Submitted to *Advances in Computers*, vol. 59, 2003.
Please do not duplicate or distribute.
Version 1.2 – 22 Apr 2003

Why new software processes are not adopted

by Stan Rifkin
MASTER SYSTEMS INC.
2604B El Camino Real
Carlsbad, California 92008 USA
✆+1 760 729 3388 sr@Master-Systems.com

Abstract. Why do we often appear not to do what is best for us, or at least what someone else thinks is best? To what extent do the reasons have to do with what is being suggested vs. to how the implementation is planned and executed? Is there a way to accelerate the rate at which the implementation of process adoption can be achieved? These questions are addressed by reviewing the considerable literature on implementations of software engineering, information systems, and quality improvement.

Table of contents

List of Figures

In general, this chapter is a literature review of how to accelerate adoption of software engineering-related processes by software managers and engineers. It adds value by acting as a lens to help make sense of the numerous field studies on the subject (175 references are cited).

1 Change is harder than we think

"Not much has changed in a system that failed: The F.B.I. and C.I.A. missed signals a year ago. Now they do well in capital turf wars." [1] So reads a recent headline related to change one year after the September 11th, 2001, attack on the World Trade Center. If ever there were motivation to change disaster has to be it.

Or does it? One way that the history of surgery is divided is pre- and post-Listerism.[1] Joseph Lister, also the namesake of Listerine-brand mouth wash, was the inventor of antiseptic surgery in the 1850s in Scotland. In the wars at the time there were more injuries from field surgery than from battle! Surgery then was considered a form of butchery because of the pain (there was no anesthesia) and the near certain death from infection. Lister missed many cues on his way to discover that cleaning the wound – and his hands, uniform, and instruments -- before, during and after surgery dramatically reduced mortality. But even after he demonstrated this dramatic decrease in mortality (from forty percent in one ward to two percent) antiseptic surgery was not adopted in England and the United States (it was adopted in Germany, where it saved many, many lives). An analysis of the diffusion of antiseptic surgery lists nine factors[2] that impeded its adoption. [2] Oddly, at about the same time, anesthesia during surgery was invented and promptly adopted in England and the U.S.

"Turf" referred to in the F.B.I. and C.I.A. headline, above, also was one of the factors in the impediments to adopting antiseptic surgery. As will be explored below, new processes often mean a change in power and a change in power can mean a change in the ability to pay the mortgage. What we ordinarily call resistance may be nothing more pernicious than protecting our ability to pay our mortgages.

"This failure to sustain [improvement processes] recurs again and again despite substantial resources committed to the change effort (many are bankrolled by top management), talented and committed people 'driving the change,' and high stakes. [T]he sources of these problems cannot be remedied by more expert advice, better consultants, or more committed managers. The sources lie in our most basic ways of thinking." (p. 6) [3]

In today's world of pressure to deliver software in a very short time, we barely have time to develop the software that is functionally required, so how would we have time to learn and become competent at a new process? This question is often produced when such "grand" improvements as the Software Engineering Institute's Capability Maturity Models are suggested, or the Experience Factory, or ISO 9000. Yet, when eXtreme programming or agile methods or the Rational Unified Process are suggested there appears to be a receptiveness absent to the grand processes. The explanation, suggested below (Section 9.1), is that the grand methods are not sufficiently divisible, even though they have a bunch of little moving parts. One has to adopt a totality of the

---

1 I am indebted to Watts Humphrey for this example.

2 Medical administration, social interpretation, professional tradition, national competition, theoretical orientation, experimental investigation, technical evaluation, surgical demonstration, and final assimilation.

processes in those grand methods in order to achieve certification or the promised benefits, but with the smaller methods the benefits appear to be incremental and closely follow the implementation of any one of the sub-practices (such as pair programming and refactoring in eXtreme programming).

We all adopt new practices for reasons, presumably rational ones. One observes that there is almost no empirical evidence for the most widely-adopted software engineering processes, which include structured programming, abstract data types, object-oriented design and programming, CASE tools, statistical process control, fourth-generation languages, and formal methods. [4] This applies as well to the more modern adaptations, such as eXtreme programming and agile methods (though there is some empirical evidence for pair programming [5]).[3]

Another observation would not be so flattering about the decision to adopt. In the most widely cited business journal article [6], the authors explain how they sought to understand how managers made decisions, which rational models did they apply to weigh the multiple factors that would need to be taken into account. Instead of ration (even bounded rationality) they found a garbage can: managers are presented solutions and problems asynchronously. Solutions are matched to problems when the presentation of the problem is proximate in time with (especially just after) the presentation of the solution: they are taken out of the garbage can when a match is made with a problem.

Whether there is a good reason to change can be seen in either light (no empirical evidence or the garbage can), neither of which is potentially very compelling.

2 The Answers

Alas, our field is known for its impatience. In that spirit, I want to sum up the whole chapter with the best two descriptions of adoption. That is, I want to present The Answers. Each is more or less from a single reference; it is always dangerous to rely on a single source, and the justifications after the explanation of the models explore why that particular article is seminal.

The basis of selection for these two answers is over-simple: They elegantly explain a great deal of otherwise monolithic approaches, such as factor studies that try to identify and isolate the controlling influences on adoption. The two answers below are more dynamic and identify that certain factors are more influential during certain epochs or under certain conditions and not at other

---

[3] In all fairness, the journal *Empirical Software Engineering* (ISSN 1382-3256) has been created to fulfill the need for empirical evidence regarding software technologies. There is no evidence that the Journal is consulted by decision-makers, nor is there evidence that decision-makers consult <u>any</u> substantiation of advantage of the technology under consideration.

times/conditions. Such a contingency style ("What is critical for adoption?" "It depends!") reveals far more than any set of single factors that are linearly aligned in an inexorable (or unstated) time sequence. Also, both answers leave plenty of room for human forces, technical details, and organizational/environmental influences, all of which are part of the rich reality of implementing software engineering processes.

2.1 *The first model*

The model is taken from Repenning [7]. The explanation of process adoption relies on Figure 1, below. The grammar of the diagram was first popularized in Senge [8], where it is called a causal loop diagram. The intuition is that there are three forces that determine whether a new process will be used in practice: normative pressure, reinforcement, and diffusion.

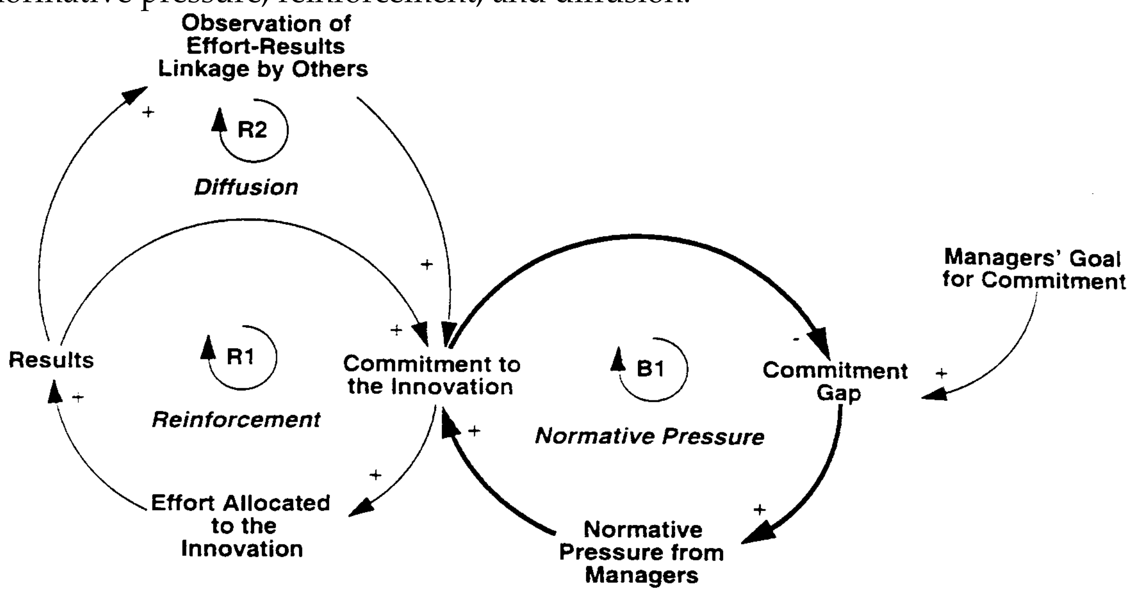

Figure 1. Arrangement of the dynamic forces of implementation. (from [7], pp. 109-127. Reprinted by permission of the Institute for Operations Research and the Management Sciences (INFORMS))

- Normative pressure is that exerted by management to meet expectations, to achieve norms. Managers set goals for commitment to implement the innovation (in this case, process improvement). If the gap between the managers' goal and the current commitment is large enough, then the pressure on those affected is increased to raise their commitment to implement.
- Reinforcement is the process by which the pressure to increase commitment is translated into effort. In this model there is a direct

relationship between effort and results, so as effort is increased then positive results are, too.
- Diffusion is something of the flywheel effect in which those affected observe improved results so they, in turn, increase their commitment to implement the improvement innovation.

The explanation -- composed of the (necessarily) linear arrangement of words, sentences, and paragraphs -- gives the appearance that managers' normative intentions might begin the whole process, and then the flow proceeds in the manner described above for the first time through. After that, things can get interesting. For example, Repenning (p. 120) described an instance where the diffusion loop damps the commitment to implement when the results appear to be disproportionately low with respect to the effort allocated.

The simulation model in the title of Repenning's article illustrates the interaction among the three forces. Essentially, the two loops with the R1 and R2 labels tend to amplify effects, because there are + marks all the way around each loop; the one marked B1, where *B* stands for balancing, because it has an odd number of – marks [9], can reduce future commitment as the gap between actual commitment and the managers' goals closes.

