Advances in Information Systems Development: From Discipline and Predictability to Agility and Improvisation

Jan Pries-Heje[1] , Richard Baskerville, Bala Ramesh[2] , Linda Levine[3]

[1] Roskilde University, DK
[2] Georgia State University, Atlanta, US
[3] Software Engineering Institute, Carnegie Mellon University, Pittsburgh, US

Abstract: When *information systems development* (ISD) was coined as a term and evolved into a research area we lived in a largely industrial economy. This traditional universe placed high value on discipline and predictability for its own sake. In the 1990s several new trends began to question and challenge the traditional view. Specifically, Internet marketplaces created a new environment for information systems development, and novel approaches such as agile methods emerged. In this paper, we present an analysis of empirical findings showing how new principles and practices have come to exist in a parallel economic universe. The traditional universe persists with its foundation in an industrial economic model; and an alternative universe has become apparent corresponding to a knowledge-based economic model. Our findings suggest that, in the future, knowledge-based activity will continue to gain ground with increased emphasis on agility and improvisation. Moreover, the purpose of discipline will be newly understood to serve an underlying role, which is supportive rather than dominant. .

1. Introduction

Information systems development (ISD) regards systems development processes and products. Systems development typically unfolds in a series of stages such as analysis, design, coding and testing. These stages do not have to be carried out sequentially but can be performed iteratively and in parallel. Often each stage operates with a defined notation and will results in a prescribed artefact, such as a requirements specification or a computer program.

An ISD methodology is a prescribed way of carrying out the development. The description typically includes activities to be performed, artefacts resulting from the activities, plus some principles for organizing the activities and attaching people to perform the activities. An ISD methodology can be aimed at a specific type of development, e.g. database-intensive applications with less than 10 people involved, or it can be specific to a company. However, many ISD methodologies claim to be of generic use.

Very early ISD methodologies were based on practical experiences, i.e. when veteran practitioners simply described how one could develop new IS. But soon things became rather messy. Structured Analysis and Design, for example (cf. Yourdon, 1989), instilled a discipline focusing on functions and data flow between function. Object-Oriented Analysis and Design (cf. Larman, 2005) modeled a system as a group of interacting objects with each object representing an entity of interest in the system being modeled. The lack of theory was also addressed: for example, Andersen et al (1990) builds on a theory that says that ISD consists of nine distinctly different activities, and Highsmith (1999) builds, in part, on the theory of lean thinking (Womack, Jones, & Roos, 1990).

In 1986 the Software Engineering Institute (SEI) in Pittsburgh was asked by the U.S. Air Force in 1986 about a systematic way to evaluate software contractors. The resulting model was called the Capability Maturity Model (CMM). The CMM is a framework characterizing a path with five maturity levels each composed of one or more key process areas. If the organization develops these areas, the software process is known to improve. “Each key process area identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capability” (Paulk, Weber, Curtis, & Chrissis, 1995, p. 32). The perspective behind CMM was that software development is a tumultuous human process (Baskerville & Pries-Heje, 1999). “It entails fast-moving computer technology … [and] skilled, extremely mobile professional workers. These workers must apply creativity and innovation in their development …Without careful management, the process will quickly disintegrate into near-chaos” (Baskerville & Pries-Heje, 1999).

In the 1990s, some doubts about discipline and predictability began to surface In many situations where an ISD method was claimed to be used, it could not be proven by researchers (Bødker & Bansler, 1993). Furthermore, the practicality of ISD methods was questioned altogether. A growing number of studies suggested that the relationship of disciplined methodologies to the practice of information systems development was altogether tenuous (Fitzgerald, 1997, 1998, 2000; Wynekoop & Russo, 1993). It seems that methodology, discipline and predictability had become so dominant in our thinking about ISD that the result was a self-fulfilling hypothesis. For example, one alternative viewpoint situates systems development as "amethodical:" the management and orchestration of systems development without the predefined sequence, control, rationality, or claims to universality implied by much of methodological thinking (Truex, Baskerville, & Travis, 2000).

We have set out to answer the following related research questions: How has ISD advanced? Does ISD for the Internet differ from traditional ISD? What roles do discipline and predictability play?

Below, first, we present our research method. Second, we discuss our findings with an emphasis on the parallel, dual economies that currently coexist in information systems development. . Finally, we consider future trends which suggest an increasing emphasis on improvisational design.

2. Research Method

To answer our research questions we undertook three phases of research.

2.1 Phase one: Interview study

The first phase of our research involved nine detailed case studies of Internet ISD companies in two major U.S. metropolitan areas. The objective of this phase was to understand whether ISD for the Internet differs from traditional ISD. This phase identified the practices used for Internet ISD and explored the role of quality in this fast-cycle development environment (Baskerville, Levine, Pries-Heje, Ramesh, & Slaughter, 2001).

Data collection was carried out using open-ended interviewing. We used a grounded theory methodology to analyze the data (Strauss & Corbin, 1990). This methodology allows the development of a theory of a problem under investigation without prior hypotheses. The chosen grounded theory approach is composed of an alternation between three different coding procedures to analyze the collected data: open, axial and selective coding. At the end of the coding we had identified categories, sub-categories and relationships in the data (see Appendix A).

