rules (perhaps a premise that an OOP design is *always* superior to a procedural design). This unquestioning behavior is what Argyris and Schön (1974) call "single-loop" learning. An alternative and more desirable learning mode is where the practitioner is open to questioning assumptions about the governing variables themselves. To scrutinize and challenge these assumptions is called "double-loop" learning. A grasp of the design qualities and how they resonate (or fail to resonate) with stakeholder intentions informs assumption challenging and assures that the stakeholder intentions as presented reflect *theory in use* rather than *espoused theory*. An illustration of

ISSN: 2167-1435
v30 n2560

this process, particularly as it relates to organizational learning, is described as follows:

When the error detected and corrected permits the organization to carry on its present policies or achieve its present objectives, then that error-and-correction process is single-loop learning. …Double-loop learning occurs when error is detected and corrected in ways that involve the modification of an organization's underlying norms, policies and objectives (Argyris and Schön, 1978: 2-3).

Single-loop learning would be normative when goals, values, frameworks and, to a significant extent, strategies are taken for granted (Smith, 2011). That is, the designer simply focuses on the efficiency of techniques; those most likely found within the robustness family of design quality (Usher and Bryant: 1989: 87). The shortcoming in single-loop learning is that all reflection is directed toward making existing strategies more effective. With double-loop learning, the naming, framing, and metaphor protocol that underlies repertoire are subject to critical review and open to alternative strategies. Agile software development (and design) methods, XP and Scrum in particular, incorporate double-loop learning through critical review, although this behavior is often under-engaged or reverts to simple single-loop behaviors (Babb, Hoda, & Nørbjerg, 2013).

6. FORMING DESIGN PEDAGOGY

The discussion of reflective action suggests protocols to enhance learning and reflective practice by building repertoire that advance toward mastery. Effective design pedagogy should be based on theory and experience both grounded in quality, immersion, craftsmanship, and lessons from guilds.

Quality

Although the science of management leans toward a positivist inclination of quantification, our experience is that workers or "makers" have qualitative competencies and skills in their repertoire essential to production. Despite automating a significant degree of production, design remains in the context of doing, of making, and of taking action. In information systems, "… [*designers*] have a kind of knowledge that is distinct from the knowledge that managers have…" which informs a profound way of seeing their discipline (Hummel, 1987). Where many stakeholders view the IS discipline as managerial, a design perspective requires that we understand IS as an endeavor of *doing*. It is in this sense that Schön (1983, 1987) provides an empirical perspective on what happens as professionals act. To achieve quality in design the professional must have a theory of quality that guides her decisions. Because of its explicit inclusion of aesthetics, we believe that Thriving Systems Theory is a viable candidate design theory for design pedagogy.

Immersion

Quality, although a worthy aim of standards and regulation, is still quite a subjective affair. Thus, quality pedagogy cannot be realized if unaccompanied by domain knowledge, technical skills, and techniques and their requisite training. These are intrinsic to mastery, as in the performing arts (e.g. music and athletics); one must practice and undertake instruction with some theory of quality as a goal. In a collegiate setting, where basic IS instruction transpires, immersion is a desirable protocol for inculcating both robustness and vitality qualities defined in Thriving Systems Theory.

There are very few contemporary examples of the immersive approach in four year baccalaureate IS programs. The IS program at Brigham Young University in Utah is one – in an AASCB-accredited college of business. Students spend the first two years completing both university core and a pre-business curriculum. In the fall of the junior year, students are placed into 5-student cohort teams and engage in an immersive study of programming, analysis and design, networking, business process analysis, data management, and enterprise architecture. At the conclusion of the fall semester, these teams engage with real-world clients in a design competition. Faculty and industry partners judge the product quality. In the following term the teams implement their designs. These students are well prepared for design-relevant internships during their junior-into-senior summer. Their senior year completes their business core with their foundation in information systems fully formed. Many students move on to an accelerated Master's program. The immersion approach of BYU program facilitates imprinting a pattern of repertoire development on every student and sets them on a firm footing for their continued maturation toward a master designer.