Now we can see the ups and downs of implementation:

- When the managers' goals for commitment are not sufficiently different from the current commitment then there will be insufficient pressure to commit to going forward.
- Whenever the effort is (too) low, then the results will be low and the commitment will decrease in a vicious cycle.
- Whenever the effort-results linkage observed is (too) low, then others will not be inspired to commit and the effort allocated will be decreased, decreasing the results still more, in a vicious cycle.

Repenning was able to reproduce in his model the situation in which managers set appropriate goals, allocate sufficient effort and then underestimate the delay needed to achieve results, so the commitment is eroded and the results fall off because of the connections among the goal, commitment, effort, and results. With another set of values, Repenning showed that once the flywheel effect of diffusion is in place, due to the long-term positive relationship between effort and results, then normative pressure does not play such an important role, can be removed, and the implementation continues its virtuous cycle.

At the end of the article, Repenning gives advice to managers facing the task of implementation:

1. Do not prepare to implement something new until and unless those who control resources become "fully committed to the effort and patient in the

months between adopting" and to having the results motivate further deployment.

2. While seeking to have the results themselves stimulate the flywheel effect, do not do this at all costs. Such a Herculean effort would be seen by future adopters as consuming an effort disproportionate to the results, so that the virtuous cycle would not happen.

The first bit of advice is important because so many authors implore their readers to frame the process improvement implementation as a project, rather like a software project. This would miss the point that planning a software project is by and large a solved problem, while planning human changes, especially by engineers and engineering managers, is not. Accordingly, Repenning's advice can be seen as a case perhaps for *planning* a process improvement as a project, but then do not *implement* it as a project, as it is too difficult to estimate the relationships among the variables.[4]

2.2 Advantages of the first model

There are several reasons that Repenning is a superior source on understanding why new processes are not adopted:

- It has face validity, that is, it tracks what we already know by personal, idiosyncratic experience, and by the experience of others (to be detailed below as part of the literature review)
- It pulls in the characteristics we customarily, perhaps cursorily, associate with implementation success, such as leadership (setting norms and sticking with them), managing change (how improvement is communicated, as in the effort-results link), allocating sufficient resources (effort in this case), rewards, and the need to begin improvement with sufficient energy.
- It takes into account many forces, not just a single one.
- Those forces are arranged in a simple structure that can have a complex, non-linear interaction. Causes may become effects, there can be competition among the forces or they can align, and, therefore, not only success can be explained but so can failure. And the possible ups and downs are illustrated by the model.
- It describes both a process and factors.
- It depends upon and sums up considerable theory. It is not just one person's bright idea.

[4] Mark Paulk frames it differently. Some software projects are planned as discovery activities, iteratively reducing equivocality in the problem, solution, and/or project spaces. Implementation can gainfully be planned and performed this way, in planned cycles that iteratively identify and reduce risk. (Personal communication.)

- Without the insight gained by using the model we are unlikely to succeed on intuition alone.

We will visit and re-visit these desiderata in the course of reviewing the history of what is known about getting best software engineering practices into actual practice.

2.3 The second model

In her article, Markus [10] guides us through the "home grounds" of the two most prevalent arguments about why process innovations are not adopted: either the process (or system of processes) itself is flawed in some technical respect (e.g., hard to use) [11], or the intended targets of the improvement (we humans) have some inherent reason to resist the implementation [12]. That is, there is a system-determined answer and a people-determined answer; the result in both cases is resistance. It is, therefore, the role of the implementer to either restructure the technical aspects of the system or restructure the people aspects (rewards, incentives, span of control, new job titles).

Markus notes that we see this dichotomy in solutions: some solutions address purely technical aspects, such as user involvement in the requirements and design phases, and others address how humans change in response to new processes trying to be introduced. She proposes a third theory, interaction, that does not rely on the assumptions of the other two. There are two variants of interaction theory:

1. Sociotechnical: it's all one system, and every part interacts with the others [13-15].
2. Political: it's about power, who has it, and who loses and gains with the introduction of the new stuff.

Markus frames her insights in terms of resistance:

	People-Determined	System-Determined	Interaction Theory
Cause of resistance	Factors internal to people and groups Cognitive style Personality traits Human nature	System factors such as technical excellence and ergonomics Lack of user-friendliness Poor human factors Inadequate technical design or implementation	Interaction of system and context of use *Sociotechnical variant*: Interaction of system with division labor *Political variant*: Interaction of system with distribution of intra-organizational power
Assumptions about purposes of information systems	Purposes of systems are consistent with Rational Theory of Management, can be excluded from further consideration	Purposes of systems are consistent with Rational Theory of Management, can be excluded from further consideration	*Sociotechnical variant*: Systems may have the purpose to change organizational culture, not just workflow *Political variant*: Systems may be intended to change the balance of power
Assumptions about organizations	Organizational goals shared by all participants	Organizational goals shared by all participants	*Sociotechnical variant*: Goals conditioned by history *Political variant*: Goals differ by organizational location; conflict is endemic
Assumptions about resistance	Resistance is attribute of the intended system user; undesirable behavior	Resistance is attribute of the intended system user; undesirable behavior	Resistance is a product of the setting, users, and designers; neither desirable nor undesirable

Table 1. Theories of resistance: underlying assumptions. (from [10], pp. 430-444. (c) 1983 ACM, Inc. Reprinted by permission.)

Like any good theory, these three can be used to predict where to look for problems and solutions:

	People-Determined	System-Determined	Interaction Theory (Political Variant)
Facts needed in real-world case for theory to be applicable	System is resisted, resistors differ from nonresistors on certain personal dimensions	System is resisted, system has technical problems	System is resisted, resistance occurs in the context of political struggles
Predictions derived from theories	Change the people involved, resistance will disappear Job rotation among resistors and nonresistors	Fix technical problems, resistance will disappear Improve system efficiency Improve data entry	Changing individuals and/or fixing technical features will have little effect on resistance Resistance will persist in spite of time, rotation, and technical improvements Interaction theory can explain other relevant organizational phenomena in addition to resistance

Table 2. Theories of resistance: predictions. (from[10], pp. 430-444. (c) 1983 ACM, Inc. Reprinted by permission.)

What she finds, and asks us readers to look closely at our own situations for, is that (even) when people- and system-determined problems are addressed and solved, "resistance" remains, but when interaction with the organizational context or power distribution is addressed, then the "resistance" goes away. Accordingly, interaction theory is a better guide for implementation.

Looking at interaction instead of people or systems implies that a certain kind of information is used as evidence of implementation. That kind of information is not usually valued by us engineers or business people. The logic of using this kind of evidence begins with a worldview or ontology. Ontologies are basic beliefs about how the world works. One example is positivism, which believes that there is an enduring reality that exists independent of our sensing or perception of it. When we turn our backs on a mountain it is still there! Another example is that the world is socially-constructed, i.e., that we make sense of what we perceive based on how society instructs us to. Each of these two examples also implies epistemology and methodology, that is, what can be known for sure and what methods generate such knowledge. Positivism, sometimes called "normal science," believes in "hard" facts – that is, quantitative measurements – obtained in such a way that the measurements can be obtained by anyone else equipped with the same instruments. Interpretivism, which corresponds to the social construction of reality, seeks to find the patterns that operate in social settings, the collections of phenomena that seem to fit together. In the interpretivist paradigm it is acceptable that the search for those patterns is in a social setting that cannot be repeated, because the environment is not controlled or even controllable, as in a test tube laboratory. Objectivity in this paradigm cannot be obtained. The methods are generally called qualitative [16-20].

The interaction framework espoused by Markus means leaving the methods of normal science (and engineering and commerce) in favor of interpretation, a form of subjective judgment. If we accept the invitation to take into account new kinds of information (namely subjective sources) then we may see things we did not before. But, it is difficult to let go what we think we can know for sure in exchange for learning more about the situation from less of an absolute perspective.

It is worth mentioning that one of the objections of normal science is that social scientists "make up" constructs, such as morale, intelligence, and power, that those constructs do not have an existence independent of their definitions. Abraham [21], a recovering physicist, has argued persuasively that the constructs of classical physics, such as distance, acceleration, and force, to mention but a few, are no less "made up" and do not exist independent of our thoughts about them. That we ascribe measurements to distance, acceleration, and force reify them precisely to the extent that measurements of morale, intelligence, and power do.

One of the popular ways to express that the social construction of reality acts as filter on what we see is the often-cited quip quoted by Karl Weick [22], p. 1. It refers to American baseball, where a ball is thrown (pitched) towards a batter. If the batter does not swing, then a judge (an umpire) calls either "ball" if the trajectory was outside a mythical box between the shoulders of the batter and his knees, or "strike" if it was inside that box. Three umpires were talking. The first said, "I calls them as they is." The second said, "I calls them as I sees them." The third and cleverest umpire said, "They ain't nothin' till I calls them." Later Weick avers that when people say "I'll believe it when I see it," they more likely mean "I'll see it when I believe it." And, quoting another source, "man is an animal suspended in webs of significance he himself has spun." (pp. 134-135)

2.4 Advantages of the second model

Like the first model, this one incorporates other theories [23], so it is not (just) one person's bright idea. It also addresses competing theories that are likely the most prevalent in the implementation literature and practice, so the insights are novel and useful. It also predicts the problems and solutions better than the other two theories. In addition, "resistance" is redefined as natural and a part of any change, not something to be conquered and overcome. And last, it invites us to broaden our computer science-, software engineering-centric methods for observing and gathering information, something that many implementers feel is necessary to be successful, that somehow trying harder with what we already know how to do is not more effective. [24]

3 Beginning the inquiry

We begin by delimiting the scope of our inquiry. We examine definitions of adoption, phases of getting processes into practice, which fields might best contribute to our understanding, and what we mean by software processes. Then we examine the sources of insight one subject at a time. The conclusion is brief, as The Answers have already been presented above.