The findings from phase one identified key factors affecting Internet ISD practice. Although the practices used in Internet ISD can also be observed in traditional ISD, the intensity with which they were applied and the way they were applied together in Internet ISD was distinctive (Ramesh, Baskerville, & Pries-Heje, 2002).

2.2 Phase 2: The Discovery Colloquium on Internet ISD

The second phase involved a one-day Discovery Colloquium that we facilitated on Innovative Practices for Speed and Agility in Internet ISD (Baskerville, Levine, Pries-Heje, Ramesh, & Slaughter, 2003; Levine et al., 2002). The objectives of this phase were to synthesize knowledge on best practices for quality and agility in Internet ISD. Grounded in Kurt Lewin's action research model, such search conferences seek to "bring the whole system in the room" to exchange views and learn from one another. Like other forms of action research, participants share observations, engage in collaborative analysis, and logically test discoveries in an interactive debate among experts. Search conferences often produce data and findings coincidently.

The Colloquium was designed to benefit from and extend the findings from Phase 1. Interviewees from the nine Phase 1 companies, as well as selected experts were invited. Participants included software practitioners from entrepreneurial small companies as well as large "brick and mortar" companies, Internet business strategists and leading ISD experts. ISD methodological perspectives also ranged from adherents of agile ISDs to adherents of more traditional ISDs, representing a broad spectrum of stakeholders.

Participants joined one of several breakout groups dedicated to exploration of a core issue. The groups worked through a process of hypothesis testing, identifying linkages, contradictions, and inter-dependencies among the hypotheses. The groups then delved into underlying assumptions to further build dialogue from generative ideas, and to allow for both divergence and convergence of insights, with maximum cross-fertilization. Principles, promising practices and other dynamics of relevance were identified. The Discovery Colloquium's foundations in action research are reflected in the way data collection, analysis and social action are conflated into one flowing exercise. The researchers were included as participant-observers in the Colloquium. All of the Colloquium participants, operating in various groups, explicated the concepts from their individual practical experiences. The groups then operated analytically on these concepts in a socially situated forum. Both data and analysis emerged from the Colloquium in the mode of action learning.

The analysis in the present paper is based on the discussion of one breakout group focused on evaluating Agile ISD. The objective of this group's discussion was to explore whether there is anything different about Agile ISD methodologies, vis-à-vis traditional ISD methodologies.

As a result of our grounded theory analysis we identified a set of practices. See Appendix A for more detail.

3. Two parallel economies with matching ISD methods

Our findings suggest that IS developers in the 21st century environment adopted a considerably different set of practices from those associated with traditional ISD. Firms have developed new ISD principles and have also adapted old principles in the search for best practices for this new environment. These new principles and practices have not replaced former principles and practices, but rather the two sets now exist in parallel economic universes. The traditional universe continues with its foundation in an industrial economic model while the alternative universe corresponds to a knowledge-based economic model. Moreover, the industrial economic model and the knowledge-based economic model can operate in parallel, often within the same firm, driving different forms of ISD organizations, each with dominant principles and practices. A comparative analysis of these economic models and the empirical results from our studies yields an emerging pattern of alignment. This pattern reflects the recent movement of certain sectors of the ISD community from an industrial economy (sometimes called "Fordist") toward a post-industrial (sometimes called "post-Fordist" or knowledge-based) knowledge economy.

The distinction between industrial and post-industrial economics offers several important ideas that are useful in contrasting agile methods with traditional ISD methods. The rationalized and mechanistic production approaches found in industrial economics do not require a highly skilled workforce, whereas post-industrial economies require a new kind of "shop floor;" one that conflates the factory and research and development, focused on the power of human creativity and knowledge. That is to say, the intellectual power of the workforce is considered a critical factor of production. Post-industrial economics recognizes that information industries have assumed a powerful role among consumer societies (Rustin, 1989).

Knowledge-based economics recognizes new elements and modes of production. Knowledge has eclipsed land, labor and capital as economic factors of production and is now believed to be the prime resource for production. This represents a major shift in economic thinking and raises information technologies to a supreme level, as a tool that embodies the efficiency and effectiveness of the prime economic resource for production.

The knowledge economy requires organizations to deal with increasing complexity in their products and processes, to manage a rising inventory of technical and non-technical knowledge, to enable quick learning processes in response to increasing competition with shorter product life cycles, and to organize work done by a flexible workforce.

3.1 Comparative analysis

For a comparative analysis, we will use the key attributes detailed by Rustin (1989) as indicators of the ideal types of industrial and knowledge modes of production (Figure 1).

Industrial Economies	Knowledge Economies
Low technological innovation	Accelerated innovation
Fixed product lines, long runs	High variety of product, shorter runs
Mass marketing	Market diversification and niche-ing
Steep hierarchy, vertical chains of command	Flat hierarchy, more lateral
Mechanistic organization	Organic organization
Central planning, vertical and horizontal	Autonomous profit centers, network systems,

integration	internal markets
Bureaucracy	Professionalism, entrepreneurialism

Figure 1. Ideal types of production in Industrial (Fordist) and Knowledge (Post-Fordist) Economies (adapted from Rustin, 1989)

Our analysis shows how the principles governing “traditional” ISD align with the key attributes of industrial economies, while the emerging principles governing agile methods align with the key attributes of knowledge economies (Baskerville et al., 2003). Traditional ISD principles are indeed a matter of debate, but one set was rigorously developed using a multi-stage Delphi study involving well-respected ISD researchers and practitioners representing a spectrum of viewpoints, and exemplifies in our view the best attempt to date in defining general principles for ISD (Bourque et al., 2002). It is possible to compare these principles with those that proceeded from the Discovery Colloquium which are known to characterize agile.