Apprenticeship

Apprenticeship is pedagogy based upon teacher/student relationships that are one on

ISSN: 2167-1435
v30 n2560

one or one on few. The arrangement promotes the immediacy of feedback in the double-loop learning protocol. During the 2004 to 2005 academic year at New Mexico Highlands University an innovative curriculum committed to the apprenticeship model – the Software Development Apprenticeship (SDA) (Rostal & West, 2006). It too was an immersive program modeled on the developmental concept of apprenticeship as a progression toward mastery. The curriculum focused on agile software development methodology. Organized in cadres by experience, those students more advanced in the program provided systematic and formal guidance to novice and apprentice students. The small number of students and faculty allowed the immersive curriculum to focus on self-governance, learning-from-doing and learning-from-learning with mastery as an explicit goal. The program emphasized: 1) a focus on people (and their inherent subjectivity with respect to quality); 2) systems thinking enfolding the stakeholders with the artifact; 3) agility focusing on outcomes; 4) craftsmanship recognizing quality as the goal of design; and 5) a focus on software as the central medium by which systems come to fruition.

The program adhered to the university's curricular framework that presented a challenging environment for the program designers. Yet they crafted a unique experience that inspired students to succeed both as undergraduates and practicing professionals on live contracted client engagements.

Craftsmanship

Apprenticeship underscores the importance of tacit knowing through experience, through exposure, and through a mature repertoire. This is most difficult to achieve in the disjointed, bifurcated curricular designs that are common in today's IS related programs. Both BYU and New Mexico Highlands University illustrate innovative curricular design representing a design-centric view of our discipline. They demonstrate immersion consistent with the traditional and time-honored pedagogy of apprenticeship.

An apprenticeship model sets the student's personal experience in action as the primary source of learning. In their own behavior the instructor/master coaches model double-loop learning as a virtuous (quality focused) pattern of reflective practice. The result is a repertoire of habits and norms grounded in actual problem solving experience using the IS tools, techniques, and skills required of the design craft; then evaluated against professional expectations. Personal experience in *action* operationalizes the combination of quantity and quality as part-and-parcel of practicing craft. This accentuates the perception of technology and tools as implements of design but not substitutes for it. Despite advances in tools that greatly automate and facilitate the process, the act of designing systems is still very much rooted in the conceptual, in the imagination that is often labeled "creativity." As such, regardless of the "industrialization" of the discipline, design remains an endeavor grounded in human touch and craft.

Lessons from Craft Guilds

Craft guilds from medieval times served several useful purposes relevant to the pedagogy of design. Wolek (1999a, 1999b) proposes that pre-industrial craft guilds rested on three foundational principles: 1) regulation; 2) standards of accomplishment; and 3) apprenticeship. Despite any romantic notions of craft and guilds, most guilds served important needs by promoting trade, creating clear quality-driven theories establishing and recognizing mastery, and regulation of practice to promote quality.

Whereas there may have been unique social and historical contexts for the collaborative labor action that were trade/craft guilds, these guilds were effective in developing a solidarity and shared pride in craft that encouraged both innovation and mastery (Wolek, 1999a). Guilds regulated craft such that quality could be defined and improved to promote a professional sense of commerce, quality, resources, product attributes, and process. The positive aspects of guilds were that they established benchmarks for quality and frameworks for improving the craft.

Thriving Systems Theory, as an information systems design theory, adopts the blended emphasis of the objective and aesthetic expression of quality from Alexander (1977, 1979). The qualities and the choice properties that underlie them in TST offer a starting point to study and teach design. The craft and trade guilds teach us that standards can emerge from subjectivity of this sort (Wolek, 1999a).

Among the most important lessons from guilds is the commitment to apprenticeship. The aspects of competence, pride, personal responsibility, and behavioral modeling cannot be overlooked. An apprenticeship represents determined study, tenacious action, focused instruction and correction, and immersion into the tools and ways of craft that lead to both robust and vital design. A master designer does not arrive as such through casual engagement. This focused vision of learning and mastery deserves a presence in design pedagogy.