3.1 Definition of adoption

There are many synonyms, such as technology transfer, technology transition, technology infusion, diffusion, dissemination, deployment, assimilation, and implementation. In the sense we use them we mean that some practice or process or procedure is in regular, normal use by those intended as targets of usage. The focus is on usage on the job, the actual practice of a process.

3.2 Framework for inquiry

Lucas et al. [25] suggest a framework for reviewing what is known about implementation: theory, process, and factor; see also Kwon and Zmud [26]. To this we add the single category of case studies and personal (idiosyncratic) experiences, usually expressed as a narrative. Essentially, theory represents the accumulation of empirical evidence of patterns. It is our position that this is the highest form of knowledge because it sums so many observations. At the other end of the knowledge spectrum, it is our positions that case studies and personal experiences represent the least knowledge because they are points about which we must infer the salient factors for our own purposes.

In between theory and stories there are studies of the process or steps and the characteristics that imply success or failure, that is, the factors that bear on the outcome of implementation. There are naturally many studies that cross over, such as the theory of the process of implementation.

3.3 Fields touched by implementation research

Why and how some implementations of processes are successful and others are not can be seen from many perspectives. Among those surveyed here, however briefly, are:

- Innovation – This is the creation of the new process. Many believe that taking implementation concerns into account during innovation increases the likelihood of adoption. Therefore, some scrutiny of the innovation process is common. In addition, there is belief that one of the important predictors of implementation success is an "innovative" atmosphere, one that is receptive to new ideas. This is also the place for path dependence, the notion that in order for certain innovations to be successful there must have been a path or trajectory of prerequisite occurrences.
- Managing human change and organizational culture – This is usually couched in terms of identifying and countering "resistance," though sometimes one can only infer that "resistance" is being addressed because it is not explicitly stated. There are many commercially available change management approaches and programs, all of which subscribe to the people-oriented theory in the Markus article, above. The notion is that some organizational cultures (for example, the unspoken rules) are more conducive to adoption than others. Qualitative organizational learning is in this category because it tries to leverage human change lessons learned (actually lessons observed).
- Leadership and management – It is received wisdom that change is accelerated when it is sponsored by the leader of the organization. Accordingly, this field is also based on the people-oriented theory, and can be part of the process description (that is, exactly what does leadership <u>do</u>) or part of the factors (strength of leader's support). Leadership is commonly the subject of idiosyncratic stories ("Here is how I led change"), perhaps because leadership is often thought to be determinative of the outcome, and a common trait of failure: lack of upper management support (whatever that might mean) augurs for an unsuccessful outcome.
- Social construction of reality – All engineering takes place in a context, much of which is socially constructed. Therefore, it is useful to view adoption as an activity situated in a social system. Indeed, such a view is indispensable because it enables the planner of an implementation to take into account the human and collective aspects in addition to the technical or engineering aspects. The difference between this area and the one on managing human change is the unit of analysis. The unit in this area is a

team, group, division, or other collective. The unit in the human change area is the individual, be it a leader, manager, champion, agent, or target.

- Social shaping of technology – Technology is not a value-free, neutral conduit through which new ideas flow. Rather, the technical aspects of innovations themselves are the results of choices, sometimes on a scale that is impacted by national policy, habits, culture, and economics. Sometimes, for example, the technology represents a dialectic between labor and management, as when it is applied to the de-skilling of workers.
- Mathematical modeling – This is normally applied to the view of adoption as diffusion, something like a contagion or bacterium spreading in a finite medium. This also includes quantitative learning curves, the steady improvement with practice.

3.4 Ambit of software processes

The scope of this chapter is software engineering processes and software development management processes. These processes include the type that are standardized by international standards organizations, such as ISO 9000, and standardized by governments, such as military standards, federal civilian agency standards (e.g., U.S. National Aeronautics and Space Administration, U.S. Federal Aviation Administration). They also include de facto standards, such as the Software Engineering Institute's Capability Maturity Model, Bootstrap, and other normative process standards. In addition, software engineering processes include computer-aided software engineering (CASE), about which much has been written with regard to implementation, adoption, deployment, and their many synonyms.

While we focus on software processes, we do not confine our inquiry solely to them. We borrow where appropriate from other engineering and business disciplines, including adopting new products. While products and processes have several important differences, primary among them the ability to observe adoption, we borrow from the understanding of product adoption when it helps us understand process adoption. See, for example, [27].

4 Process descriptions of implementation

4.1 Description of stage or phase models

Adoption can be viewed as one phase or stage in a sequence of events. Here are several descriptions of the stream:

Redwine et al. [28] use:

1. Concept formulation, the emergence of the key idea
2. Development and extension, usually via a seminal paper or demonstration system

3. Enhancement and exploration (internal), in which usable capabilities are available
4. Enhancement and exploration (external), which shifts usage outside of the development group
5. Popularization, substantial evidence of value and application, such as 40-70% usage.

Maier [29], in Figure 2, below, relying on Schumpeter [30], uses three stages:

1. Invention, when new products or processes are developed
2. Innovation, when the products or processes are introduced in the market
3. Imitation or diffusion, when they are spread

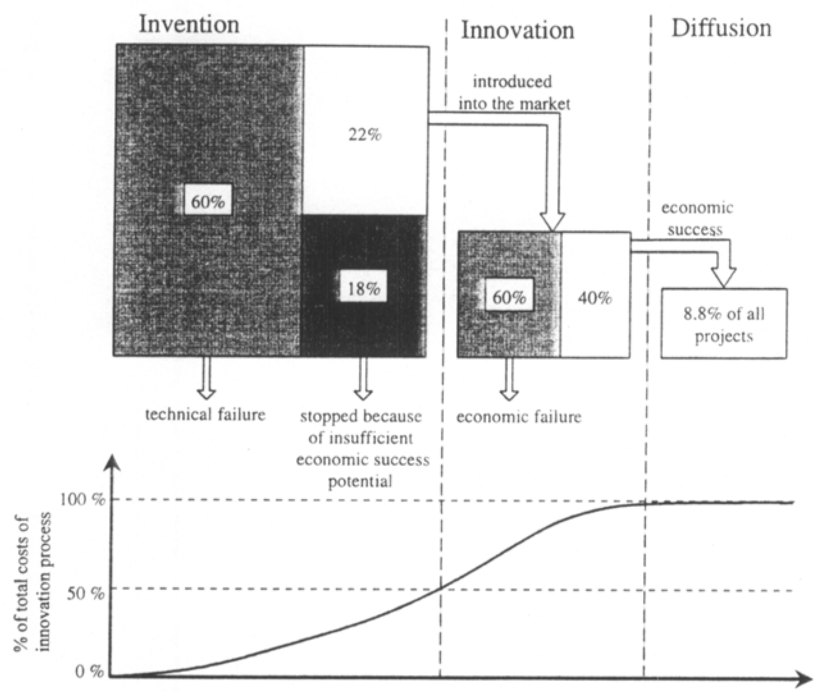

Figure 2. Cascading outcome of innovation activity (from [29], p. 286. © John Wiley & Sons Limited. Reproduced with permission)

The Software Engineering Institute has created the IDEAL model [31] to sequence the phases:

1. **I**nitiating, a discovery activity, looking for motivation and alternatives
2. **D**iagnosing, performing an appraisal of the baseline
3. **E**stablishing, setting goals and planning
4. **A**cting, actually introducing the new process (in this case)
5. **L**everaging, observing lessons and trying to feed them back into the next improvement cycle.

Caputo uses a framework that has grown through oral repetition [32-33]:

1. Contact with the new idea

2. Awareness of the technical merits and its possible impact
3. Understanding what it could mean in this organization
4. Definition of the new process or how the new product will be used
5. Installation of the new product or process, evaluate first instances
6. Adoption, requiring regular usage
7. Institutionalization, during which the practice becomes the normal way
8. Internalization, when one can no longer remember doing it any other way

4.2 Duration of stage or phase models

Two studies have measured the duration for software engineering process innovations to transit roughly from awareness to regular usage. Redwine and his colleagues found the duration to be 15 to 25 years across an industry [28], and Zelkowitz found it took four to five years within a single organization [27]. One other study, of Hewlett Packard's adoption of the formal software inspection process, suggests it can take ten years to reach a 25% adoption level. [34]

4.3 Non-linear (messy) models

Leonard-Barton, [35] in a model that augurs the future, proposes a messy process of mutual adaptation, where the technology to be adopted is modified as it is assimilated and the organization transforms, too, as the technology is assimilated. Each – the technology and the organization – accommodate to each other. Her figure (Figure 3) clearly indicates that this process is not algorithmic, not linear, not even predictable except at its highest level of granularity.

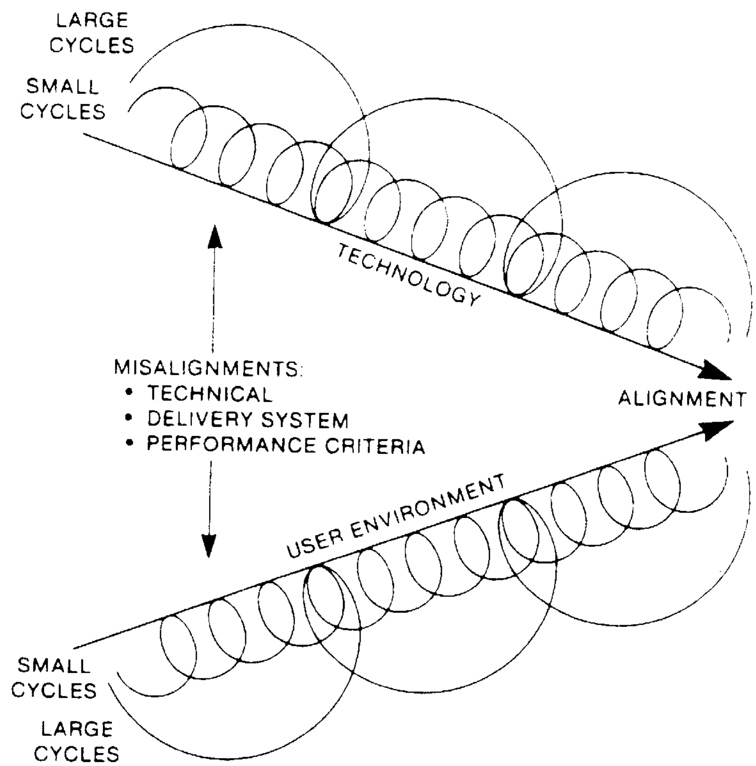

Figure 3. Mutual adaptation of technology and organization. (Reprinted from [35]. © 1988, with permission from Elsevier Science.)