Our comparison has three sections: First, the distinctive traditional principles of ISD (those traditional principles with no stated equivalent agile principle); second, the distinctive agile development principles (those agile principles with no stated equivalent traditional principle) and third, the overlapping set of comparable traditional and agile development principles (Figure 2). In each section, we analyze the contrasting economic models that inhabit the comparisons. We then summarize the insights from this analysis.

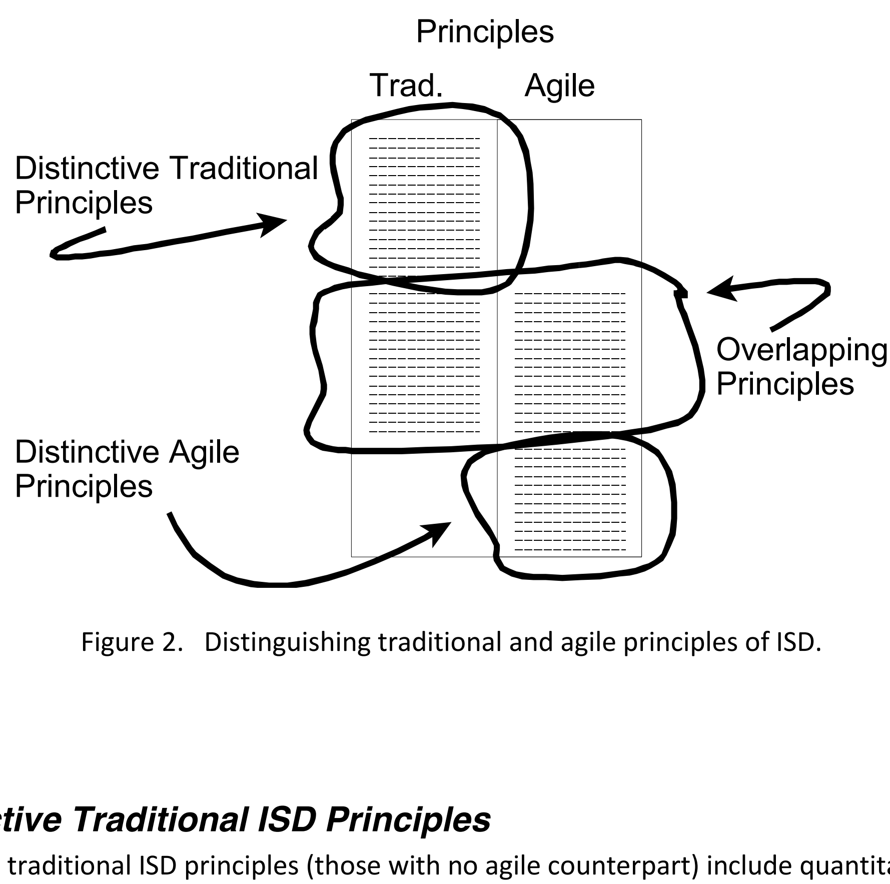

Figure 2. Distinguishing traditional and agile principles of ISD.

3.2 Distinctive Traditional ISD Principles

The distinctive traditional ISD principles (those with no agile counterpart) include quantitative measurement, interchangeable components (reuse), complexity and uncertainty control, rigorous requirements specification, formal quality management, documentation, component coupling, stepwise assembly, and disciplined process. Agile principles may have omitted these traditional ISD principles because they have been inherited from the management of disciplined production processes such as mass production assembly lines. These traditional principles are essential for managing software projects that are large, long, expensive and complex. In contrast, agile developers have scoped their projects using a release approach that keeps the products relatively small, short, cheap and simple. For example one of the Internet software firms we interviewed implemented a new software release every other week. Another firm that was developing a B2B site for transportation told us that customers had asked them to slow down their release speed. Customers simply couldn't cope with major changes every month! The informality of development processes and the small, frequent releases suggest that agile developers operate in a "job shop" environment rather than in a mass production software "factory" (Fernstrom, Narfelt, & Ohlsson, 1992) .

Industrial Economies	Distinctive Traditional Principles
Low technological innovation	Rigorous requirements specification
Fixed product lines, long runs	Stepwise assembly, component coupling
Mass marketing	Interchangeable components (reuse)
Steep hierarchy, vertical chains of command	Complexity and uncertainty control
Mechanistic organization	Quantitative measurement, documentation
Central planning, vertical and horizontal integration, bureaucracy	Disciplined process, formal quality management

Figure 3. Alignments between key features of ideal type industrial economies and distinctive principles of traditional ISD (adapted from Rustin, 1989).