7. CONCLUSIONS

We argue that design holds a central importance in the discipline of information systems and that importance should be clearly and distinctly reflected in IS pedagogy. The quality experience society deserves from the information systems that are shaping culture, commerce, and lifestyle must address human sensibilities and aesthetics; lessons widely missing from the IS curricula today. Although much of the learning necessary for IS professionals is both theoretical and technical, all of the systems our students produce, augment, or maintain owe most of the quality that stakeholders and users experience in them to enlightened and well-practiced design.

Thriving Systems Theory is a likely candidate for infusing subjective aspects of quality into design pedagogy. The opportunities represented in reflective practice, double-loop learning, and immersive and apprenticeship learning protocols will require the innovative curricular efforts demonstrated by BYU and New Mexico Highlands. These exemplars stand as feasible and effective prototypes.

This paper promulgates a perspective from which both dialog and curriculum design may begin. The IS profession needs a curriculum that facilitates a design-based IS education focused on reflection, inclusive of both aesthetics and function, and promoting a professional progression based on an apprenticeship/craftsmanship model. Our hope is that this discussion will advance development of a curriculum model centered on design with pedagogy preparing students to grow into professional designers who will design systems that thrive.

8. REFERENCES

Alexander, C., Ishikawa, S. & Silverstien, M. (1977). A Pattern Language: Towns, Buildings, Construction. Oxford University Press, London.

Alexander, C. (1979) *The Timeless Way of Building*, Oxford University Press, London.

Alexander, C. (2002). *The Nature of Order An Essay on the Art of Building and the Nature of the Universe:* Book I - The Phenomenon of Life, The Center for Environmental Structure, Berkeley, CA, 2002.

Alter, S. (2008). Defining information systems as work systems: implications for the IS field. *European Journal of Information Systems*, 17(5), 448-469.

Argyris, C., Schön, D.A. (1974). *Theory in Practice: Increasing Professional Effectiveness*. Jossey-Bass, San Francisco

Argyris, C., Schön, D.A. (1978). *Organizational Learning: a Theory of Action Perspective*. Addison-Wesley, Reading, Mass.

Argyris, C., Schön, D.A. 1996. *Organizational Learning II: Theory, Method and Practice*. Addison-Wesley, Reading, Mass.

Babb, J. S., Hoda, R., & Nørbjerg, J. (2013). Barriers to Learning in Agile Software Development Projects. In Baumeister, H., & Weber, B. (Eds.), *Agile Processes in Software Engineering and Extreme Programming*. 1-15. Heidelberg: Springer

Babb, J. S., Keith, M. (2012). Co-Creating Value in Systems Development: A Shift towards Service-Dominant Logic. *Journal of Information Systems Applied Research*, 5(1), 4-15.

Beck K., Beedle M., van Bennekum A., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R.C., Mellor S., Schwaber K., Sutherland J., & Thomas D. (2001). *Manifesto for Agile Software Development*. Retrieved June 26, 2013 from agilemanifesto.org

Benbasat, I., & Zmud, R. W. (2003). The identity crisis within the IS discipline: Defining and communicating the discipline's core properties. *MIS quarterly*, 27(2) 183-194.

Christensen, C. M. (1997), *The innovator's dilemma: when new technologies cause great firms to fail*, Harvard Business School Press, Boston, Massachusetts, USA.

Fowler, M. (2002). *Patterns of Enterprise Application Architecture*, Addison-Wesley, Boston.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Boston.

Gregor, S. & Jones, D. (2007). The Anatomy of a Design Theory. *Journal of the Association for Information Systems*, 8(5), 312-335.

Hevner, A. and Chatterjee, S. (2010). *Design Research in Information Systems: Theory and Practice*, Springer, New York, NY.

Hummel, R. P. (1987). Behind quality management: What workers and a few philosophers have always known and how it adds up to excellence in production. *Organizational Dynamics*, 16(2), 71-78.

Kahneman, D. & Klein, G. Conditions for Intuitive Expertise. *American Psychologist*, 64(4), 515-526.

Lakoff, G., & Johnson, M. (2008). *Metaphors we live by*. University of Chicago press, Chicago.