She also introduces the logic of "fit" by showing the potential for misalignments among the technical details of the technology, how success is measured, and how the technology is used in the user environment. Perhaps more than any other description of the process of adoption, this one tips away from normal engineering and towards a more liberal allowance for the evidence that will be admitted (translated) as knowledge. This view will argue against application of traditional project management for adoption because loops are not permitted in normal descriptions of projects and also because it will be difficult to estimate the transit time and resources needed to make forward progress. Rather, progress in this model is made by surfacing and addressing issues and bottlenecks, and it is difficult to anticipate what those might be and how long it might take to resolve each one. Accordingly, managing adoption *qua* learning has apparent appeal.

Inspired by Leonard-Barton, we can find additional evidence of the messiness of implementation. Caputo [32] offer a unique perspective, in which cycles of doubt, concern, and certainty are normal. She found that if two of these cycles could be completed then change, that is implementation, is successful.

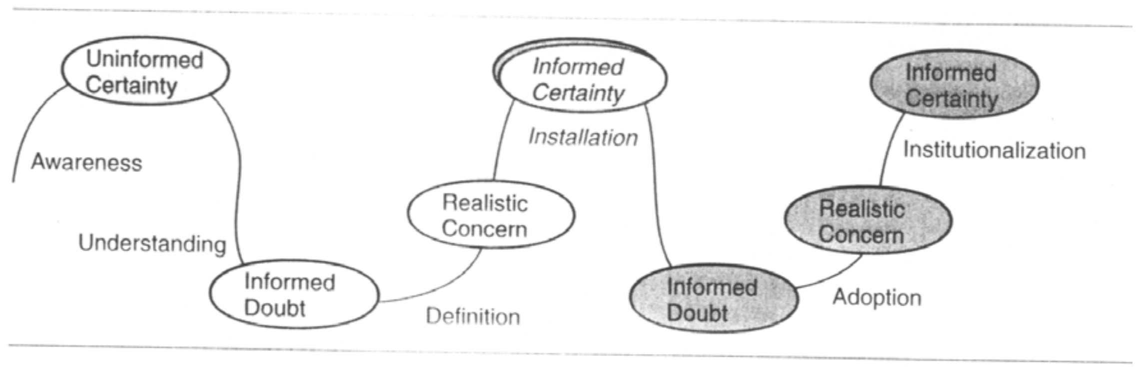

Figure 4. Keep the rhythm going for two beats to make successful change (from [32], fig. 4-12, p. 61. © UNISYS Corp. Reprinted by permission of Pearson Education, Inc.).

Another view that supports the cyclic nature of change, and therefore the problem of predicting how many cycles an organization will transit in order to make change, comes from the addiction literature. [36] At the risk of conflating the unit of analysis by moving from organizational to individual, there is some value in seeing that in certain contexts it is normal that change is not a linear, step-by-step process, but rather an (*a priori*) unknown number of iterations, each of which consumes an unknown length of time.

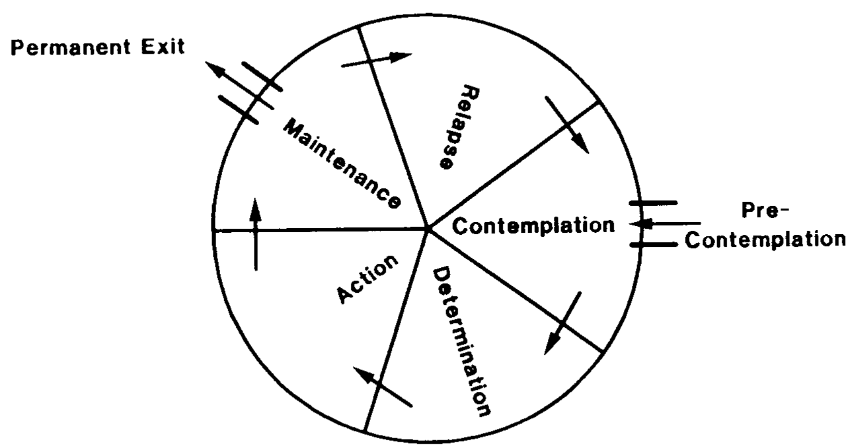

Figure 5. Prochaska and DiClemente's six stages of change. (from [36].)

In yet further support for the cyclic nature of adoption, Tyre and Orlikowski [37-38] found that adoption and utilization of technology is not an incrementally adaptive pattern described by stage models. Instead they found that mutual adaptation à la Leonard-Barton occurs in a discontinuous pattern that frequently displays periods of routine use. [39]

Perhaps the least "process" of the process models is due to Fowler and Rifkin, [40] the so-called double-bubble; also shown without attribution in Rai

[41](p. 99). It differentiates between the push of technology attractiveness and the pull of market or technology needs. [42] The intuition is that technology producers create innovations that are advocated inside their own organizations, perhaps by staff members who have a marketing role. Those advocates communicate to a population containing potential adopters, sometimes using advertising or other public methods of communication. In organizations that are "eligible " to adopt there are other staff members who represent a "surface" of needs to the world. Those staff members have (many) contacts in the technology provider community so they are "connected." [43] When such a technology receptor locates what appears to be a solution to his/her organization's problems, then the diffusion process described by Rogers, below, begins.

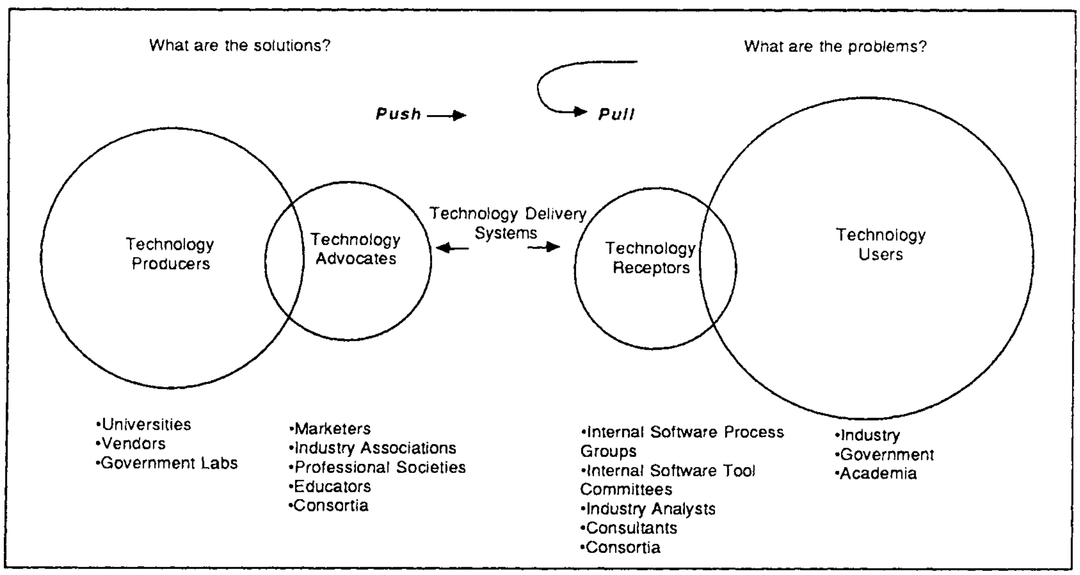

Figure 6. "Double-bubble" process of technology adoption. (from [40], p. 118, permission to reproduce © 1990 by Carnegie Mellon University is granted by the Software Engineering Institute.)

Other process descriptions include Huff and Munro, [44] and Lassila and Brancheau. [39]

5 Diffusion: the most popular explanation

"Diffusionism does not consist of a single idea." [45] (p.67) Rogers has provided an encyclopedic description [46] of the diffusion of innovation from thought to implementation. He defines "diffusion is the process by which an innovation is communicated through certain channels over time among the members of social system." (pp. 5-6) While Rogers takes pains to argue the contrary, most interpreters view diffusion as a linear, one-way process in which a small group of first adopters ("innovators" in Rogers' terminology) inform the next round of adopters, who in turn inform the next round, etc. And the usual growth of adoption is a pattern like the growth of bacteria in a finite medium, the familiar cumulative S-curve.

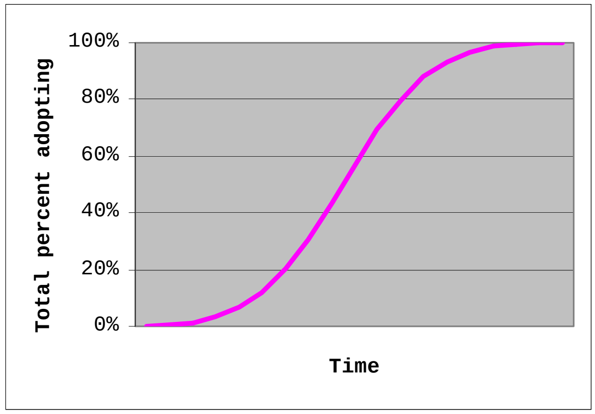

Figure 7. Usual pattern of diffusion, according to [46], p. 106.