Figure 3 summarizes the evidence of the alignments between distinctive traditional principles and key features of industrial economies. There is a focus on rigor in the requirements specification process rather than creativity in the design process. Stepwise assembly and coupling of components are techniques that echo the factory model of assembly lines. The reuse of interchangeable components adds an emphasis on broader applications for components: akin to the mass markets of industrial models. Complexity and uncertainty control often introduce project management hierarchies. Quantitative measurement and heavy documentation add a mechanistic tone to project organizations. Principles of disciplining processes and managing quality in formal structures can centralize planning and increase the bureaucratic nature of the project management.

3.3 Distinctive Agile ISD Principles

The distinctive agile principles (those with no traditional counterpart) regard teamwork and on-the-fly software process adaptation. Agile principles emphasize informal knowledge exchange, collaboration, and experience as important elements in ISD, and acknowledge more sensitivity to tailoring project practices to environmental conditions. These principles are essential for managing software projects in volatile settings where fast changing technologies and markets drive fast changing skills and knowledge. Agile developers are more aware of, and learn more quickly about shifts in the environments surrounding their software products. In their view, the 'overhead' required to establish formal processes and mechanisms does not justify their use in a fast changing environment. Thus, agile developers operate in a dynamic and evolving environment rather than in a mature and standardized software market. A developer we interviewed in a large software firm emphasized the importance of being able to dynamically adapt software processes, claiming that "When they [management] start implementing rules and procedures, then I start looking for a new place to work." From his perspective, process rigidity and formality would inhibit the informality and dynamism that was needed in this environment.

Knowledge Economies	Distinctive Agile Principles
Accelerated innovation	On-the-fly software process adaptation
High variety of product, shorter runs, organic organization	Tailoring project practices to environmental conditions
Flat hierarchy, more lateral, professionalism, entrepreneurialism	Teamwork, informal knowledge exchange, collaboration, dependence on experience

Figure 4. Alignments between key features of ideal type knowledge economies and distinctive principles of agile ISD.

Figure 4 summarizes evidence of the alignments between distinctive agile principles and key features of knowledge economies. Agility is focused on incremental innovation, and the principles accordingly allow for an acceleration of development process adaptation. Project practices can be tailored according to the situation. A project may be organized in a certain way for a single product, and the next product may be produced in entirely different way according to its context. Agility embraces teamwork and person-to-person exchange of knowledge and values collaboration and professional experience very highly. These values align well with knowledge economy features like professionalism, entrepreneurialism and flat organizations.

3.4 Overlapping Traditional and Agile ISD Principles

The industrial economy did not end when the knowledge-based economy began. These two modes of production continue in parallel. They overlap. Industrial manufacturing continues to deliver low-cost, mass-produced commodities, while knowledge-based manufacturing delivers reasonably-priced, job-lot commodities. It remains possible to produce software systems in either mode.

Our analysis also yields overlapping principles that characterize both traditional and agile ISD. The overlapping principles include: product flexibility, understanding the functional requirements, responding to change, and learning from experience. In our analysis, the overlapping principles are essential for keeping software products aligned with changing requirements. Agile developers have preserved the principles necessary to support shorter project lifecycles and to respond to complex, fast-moving, and competitive marketplaces. Agile developers have to be as flexible and responsive to change as other software engineers. They must also be just as attuned to the functional requirements and prepared to learn from their own experience. Indeed, as suggested by the distinctive agile principles, agile developers will forego formal structures and exercise extreme adaptive options, such as changing their processes on the fly.

3.5 Insights from Comparative Analysis

Overall, we find a palpable alignment between the attributes of a knowledge economy and the principles that distinguish agile methods. That is, agile ISD occurs in a more informal, dynamic, learning environment rather than a mature and standardized software market. Agile methods support shorter project lifecycles and improvisation in order to respond to complex, fast-moving, and competitive marketplaces. For example, agile methodologies reflect a willingness to use multiple approaches and accommodate change to accelerate innovation for a high variety of products. The use of customer engagement drives market diversification and the development of specialized products that support market niches. An emphasis on teamwork and the opportunistic adaptation of methods and practices rely upon the professional expertise of entrepreneurial developers.

However, our analysis suggests that the new environment for ISD does not appear to be supplanting its predecessor. Rather, the application of traditional ISD principles continues in the traditional software application arenas. This co-existence of parallel development universes is most apparent in large ISD firms that have a broad portfolio of projects. The "back-end" elements of systems, based frequently on large-scale mainframes and database engines, continue to be developed in traditional ways while the "front-end" elements of the systems, based frequently on web-based architectures, employ agile development. For examples, see Cook et al. (2002). The development approaches appear to be selected on the basis of the economic environment of the application rather than its technological environment. In some cases, it is impossible to achieve both speed and the necessary economies of scale and quality. Note, however, that when front-end parts of systems are connected to back-end elements, the different development universes must somehow intersect. Some organizations resolve this potential integration problem by applying agile techniques to the back-end to speed up development of their back-end elements, while other organizations apply traditional methods to the front-end to slow down development of their front-end elements. Still others physically separate the development activities and pause projects while the back-end catches up with the front-end.