Medina, R. (2008). Christopher Alexander still matters. http://www.regismedina.com/blog/2008/04/christopher-alexander-still-matters. Accessed June 15, 2013.

Mills, C.W. (1959). *The Sociological Imagination*. Oxford University Press, Oxford.

Naur, P. (1985). Programming as theory building. *Microprocessing and Microprogramming*, 15(5), 253-261.

Quillien, J., Rostal, P., and West, D. (2009). Agile anthropology and Alexander's architecture: an essay in three voices. *Proceedings of the 24th ACM SIGPLAN conference on Object oriented programming systems languages and applications (OOPSLA '09)*. 44(10), 529-546.

Quillien, J. & West, D. (2010). Rubber ducks, nightmares, and unsaturated predicates: proto-scientific schemata are good for agile. *Proceedings of the ACM international conference on Object oriented programming systems languages and applications (OOPSLA '10)*. ACM, New York, NY, USA, 901-917.

Schön, D. A. (1983). *The reflective practitioner: How professionals think in action*. Basic Books, New York.

Schön, D. A. (1987). *Educating the reflective practitioner*. Jossey-Bass, San Francisco.

Schön, D. A. (1993) Generative metaphor: A perspective on problem-setting in social policy. In Ortony A. (Ed.) *Metaphor and Thought*, Cambridge University Press, Cambridge University, 137-163

Smith, M. K. (2011). Donald Schön: learning, reflection and change, *the encyclopedia of informal education*. http://www.infed.org/thinkers/et-schon.htm. Retrieved: 06/20/2013.

Taverniers, M. (2002). *Metaphor and metaphorology: a selective genealogy of philosophical and linguistic conceptions of metaphor from Aristotle to the 1990s (Vol. 1).* Academia Press.

Turner, D. (2007). The secret of Apple design. MIT Technology Review. http://www.technologyreview.com/featuredstory/407782/the-secret-of-apple-design/. Accessed June 15, 2013.

Usher, R., & Bryant, I. (1989). *Adult education as theory, practice and research: The captive triangle*. Routledge, London.

Waguespack, L.J. (2010). *Thriving Systems Theory and Metaphor-Driven Modeling*. Springer-Verlag, London.

Waguespack, L. (2011). Design, The "Straw" Missing From the "Bricks" of IS Curricula. *Information Systems Education Journal,* 9(2) pp 101-108. http://isedj.org/2011-9/ ISSN: 1545-679X. (A preliminary version appears in The Proceedings of ISECON 2010)

Waguespack, L.J. & Schiano, W.T. (2012). SCRUM Project Architecture and Thriving

 ISSN: 2167-1435
v30 n2560

Systems Theory. *45th Hawaii International Conference on System Science (HICSS)*, 4943-4951, 4-7 Jan. 2012

Waguespack, L.J. & Schiano, W.T. (2013). Thriving Systems Theory: An Emergent Information Systems Design Theory. *46th Hawaii International Conference on System Science (HICSS)*, pp.3757,3766, 7-10 Jan. 2013

Walls J.G, Widmeyer, G.R. & El Sawy, O.A. (1992). Building and Information System Design Theory for Vigilant EIS. *Information Systems Research*, 3(1), 36-59.

Walls J.G, Widmeyer, G.R. & El Sawy, O.A. (2004). Assessing Information Systems Design Theory in Perspective: How Useful was our 1992 Initial Rendition? *Journal of Information Technology Theory and Application*, 6(2), 43-58.

Walsham, G. (1993). *Interpreting information systems in organizations*. John Wiley & Sons, Inc.

West, D. and Quillien, J. 2011. Patterns for Story Craft. February 2011. ACM 22 pages.

West, D., Rostal, P., & Gabriel, R. P. (2005, October). Apprenticeship agility in academia. *In Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications,* 371-374.

Wolek, F. W. (1999a). The managerial principles behind guild craftsmanship. *Journal of Management History (Archive)*, 5(7), 401-413.

Wolek, F. W. (1999b). Ye olde total quality management. *Total Quality Management*, 10(7), 1077-108

ISSN: 2167-1435
v30 n2560