The first derivative of the S-curve can be the normal distribution, so many authors, including Rogers, use the symmetric bell curve to describe the population of adopters.

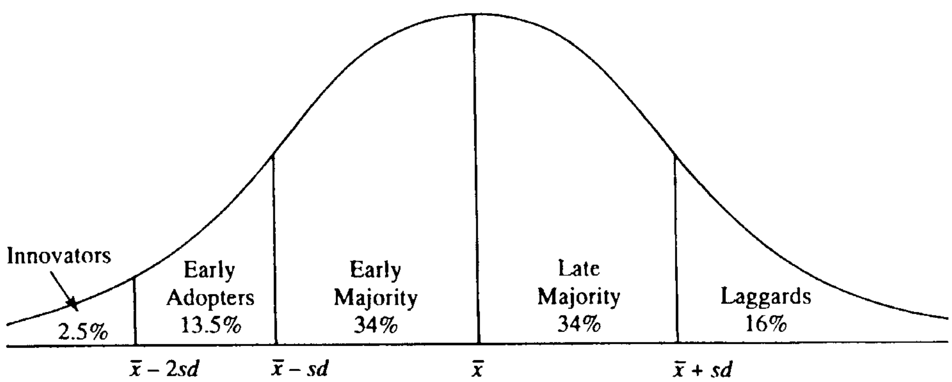

Figure 8. Adopter categorization on the basis of innovativeness (from [46], p. 262. Reprinted with the permission of The Free Press, an imprint of Simon & Schuster Adult Publishing Group. © 1995 by Everett M. Rogers. © 1962, 1971, 1983 by The Free Press.)

Somebody develops an innovation. The innovation has (user) features that can be fairly exactly described and it is clearly separated from other physical objects or abstract phenomena. The innovation is in essence without modifications, spread to people who individually decide whether or not to adopt the innovation. Information about the innovation is initially spread through channels such as professional associations and journals. Next, the news about the innovation is communicated through a social network where the first adopters are key. From these prerequisites follows the division of adopters into the categories … [47] (p. 36)

Since it is based on a mathematical formulation of a communication process, several authors [48-53] use diffusion to predict adoption. In particular they often use a formulation due to Bass [52] (q.v. for a diskette containing an Excel spreadsheet to compute the diffusion), in which additional forces are taken into account, namely the (internal) pressure to imitate and the (external) pressure to innovate.

The mathematical formulation of diffusion can be stated as: $\frac{dN(t)}{dt} = g(t)\left[\overline{N} - N(t)\right]$, where $N(t)$ is the cumulative number of adopters at time t, $\overline{N}$ is the total number of potential adopters in the social system at time t (the ceiling or asymptote of the adoption curve), and $g(t)$ is the coefficient of diffusion. Then the general, mixed influence model is $\frac{dN(t)}{dt} = (a + bN(t))\left[\overline{N} - N(t)\right]$, where everything is as before, and a is the coefficient of external influence and b is the coefficient of internal influence. As mentioned above, external influence refers to the pressure to innovate and internal influence refers to the pressure to imitate. With appropriate manipulation, the mixed influence equation can be solved for $N(t)$ for estimated values of a and b (which are usually estimated from history). [51]

The International Federation for Information Processing (IFIP) Working Group 8.6 was established by IFIP in 1994 to focus on diffusion, transfer & implementation of information technology. Working Group 8.6 conducts conferences and workshops, maintains a listserv, publishes books (usually workshop proceedings [54-57]) and a semi-annual newsletter (Eight.six). (http://www.isi.salford.ac.uk/ifip/home.html) In addition, the Diffusion Interest Group in Information Technology (DIGIT) usually holds an annual workshop in conjunction with the International Conference on Information Systems (http://www.icisnet.org/).

5.1 Problems with diffusion as an explanation

Despite its surface appeal there have been numerous objections. Perhaps the most striking one is "No theory of diffusion has been developed as yet. Hence, diffusion, at best, might is [sic] an umbrella for strategy, innovation, network theory, social structural theory, and a host of other approaches to understanding change in organizational settings." [47] (p. 35) Rogers himself has a section on "Criticisms of diffusion research" [46] (Ch. 3) that addresses many of the arguments against diffusion as an explanation of adoption.

Perhaps two of the most unaddressed areas by Rogers are complexity and colonialization. Numerous authors [26,47,58-64] remark that Rogers' view of diffusion is too simple, too linear. It does not take into account price, substitutable alternatives, marketplace externalities (such as standardization or widespread adoption), network externalities (how many others are using it that I need to interconnect with), [65] the diversity of the adopter population, the complications of making the adoption decision (one part of the organization decides to adopt, another pays for it, and yet another is actually the target of change), simple vs. complex innovations, and radical vs. incremental innovation. Granstrand, for example, proposes a model where the diffusion is separable between buyers and sellers [66], Glaziev and Kaniovski propose a model that is stochastic, not just deterministic (as is Rogers) [67], Fichman and Kemerer study the situation in which the technology has been acquired but not deployed [59], Lyytinen takes into account transaction costs [68], Swanson and Ramiller note that innovation and diffusion are not separate and disconnected stages [69],[5] and Chaddha and Chitgopekar argue that Rogers explains successful diffusion but not its failure, in this case Picturephone [70].

One of the most popular of these views is Moore [61], a marketing specialist, who tried to apply Rogers' adopter categories and instead found gaps, chasms, between adjoining categories. Moore redrew Rogers' bell curve:

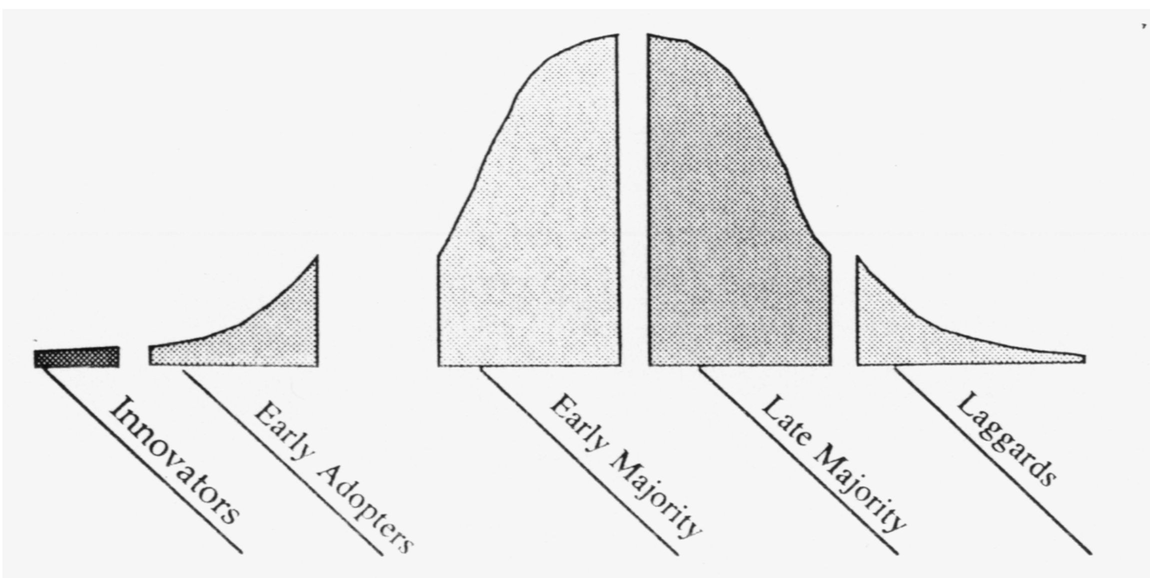

Figure 9. The *revised* technology adoption life cycle. (from [61], p. 17. © 1991 by Goeffrey A. Moore. Reprinted by permission of HarperCollins Publishers Inc.)

[5] This is akin to the social structuring of technology.

The speculation based on experience is that adoption is not a smooth process like a contagion, rather it is a difficult selling effort in which adopters of earlier categories do not and possibly cannot influence future adopters. Rather, it is up to marketing and sales forces to reframe the reasons for adoption and to act as the power behind the diffusion.

Similar to this view is that diffusion is a construct of reductionists who see the world as a set of problems to be solved, where the problems are defined in terms of a (limited) number of variables that are usually related linearly. This ontological commitment relies on a persistent or observer-invariant truth, with time moving linearly at a universal rate with no consideration of different time perceptions. [71]

The principal antidote applied to the diffusion perspective for information systems is Soft Systems Methodology [72,73] and its cousin, Actor Network Theory [17]. These approaches are more qualitative than quantitative, see innovations not so much as waiting to be discovered but rather already in the landscape to be reframed (translated) into useful knowledge by any of the actors, not some specially-designated "inventors" or geniuses. There is a collective, holistic sense, not atomistic (that is, reductionist) sense, to these methods, so they tend to be rich with detail and a complexity that reflects the situation (that is, requisite complexity). In addition, these methods do not separate technology and the social system, rather technology is situated along with many other things in the social system. Another way of saying this is that adoption of technology is socially constructed. [74]

The concept of the social construction of business events is illustrated by a business researcher, [75] "Organizational change is stimulated not by *pressures* from the environment, resulting in a buildup of problems triggering an automatic response, but by the *perceptions* of that environment and those pressures held by key actors." (p. 281)[italics in original] Later she writes, "Organizational change consists in part of a series of emerging constructions of reality, including revision of the past, to correspond to the requisites of new players and new demands." (p. 287)

Perhaps the most intriguing critique is that diffusion à la Rogers is fundamentally imperialism, the standard model of Western colonialism. [45] Basically this view is that we view adopters as open vessels ready to accept (that is, adopt) our ideas because we are superior so then are our ideas. Traditional diffusion assumes evaluation of innovation on the basis of the transmitter, not necessarily of the receiver. "For example at a fundamental level, diffusionism takes 'facts' to be pre-existing (often hidden), waiting to be uncovered at some point by heroic discoverers and inventors." (p. 68)

6 Resistance

Markus [10] is not the first to observe that we often attribute our frustration with the rate of adoption to the individual personalities of the adopters. In particular, we typify them as "bad characters," resisters who eschew change, stuffy, viscous, and ossified. Certainly one of Rogers' contributions is that to the extent personalities enter into the adoption decision, there is a range of possibilities to consider, including those that easily, quickly, and readily adopt.