This continuation of traditional production approaches in parallel with new innovation-driven production approaches is also supported by theories of organizational innovation. To illustrate, in the Minnesota Innovation Research Program (Van de Ven, Angle, & Poole, 1989), fourteen longitudinal field studies of a variety of innovations revealed that innovation does not arise in a linear sequence of steps or stages, or even in a concurrent set of identifiable events or activities. Rather, these emerge as a splintering of organizational production directions in which innovative new activities occur as new pathways of organizational activities that are distinct in direction from the directions represented by older ongoing activities. Figure 5 shows how the innovation process in organizations shifts and splinters production directions over time. The original direction of the organization's activities ("Old") is joined by a new or reoriented set of activities with a new production direction (towards the right upper corner). The innovation that develops into the new direction arises in an organizational shock, such as a sudden change in the marketplace. New or changing activities tend to proliferate driven by multiple ideas and potential new organizational directions. Not all of these new directions prove fruitful, and some terminate after suffering setbacks. Other new directions will merge and integrate, perhaps merging activities with old and new directions. Early in this process, there is a tendency for top management to intervene in pressing for new-idea activities to merge into old-idea directions. (This is a downward pressure in the figure.) That is, management tends to press for the status quo until or unless it recognizes the value in the new production direction. When this recognition happens, top management engagement will intervene instead to restructure the organization and formalize the new strategic directions. Notably, the original ("old") production directions continue in parallel with the new. These original directions may end after a long term with setbacks, but this distant event is not shown in the diagram.

This model of emerging innovation explains how and why agile development might appear to be novel to some while to others, agile development is not new at all − as traditional principles of ISD remain valid even when using agile development methods. Agile development functions in a knowledge-based economy that coexists with the traditional economy; that is, there are parallel economic universes that foster different ISD principles. One economic universe has appeared more recently and has demanded innovation. However, it neither eclipses nor supersedes the older, traditional economic universe that continues to demand standard industrial practices.

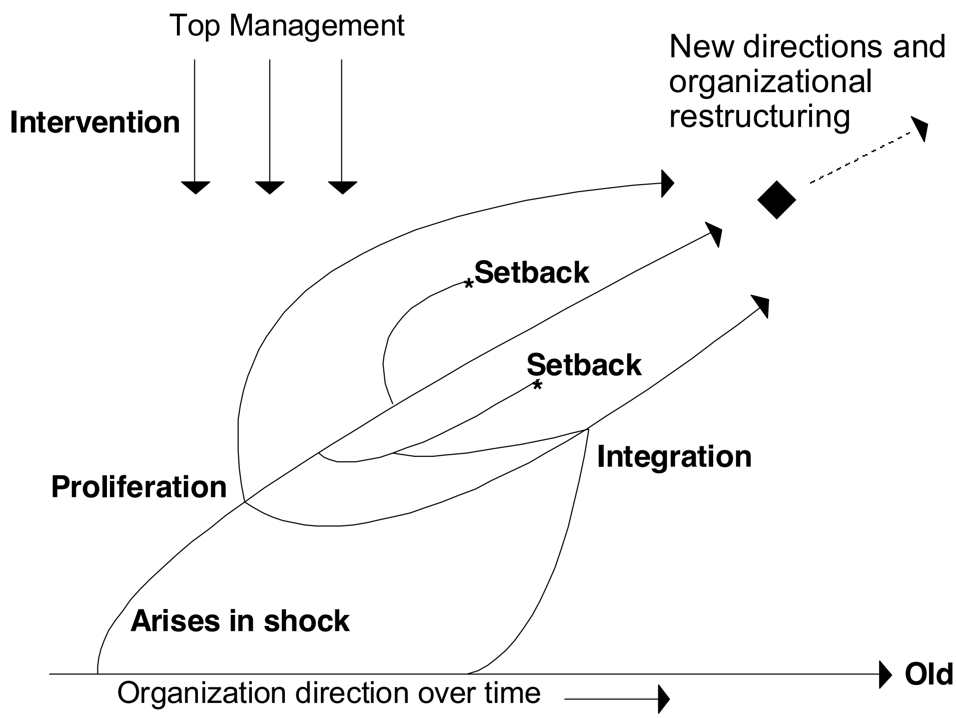

Figure 5. Diagrammatic Elements of the Emerging Innovation Process Model (adapted from Schroeder, Van de Ven, Scudder, & Polley, 1989, p. 131)

Using the notation suggested by the Minnesota studies (Schroeder et al., 1989), Figure 6 illustrates how agile development has emerged in parallel with a new software economy. Macro economic forces are pressing both for the continuation of traditional practices and for the development of a knowledge economy. In response, software markets are demanding both traditional practices and agile practices, based on different, but overlapping sets of principles.

The existence of parallel economic universes explains why agile ISD may not lead to any obsolescence of traditional software. Conventional wisdom would expect that the traditional software practices would gradually decline in favour of the new, "better" and more progressive agile practices. While this rival hypothesis appeals to the proponents of agile development, it is likely that these parallel economic universes will preserve a place in ISD for both sets of practices for the foreseeable future. Driven perhaps by the need for quality, economy of scale, and stability, traditional ISD will remain in demand along with the need for agile development.

We see parallels in the literature on organizational ambidexterity refers to the pursuit of conflicting demands on an organization that require tradeoffs (Gupta, Smith, & Shalley, 2006). The development of appropriate processes and systems for a given context that may achieve the desired balance between opposing demands is increasingly recognized as critical for success of organizations (Gibson & Birkinshaw, 2004). Whereas structural ambidexterity emphasizes separation of concerns, contextual ambidexterity emphasizes dual capacities of individuals and systems. While traditional ISD used in industrial economy focus on stability, rigor, and discipline, agile methods focus on flexibility and responsiveness to change. Therefore, achieving ambidexterity involves reconciling the conflicts between practices in traditional ISD and agile methods. Past research on organizational ambidexterity often considers different types of conflicting demands as competing rather than complementary or orthogonal (Katila & Ahuja, 2002), primarily on the basis of constraints on resources available to fulfill both demands.

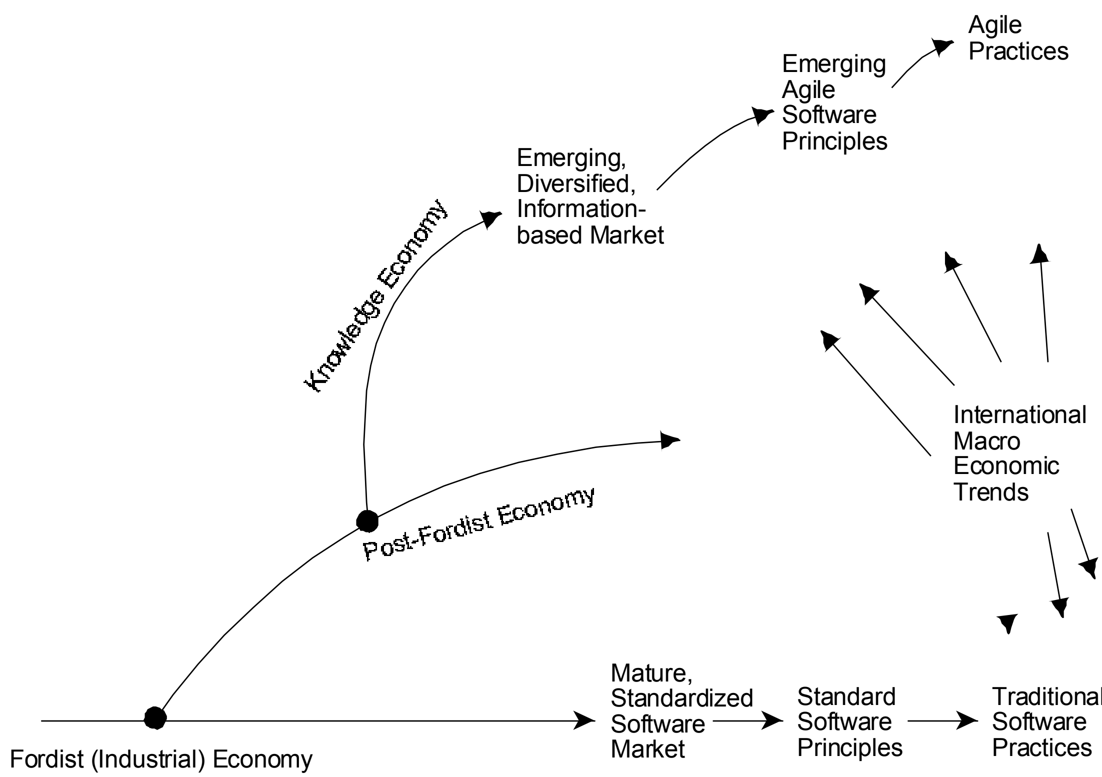

Figure 6. How Agile Methodologies have arisen in Parallel with Traditional Practices.

4. Improvisation: The future challenge

Use of agile methods and agility is consistently associated with software development techniques. But more recently, we have seen fledgling signs of expansion. Ironically, the contracting of the market and the tightening of resources has contributed to an enlarged scope and increased complexity--and a need for balancing at the portfolio and organization levels. This may spur further growth for agile approaches in atypical areas.

That said, the current state for agile methods is still isolated and limited. We have a partial understanding of what agility means for ISD activities. For example, we know that agile methods work well with small teams (especially those that are co-located), where requirements are emergent, and in a turbulent environment of constant change. Agile methods are not recommended in the development of life critical systems; and its use in developing embedded software remains unclear (Ambler, 2004).

We have little understanding of the consequences of agile approaches for technology adoption and implementation activities. Further, within the development and adoption arenas, we have yet to fully grapple with the implications of agility for people, process, and new technology. Our best insights into agility are still achieved through discrete activities—through projects which exist like islands in our organizations. From the development perspective, we have information on different agile methods, where they apply, particular emphases, and some acknowledged limitations. From an adoption perspective, we can speculate that an agile approach would favour pilots, trials, and demonstration projects; and from a knowledge transfer perspective, an agile approach would favour high customer involvement through face-to-face interaction or "body contact."