6.1 Reluctance because the proposed change is a bad idea. That is, there is conflict!

Sometimes conflict itself is the beacon that can serve to warn us that the change(s) we are proposing are bad ideas. It's not the actors, it's the technology being implemented: it does not fit in some significant way. This tension can be managed in a dialectic, [76] or used as a barometer such that implementation is not attempted until and unless the tension has been resolved rationally.

One of the conflicts identified is a misfit with strategy. [77,78] Essentially, most process improvements are aimed at a particular organizational strategy or value proposition called operational excellence. This strategy is to be the lowest cost provider in a market by having the highest quality. Operationally excellent organizations have short menus of goods or services, and have a "formula" for addressing buyer needs. But that leaves two other strategies underserved by traditional process improvement. Those strategies are product innovativeness and customer intimacy.

Each of the three strategies requires a different set of software engineering process innovation in order to optimize the values it delivers to its clients. Therefore, one size cannot fit all, and therefore some of the organizations who attempt to adopt a particular innovation might find that it is ill-suited for its purposes. Naturally, the members of that organization should be counted on to raise this possibility to those supporting adoption, not as resisters, but rather as protectors of the organization's unique value proposition.

Perhaps it is worth mentioning that sometimes hierarchical power is used to counter resistance. While in general beyond the scope of this chapter, there is at least one study [79] of software engineering innovation that indicates that when the power inherent in the hierarchy is imposed on those required to adopt we can expect a lack of adoption. The exercise of the power of the hierarchy is not an effective accelerator of innovation adoption.

In addition, exercise of the power of the hierarchy is coercive [80], something usually perceived and resented by software engineers. The most effective innovative organizations use influence instead of authority to stimulate change [81].

6.2 *Reluctance because we are inertial beings and we resist change.*

Tushman and his colleagues [82,83] examined organizations that had made successful technological changes and compared them with organizations that had not successfully made such changes. They sought the critical difference that spelled success or failure.

They developed a construct, competency-enhancing and competency-destroying, to characterize technology and the way it was introduced for implementation. Competency-enhancing technology is that which performs functions we already perform another way. The standard examples are word processors and spreadsheet programs, as we have been writing and calculating for a long time before the advent of these computerized tools. We implement competency-enhancing technologies by executing the tutorials, reading the help screens, and asking central help desks.

Competency-destroying technologies, on the other hand, require that we learn something so new that everything we have learned to date may not help us understand this new technology. In the realm of software engineering, object orientation is a competency-destroying technology to a lifelong COBOL programmer. Nothing about COBOL or the years spent becoming competent would help prepare a COBOL programmer for object orientation. The authors in their study found that implementing a competency-enhancing technology is different than implementing a competency-destroying one. For competency-destroying, we are undermining the power relations in the organization, we are making people who have become competent and skilled look stupid and incompetent. We are threatening to move them from self-actualization down the Maslow hierarchy back to shelter and hygiene. That is, we are threatening them with losing their jobs.

Even competency-enhancing technologies, such as new versions of word processors, may require that users initially reduce their productivity from current levels while they climb the lower rungs of the learning curve for the "enhanced" technological innovation. If users are not given time and resources to cross the "adoption chasm," the technology will be viewed as competence-destroying by the users and their first-level managers. Users thus get caught in the "competency trap" [83a] of a now obsolete technology that they have mastered.

In order to protect the ability to pay mortgages, people who have to adopt competency-destroying technologies have to be reassured that their competence will be gradually restored in a planning and managed way, that their power and ability to pay their mortgage are not at stake, and that they won't have their performance assessed for a long time as they learn this new technology. The method of implementation is more idiosyncratic, personal, one-to-one, adapted to the particular characteristics of the new learner.

There are other variations than competency-enhancing vs. -destroying, e.g., evolution vs. revolution [84].

Many authors have suggested that "resistance" be dropped from our collective vocabularies [85-92] at least because by labeling a behavior we stop examining it, stop continuing to develop an understanding of it. One thing seems evident: we are not inertial beings, resistant to change. But it is natural that we do express doubts about changes that challenge either what we understand is the purpose of our organization or our long-earned power inside the organization.

6.3 Institutional forces invite us to imitate, to conform

Institutionalization in sociology is the habit of an organization to repeat what it knows and to imitate others that it admires.[6]

Just look at the title of the seminal article on the subject: *The Iron Cage Revisited: Institutional Isomorphism and Collective Rationality in Organizational Fields.* [93] The Iron Cage! Iron cage is the literary term for prison. Max Weber, one of the most famous sociologists, wrote: "… the care for external goods should only lie on the shoulders of the 'saint like a light cloak, which can be thrown aside at any moment.' But fate decreed that the cloak should become an iron cage." DiMaggio and Powell write that by this Weber warned that rationalism had ushered in an era in which capitalism and its off-spring, bureaucracy, had become an iron cage for humanity (p. 147).

What accounts for the lack of diversity in organizational life when organizations themselves – from the standpoint of the diversity of the people in them and the diversity of their markets and market disciplines – seem so different? It's that organizations copy one another and there is great pressure to look and act alike, the authors show. The main point of the literature on institutionalization is what a strong, pervasive, and latent force it is. Organizations may not so much resist change, rather they conform to very large, powerful norms. One can see it in the number of organizations that try to imitate Microsoft's software development practices, but few try to imitate the Software Engineering Laboratory at NASA's Goddard Space Flight Center, a standard example of a measurement-centric high process maturity organization. (http://sel.gsfc.nasa.gov/)

---

[6] Unfortunately the Software Engineering Institute has used the term to mean adoption, the way Rogers defined it as "regular usage."

6.4 Latency because there is gap between knowing and doing.

Authors Pfeffer and Sutton, in *The Knowing-Doing Gap* [94]. explore a phenomenon that many of us see every day: we fail to do what we know we should. Their four-year in-depth study indicates that there are no simple answers, except, perhaps that more information is not needed (for example, ask a cigarette smoker if he/she needs more information in order to stop). They found eight guidelines for closing the gap between knowing what to do and actually doing it.

6.4.1 Why before how: Philosophy is important

Organizations that try to copy the processes of others often find those processes ineffective. The survey illustrates that processes are situated in organizations that have reasons for performing those processes and the borrowing organization might not have the "why," so the "how" does not work.

6.4.2 Knowing comes from doing and teaching others how

To some degree this is restatement of what Peters and Waterman [95] found long ago among some of the most successful organizations: a bias towards action. In a theme repeated often, we cannot know how to implement unless we try it because it's a messy process. And we should not mistake talk for action, we should not accept that deciding to implement is the same as implementing.

6.4.3 Action counts more than elegant plans and concepts

To some degree this is restatement of the principle above and borrows from Peters and Waterman's famous observation of "ready, fire, aim." There is such value in learning by doing that plans and concepts are no substitute.

6.4.4 There is no doing without mistakes. What is the company's response?

In the world of action mistakes are inevitable. Organizations that have closed the knowing-doing gap treat mistakes as a natural part of doing and use the mistakes as occasions for learning. Surgeons call this "forgive and remember"! (p. 132)

6.4.5 Fear fosters knowing-doing gaps, so drive out fear

If we fear for our jobs then we are less likely take the chances that are inherent in performing some new action, making the inevitable mistakes. We would fear that such mistakes would count against us and may form a basis for an assessment of poor performance, and then we could lose our jobs.

6.4.6 Beware of false analogies: fight the competition, not each other

In some organizations, particularly ones that are considered fiercely competitive, it is difficult to achieve teamwork because the external competitive spirit "leaks" into internal behavior and undermines cooperation. Therefore, internal competition defeats closing the knowing-doing gap.

6.4.7 Measure what matters and what can help turn knowledge into action.

Those organizations that have closed the knowing-doing gap use (a) a few simple measures, (b) that have a clear implication of the impact of each person's performance on the goals of the organization. Some organizations use a balanced scorecard, but cut the number of measures way down from the number usually suggested for "completeness."

6.4.8 What leaders do, how they spend their time and how they allocate resources, matters

Time is a non-renewable resource. We all know that, so if our leader spends his or her time acting on knowledge then we all see that knowledge is valued and we begin to understand that it is part of our job, too.

7 Path dependence theory

Path dependence is the observation that sometimes the trajectories of events leading to an innovation all pointed in the same direction. There was a path, more or less inexorable, along which our adoption travels. How could we fail to adopt microwave ovens and cellular phones? There were historical antecedents of what we thought is novel.

Sometimes this temporal process that underlies the construction of phenomena is called creative destruction [96]. If there is an inexorable path, then what about innovators and entrepreneurs? Garud and Karnøe [97] come to the rescue by proposing a relatively new construct, path creation. Stated most succinctly,

> For entrepreneurs attempting to create paths, the world is constantly in the making. Indeed, entrepreneurs creating new paths are more likely to embrace a logic of mindful deviation. Such logic involves spanning boundaries between structures of relevance. On one hand, entrepreneurs are insiders possessing knowledge of a technological field and an appreciation of what to deviate from and the value of pursuing such a strategy. On the other hand, they are outsiders evaluating how much they can deviate from existing relevance structures. And because many deviations are perceived as threatening, entrepreneurs have to buy time, with which and within which to protect and nurture new ideas and create provinces of meaning. From this

> perspective, ideas are carefully evaluated on an ongoing basis and even those that are abandoned may play a role in shaping ideas that survive over time. Temporal elasticity is linked with intertemporal acumen. (p. 9, without notes and references)

This description is akin to Leonard-Barton's in Figure 3, above. In this sense, we implementers are the entrepreneurs about which Garud and Karnøe speak because we are treading new ground as we try to weave technology and organization together.