The challenge for the future is two-part. First, we must optimize the current state with vertical coupling to loosely integrate and propagate agile approaches for development, deployment, and knowledge transfer. This lightweight alignment would allow us to leverage what we know, and to reinforce these otherwise discrete areas of success. Second, and more radically, we must tackle the issue of scaling to investigate options for agile approaches and opportunities that can span organizations. On its face, this might seem contradictory since use of agile methods favours small teams with high contact. But to realize the potential for agile, we must ask how such methods adapt and scale. Perhaps they will do so in entirely new ways.

Austin and Devin (2003) speculate that old production models for ISD are no longer useful. Rather, agile ISD has the potential to be *artful making*. They write: "Artful making (which includes agile software development, theatre rehearsal, some business strategy creation, and much of other knowledge work) is a process for creating form out of disorganized materials. Collaborating artists, using the human brain as their principal technology and ideas as their principal material, work with a very low cost of iteration. They try something and then try it again a different way, constantly re-conceiving ambiguous circumstances and variable materials into coherent and valuable outputs" (pp. xxv-xxvi). Whereas industrial making places a premium on detailed planning, closely specified objectives, processes, and products, artful making is different, fusing iteration and experimentation. Austin and Devin point out that, "if you think and talk about iteration as experimentation, low cost of iteration seems to make business more like science. Its broader effect, though, is to make business more like art" (p. xxv). The authors go on to build an artful framework employing the analogies of theatrical production, extending beyond surface collaboration to the on-cue innovation that theatre companies routinely achieve.

In a similar vein, Stefan Thomke (2003) investigates experimentation in innovation, as it "encompasses success and failure; it is an iterative process of understanding what doesn't work and what does" (p. 2). He reminds us that both results are equally important for learning.

Finally, on a related topic, Dee Hock (1999) has characterized the organization of the 21st century organization as a *chaord*. The term chaord was formed out of combining the first three letters of the word chaos, with the first three letters from the word order. Hock and other leading scientists believe that the primary science of the next century will be the study of complex, self-organizing, nonlinear, adaptive systems, often referred to as complexity theory or chaos theory (DeGeus, 1997; Wheatley, 2001). They assert that living systems arise and thrive on the edge of chaos with just enough order to give them pattern, but not so much to slow their adaptation and learning. This is not unlike the challenge for agility in next-generation systems and organizations. We ask: Does this represent the larger paradigm shift of which agile methods are a part?

Some software development organizations achieve structural ambidexterity through separating traditional and agile development into dual structures and by establishing appropriate communication mechanisms across these structures. However, consistent with past literature on contextual ambidexterity (Gibson & Birkinshaw, 2004), organizations may also select systems and processes that define the context in which conflicting demands of traditional ISD and agile methods can be pursued simultaneously. In essence, contextual ambidexterity suggests behavioral orientation towards mixed capacities rather than focusing on any particular concern. This view is consistent with the our argument on the existence of parallel economic universes and the corresponding approaches to software development.

5. Conclusion

The research questions we phrased were:

1. How has ISD advanced?
2. Does ISD for the Internet differ from traditional ISD?
3. What roles do discipline and predictability play?

Through a discovery colloquium and a grounded theory analysis we found that that new principles and practices for ISD have come to exist in a parallel economic universe. The traditional universe persists with its foundation in an industrial economic model and with a focus on structure, discipline and predictability. But in parallel an alternative universe has become apparent corresponding to a knowledge-based economic model.

Our findings suggest that, in the future, knowledge-based activity will continue to gain ground with increased emphasis on agility and improvisation.

Moreover, the purpose of discipline and predictability will in the future be newly understood to serve an underlying role, which is supportive rather than dominant.

References

Ambler, S. (2004). *The Object Primer* (3rd ed.). Cambridge, England: Cambridge University Press.

Andersen, N., Kensing, F., Lundin, J., Mathiassen, L., Munk-Madsen, A., & Sørgaard, P. (1990). *Professional systems development: experience, ideas and action* Upper Saddle River, NJ, USA: Prentice-Hall.

Austin, R., & Devin, L. (2003). *Artful Making: What Managers Need to Know about How Artists Work*. Upper Saddle River, NJ, USA: Pearson Education Inc. Publishing as Financial Times Prentice Hall.

Baskerville, R., Levine, L., Pries-Heje, J., Ramesh, B., & Slaughter, S. (2001). How Internet Software Companies Negotiate Quality. *IEEE Computer, 34*(5), 51-57.

Baskerville, R., Levine, L., Pries-Heje, J., Ramesh, B., & Slaughter, S. (2003). Is Internet-speed software development different? *IEEE Software, 20*(6), in press.

Baskerville, R., & Pries-Heje, J. (1999). Knowledge Capability and Maturity in Software Management. *ACM SIGMIS The Data Base for Advances in Information Systems, 30*(2).

Bødker, K., & Bansler, J. (1993). A reappraisal of structured analysis: design in an organizational context. *ACM Transactions on Information Systems (TOIS), 11*(2).

Bourque, P., Dupuis, R., Abran, A., Moore, J. W., Tripp, L., & Wolff, S. (2002). Fundamental principles of software engineering - a journey. *The Journal of Systems and Software, 62*, 59-70.

Cook, D. L., Benz, J., Pries-Heje, J., Purao, S., & Wareham, J. (2002). Internet Retailing in the United States. In S. Elliott (Ed.), *Electronic Commerce: B2C Strategies and Models*. (pp. Chapter 7). Chichester: John Wiley & Sons.

DeGeus, A. (1997). *The Living Company*. Boston, MA, USA: Harvard Business School Press.

Fernstrom, C., Narfelt, K.-H., & Ohlsson, L. (1992). Software factory principles, architecture, and experiments. *IEEE Software, 9*(2), 36-44.

Fitzgerald, B. (1997). The use of system development methodologies in practice: A field study. *Information Systems Journal, 7*(2), 201-212.

Fitzgerald, B. (1998). An empirical investigation into the adoption of systems development methodologies. *Information and Management,*(34), 317-328.

Fitzgerald, B. (2000). Systems development methodologies: The problem of tenses. . *Information Technology and People, 13*(2), 13-22.

Gibson, C. B., & Birkinshaw, J. (2004). The Antecedents, Consequences, and Mediating Role of Organizational Ambidexterity. *Academy of Management Journal, 47*(2), 209.

Gupta, A. K., Smith, K. G., & Shalley, C. E. (2006). The Interplay between Exploration and Exploitation. *Academy of Management Journal, 49*(4), 693-706.

Highsmith, J. (1999). *Adaptive Software Development*. New York: Dorest House.

Hock, D. (1999). *Birth of the Chaordic Age*. San Francisco, CA, USA: Berrett-Koehler Publishers.

Katila, R., & Ahuja, G. (2002). Something Old, Something New: A Longitudinal Study of Search Behavior and New Product Introduction. *Academy of Management Journal, 45*(6), 1183-1194.

Larman, C. (2005). Applying UML and Patterns - Introduction to OOA/D & Iterative Development. In (3rd ed.): Prentice Hall.

Levine, L., Baskerville, R., Loveland Link, J. L., Pries-Heje, J., Ramesh, B., & Slaughter, S. (2002). *Discovery colloquium: Quality software development @ Internet speed* (No. SEI

Technical Report CMU/SEI-2002-TR-020, ESC-TR-2002-020). Pittsburgh, PA: Software Engineering Institute.o. Document Number)

Paulk, M. C., Weber, C., Curtis, B., & Chrissis, M. B. (1995). *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, Mass.: Addison-Wesley.

Ramesh, B., Baskerville, R., & Pries-Heje, J. (2002). Internet Software Engineering : A different class of Processes. *Annals of Software Engineering, 14*(1-4), 169-195.

Rustin, M. (1989). The politics of Post-Fordism: or, the trouble with 'New Times'. *New Left Review, 175*(1), 54-77.

Schroeder, R. G., Van de Ven, A. H., Scudder, G. D., & Polley, D. (1989). The development of innovation ideas. In A. H. Van de Ven, H. L. Angle & M. S. Poole (Eds.), *Research on The Management of Innovation: The Minnesota Studies* (pp. 107-134). new York: Harper & Row.

Strauss, A., & Corbin, J. (1990). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Beverly Hills, CA, USA: Sage Publications.

Thomke, S. (2003). *Experimentation Matters: Unlocking the Potential on New Technologies for Innovation*. Boston, Massachusetts, USA: Harvard Business School Press.

Truex, D., Baskerville, R., & Travis, J. (2000). Amethodical Systems Development: The Deferred Meaning of Systems Development Methods. *Accounting, Management and Information Technology*(10), 539.

Van de Ven, A. H., Angle, H. L., & Poole, M. S. (1989). *Research on The Management of Innovation: The Minnesota Studies*. New York: Harper & Row.

Wheatley, M. J. (2001). *Leadership and the New Science: Discovering Order in a Chaotic World (revised ed.)*. San Francisco, CA, USA Berrett-Koehler Publishers.

Womack, J. P., Jones, D. T., & Roos, D. (1990). *The Machine That Changed The World*. New York: Macmillan Publishing.

Wynekoop, J., & Russo, N. (1993). *System Development Methodologies: Unanswered Questions and the Research-Practice Gap*. Paper presented at the Proceedings of the 14th International Conference Information Systems (ICIS), Orlando, USA.

Yourdon, E. (1989). Modern Structured Analysis. Upper Saddle River, Nj, USA: Yourdon Press.

Appendix A: Grounded theory approach: Three coding procedures

The goal of *open coding* is to reveal the essential ideas found in the data. Open coding involves two tasks: labeling phenomena and categorizing. Labeling discriminates concepts in the data. Each discrete incident or idea receives a representative name or label. These names represent a concept inherent in the observation. Categorizing is the process of grouping related concepts and themes under joint headings.

Axial coding involves two tasks further developing the categories and properties. The first task connects categories in a sequence of relationships. For example, a causal condition or a consequence can connect two categories. The second task is validation of the relationships in the data, yielding the discovery and specification of the differences and similarities among and within the categories. This discovery adds variation and depth of understanding.

Selective coding is the process of determining a core category or story line that explains the categories with minimal contradictions. The core category should be related to most or all other categories, and these relationships must be validated and elaborated. Once settled (called “saturated” in grounded theory methodology), the core category explains all of the other categories.

The grounded theory analysis resulted in the identification of core categories or concepts and relationships among them that explain how and why Internet-speed ISD is different from traditional approaches.