8 Process studies

Besides the process studies cited in Section 4, above, we add here a few of the details of the process of adoption from Rogers' explanation of diffusion. [46] He defines "the innovation-decision process as the process through which an individual (or other decision-making unit) passes from first knowledge of an innovation to forming an attitude toward the innovation, to a decision to adopt or reject, to implementation and use of the new idea, and to confirmation of this decision." (p. 20) The five main steps in time order, therefore, are (1) knowledge, (2) persuasion, (3) decision, (4) implementation, and (5) confirmation. For Rogers, the transit is a communication process, where individuals use communication channels to traverse the steps.

We can visualize the steps Rogers describes by thinking of a different kind of communication, that of a disease, the type that is, well, communicable. The progression of steps that Rogers describes is an epidemic, started in a particular locus and then transmitted to an ever-increasing radius of individuals by positive contact. [98] In this model, increased communication implies increased adoption. Increased speed of communication implies increased speed of adoption.

9 Factor studies

Factor studies seek to identify and isolate the variables that correlate with outcomes. The usual criticism of factor studies most significant in our case is that factors represent some linear combination of influences, but offer little about the timing, interaction, and causal implications of the influences. Some studies cover a large number of factors [99-101] and others focus on a single factor or cluster of them that might be identified by a single term, such as "leadership."

Perhaps the best study for our purposes is Lopata's dissertation. [102] She examined a range of factors in four categories (see Figure 10, below).

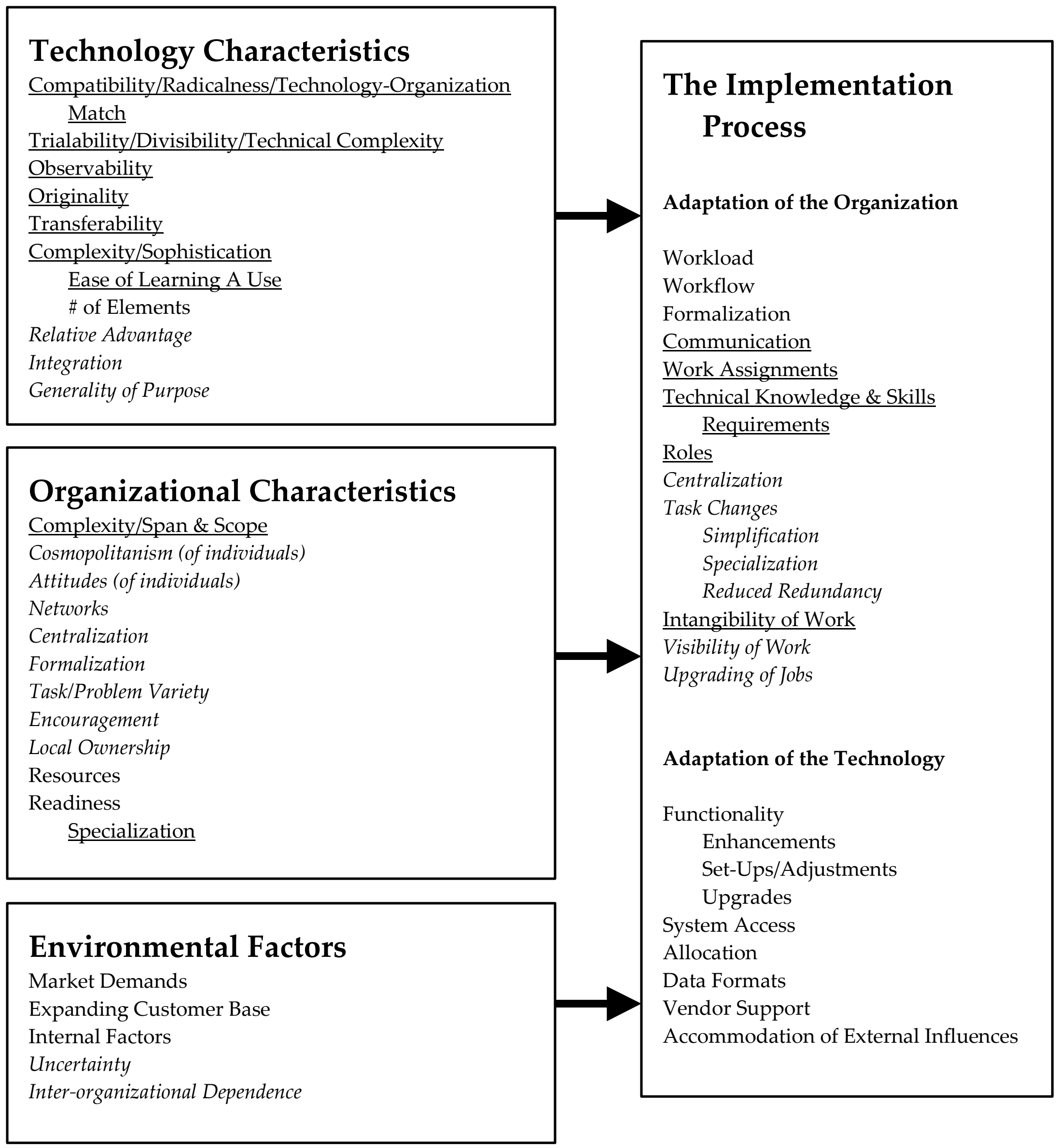

Figure 10. Factor model of the information system implementation process. Adapted from [102], p. 95. Legend: Plain = not predicted by the literature, yet found in the Lopata study, that is, a new factor; *italics* = predicted in the literature and not found in the study; underlined = predicted in the literature and found in the study.

As the reader can see, Lopata attempts to predict what drives Leonard-Barton's model mutual adaptation between the organization and the technology [35]. Lopata's study is one of the very few that have any quantitative data, holding out the hope that one day we may be able to predict the duration, effort, and resources required to implement an information system.

9.1 Characteristics of the innovation

Lopata uses five of Rogers' characteristics of innovations that predict adoption [46] (pp. 15-16):

1. Relative advantage is the degree to which an innovation is perceived to be better than the idea or product or process it supersedes.
2. Compatibility is the degree to which an innovation is perceived to be consistent with the existing culture and needs. See also Ramiller for a counter-argument. [103]
3. Complexity is the degree to which the innovation is difficult, or at least more difficult that its competitors.
4. Trialability and divisibility are measures of the degree to which an innovation may be taken apart and only a part tried. A thick, monolithic innovation has a lower trialability than one that has separable components, each of which adds some value.
5. Observability is the degree to which the results of the implementation will be visible.

Tornatzky and Klein [104], studying 75 reports of innovation characteristics, augment the list with:

6. Cost, presumably negatively related to adoption.
7. Profitability, presumably positively correlated with adoption.
8. Social approval is the degree to which one's status is improved by the innovation. In light of the "cost" and "profit" categories, this one refers to a non-financial reward.

Lopata found these factors to add:

9. Originality is the novelty of the innovation.
10. Transferability refers to the ability inherent in the innovation to transfer skill and knowledge about it to others.
11. Generality of purpose is the degree to which an innovation fills a large space of needs. Low generality would be a very specific innovation.

There is an important sense that in order to improve the chances of implementation, one must design into the technology (that is, product, services, or process) characteristics that make it possible to adopt a little at a time, factors such as trialability, divisibility, observability, and transferability. These all imply the "chopping" up of the technology so that it can be absorbed in small pieces, not as one whole, big chunk. This may be the reason for the popularity of the 12 eXtreme programming practices or the Rational Unified Process: one can select as much or as little as is needed in one application. And this may be part of the problem with grand improvement schemes, such as the SEI CMM or ISO 9000:

they are a lot to swallow and you have to swallow the whole thing to earn the certification.

Swanson creates a typology of information systems innovations and notes that the pattern of adoption is different for different types of innovations. [105]

9.2 Organizational characteristics

The organization is the collective that is going to use the new product or process. There are factors in the target organization that can accelerate or impede implementation. Lopata [102] cites the following factors among many others:

1. Networks refers to the existence of communication channels via which information (buzz) about an innovation can travel.
2. Centralization is the degree to which decisions are made centrally, presumably taking into account factors from a wide perspective.
3. Formalization is the degree to which decisions are formally decided, with a written trail (see also [106]).
4. Task/problem variety is the degree to which the work that the innovation addresses is routine or varied.
5. Encouragement is the degree to which there is a climate of risk taking with respect to innovation.
6. Local ownership is the degree to which implementation is managed locally, independent of the decision to adopt (which would be centralization).
7. Resources is the degree to which the organization has the ability to adopt the innovation.

In addition, organization structure has been studied [107], the impact of information overload [108], task-technology fit, technology-strategy fit [109], product championship and top management support [110], and overall organizational context (a contingency approach) [111-115].

9.3 Environmental factors

There are factors that surround the organization, adopters, and the innovation. They are large-scale forces that can impact the environment in which innovations are being created, introduced, and implemented.

1. Market demands is the degree to which this innovation is required by the market, reflects an imitation of a competitor organization, or is a ahead of the market.
2. Uncertainty is the degree to which market conditions are unforeseen or unforeseeable.
3. Inter-organizational dependence is the degree to which an innovation will have a ripple effect among related organizations.

Slightly different lists can be found in other studies [116-119]. Also, there have been studies that focus on the relative strength of the pull of market needs vs. the push of technological advantages [120-122].

9.4 Adopter characteristics

Rogers [46] lists the five categories that are in Figure 8 and Figure 9. He calls first adopters or innovators venturesome, early adopters respectful, the early majority deliberate, the late majority skeptical, and the laggards traditional. Moore [61] calls the first adopters deviant, early adopters visionaries, early majority needing a business case, late majority want the innovation shrink-wrapped, and laggards are, well, never going to adopt.

Lopata [102] adds (p. 95):

6. Span and scope, which refers to the reach of communications by an individual. This is sometimes called sphere of influence.
7. Cosmopolitanism, which implies that people who are more worldly adopt more easily.
8. Readiness is the degree to which an individual has the resources to attend to an innovation.
9. Specialization is the degree to which special knowledge is needed to implement the innovation or the benefits of the innovation. See also Fichman and Kemerer [58] and Marshall et al. [123].

Other authors offer additions, in particular voluntariness, management support, expectation realism, and the participation in the adoption decision by the targets [124], demographics [125], adoption beliefs [126], implementation history [127], characteristics of the external information sources and communication channel effectiveness [41,128], and job experience and persuasion strategy [129].

9.5 Leadership

Perhaps one of the most-cited characteristics of successful implementation efforts is executive sponsorship, that is, how people at the top of organizations express their leadership. Perhaps the most articulate and detailed advocate for the leadership effect on implementation is Rosabeth Moss Kanter. [75,130-158]

Moss Kanter, through extensive and intensive case studies, has found a number of philosophies and behaviors that if leaders adopt them then there is markedly increased probability of implementation success. Her work straddles leadership, innovation, managing change, and implementation. Her mantra is "The imagination to innovate, the professionalism to perform, and the openness to collaborate, this is how to lead the change-adept organization." [155] She is sanguine about providing The Answer: [158] "I also learned there are no easy answers. Indeed, I conceive of the task of 'managing change' – a task we perform in our personal lives as well as our business lives – as a series of perennial balancing acts. We must juggle contradictions, we must make tradeoffs between contrasting goals, and we must steer a course that does not go too far in any one direction lest events require an about-face. We are perched on a pendulum that is swinging back and forth faster and faster." (p.13)

9.6 User acceptance

A number of studies address the distinction among the factors of usage, perceived ease of use, and perceived usefulness [77,159]. Others have addressed user involvement in the systems development process, which, in our context would be akin to software engineers being involved in the design and development of their own management and engineering practices [160,161].

10 Case studies

Case studies are perspectives or retrospectives that depend upon the observer's direct experience. While all case studies try to offer information that can be used generally, it is usually unclear how to generalize these personal observations.

10.1 Diffusion

As diffusion is a simple, easily-understood, and often-cited framework for adoption, there are many case studies that bear on software engineering processes [162-164]. See Zelkowitz for a particularly careful case study. [27]

10.2 Other case studies

Swanson is a collection of case studies [165]. In addition there is a survey of software developers' perception of the value of software process improvement [166], expert systems adoption that does not appear to use a theoretical basis for data collection [167], evolution of CASE adoption in Finland [168], CASE adoption in Taiwan [169], the difference between user and non-users of CASE [170], the use of innovation characteristics to predict failure [171], phased adoption applied to reuse [172], and adoption patterns and attitudes about computer-supported meetings [173].

11 Discussion

One observation is inescapable: we don't have models of adoption that we can use quantitatively to estimate the duration of adoption, the cost of adoption, or the impact of various accelerators and barriers on the rates and degree of penetration. Two studies give us hope:

- Lopata's in which she counted the number of meetings, hours, durations, documents, and other artifacts of adoption. [102].

- Byrd and Marshall's model of the determinants of how information technology impacts organizational performance, [174] which is what we seek for adoption. See also Cale and Curely [175].

A common theme is that there is not one single answer to why software engineering processes, or any other processes, are not implemented. Instead, implementation can be a messy, social process among humans and not have much in common with the expected practice of engineering. To illustrate this and give us hope that we can marshal the actions of implementation we might look at what Rosabeth Moss Kanter [75] found as she chronicled the implementation of innovations in many organizations (pp. 284 ff):

- Individuals disappear into collectives
- Early events and people disappear into the background as later events and people come forward
- Conflicts disappear into consensuses
- Equally plausible alternatives disappear into obvious choices
- Accidents, uncertainties, and muddle-headed confusions disappear into clear-sighted strategies
- Multiple events disappear into single thematic events
- The fragility of changes (that exist alongside the residues of the old system) disappear into images of solidarity and full actuality.

12 Acknowledgements

This chapter has benefited from improvements suggested by Eric Busboom, Ray Fleming, Suzanne Garcia, Robert Glass, Watts Humphrey, Philip Johnson, John Kunz, RayLevitt, Steve Ornburn, Mark Paulk, Shari Lawrence Pfleeger, and John Tittle. I am especially grateful to the Series Editor, Marvin Zelkowitz, for letting me express some thoughts that had been brewing for a long time.

13 Reference List

1. Risen, J., & Johnston, D. (2002, September 8). Not much has changed in a system that failed: The F.B.I. and C.I.A. missed signals a year ago. Now they do well in capital turf wars. *New York Times*, Section 4, p. 1.
2. Gaw, J. L. (1999). "'A time to heal': The diffusion of Listerism in Victorian Britain." American Philosophical Society, Philadelphia, PA.
3. Senge, P., and et al. (1999). "The dance of change: The challenges of sustaining momentum in learning organizations." Currency Doubleday, New York, NY.
4. Fenton, N., Pfleeger, S. L., and Glass, R. L. (1994, July/August). Science and substance: A challenge to software engineers. *IEEE Software, 11*(4), 86-95.
5. Williams, L., Kessler, R. R., Cunningham, W., and Jeffries, R. (2000, July/August). Strengthening the case for pair programming. *IEEE Software, 17*(4), 19-25.

6. Cohen, M. D., March, J. G., and Olsen, J. (1972, March). A garbage can model of organizational choice. *Administrative Science Quarterly, 17*(1), 1-25.
7. Repenning, N. P. (2002, March-April). A simulation-based approach to understanding the dynamics of innovation implementation. *Organization Science, 13*(2), 109-127.
8. Senge, P. M. (1990). "The fifth discipline: The art & practice of the learning organization." Currency Doubleday, New York, NY.
9. Richardson, G. P. (1991). "Feedback thought in social science and systems theory." University of Pennsylvania Press, Philadelphia, PA.
10. Markus, M. L. (1983, June). Power, politics, and MIS implementation. *Communications of the ACM, 26*(8), 430-444 .
11. Fitzgerald, B. (1996, January). Formalized systems development methodologies: A critical perspective. *Information Systems Journal, 6*(1), 3-23.
12. Roth, G., and Kleiner, A. (2000). "Car launch: The human side of managing change." Oxford University Press, New York, NY.
13. Bostrom, R. P., and Heinen, J. S. (1977, September). MIS problems and failures: A socio-technical perspective, part I: the causes. *MIS Quarterly, 1*(3), 17-32.
14. Harris, M. (1996, March). Organizational politics, strategic change and the evaluation of CAD. *Journal of Information Technology, 11*(1), 51-8.
15. Ryan, T. F., and Bock, D. B. (1992, November). A socio-technical systems viewpoint to CASE tool adoption. *Journal of Systems Management, 43*(11), 25-9.
16. Burrell, G., and Morgan, G. (1979). "Sociological paradigms and organizational analysis." Heinemann, Portsmouth, NH.
17. McMaster, T. , Vidgen, R. T., & Wastell, D. G. (1997). Technology transfer: diffusion or translation? *In* "Facilitating technology transfer through partnership: Learning from practice and research (Proceedings of the IFIP TC8 WG8.6 international working conference on diffusion, adoption and implementation of information technology)" (McMaster, Tom, Mumford, Enid, Swanson, E. Burton, Warboys, Brian, and Wastell, David, Eds.), Ambleside, Cumbria, UK, pp. 64-75. Chapman & Hall, London.
18. Meyerson, D. , and Martin, J. (1987, November). Cultural change: An integration of three different views. *Journal of Management Studies, 24*(6), 623-647.
19. Silva, J., & Backhouse, J. (1997). Becoming part of the furniture: The institutionalization of information systems. *In* "Information systems and qualitative research (Proceedings of the IFIP TC8 WG 8.2 International Conference on Information Systems and Qualitative Research, May 31- June 3, 1997, in Philadelphia, Pennsylvania USA)" (A. S. Lee, J. Liebenau, and J. I.

DeGross, Eds.), Chap. 20, pp. 389-414. Chapman & Hall, London.
20. Orlikowski, W. J., and Baroudi, J. J. (1991, March). Studying information technology in organizations: Research approaches and assumptions. *Information Systems Research, 2*(1), 1-28.
21. Kaplan, A. (1964). "The conduct of inquiry: Methodology for behavioral science." Chandler Pub. Co., San Francisco.
22. Weick, K. (1979). "The social psychology of organizing." (2nd ed.) Wiley, New York.
23. Markus, M. L., and Robey, D. (1988, May). Information technology and organizational change: Causal structure in theory and research. *Management Science, 34*(5), 583-598.
24. Butler, B., & Gibbons, D. (1998). Power distribution as a catalyst and consequence of decentralized diffusion. *In* "Information systems innovation and diffusion: issues and directions" (T. J. Larsen, and E. McGuire, Eds.), pp. 3-28. Idea, Hershey, PA.
25. Lucas, H. C., Jr., Ginsberg, M. J., and Schultz, R. L. (1990). "Information systems implementation: Testing a structural model." Ablex, Norwood, NJ.
26. Kwon, T. H., & Zmud, R. W. (1987). Unifying the fragmented models of information systems implementation. *In* "Critical issues in information systems research" (R. J. Boland, Jr., and R. A. Hirschheim, Eds.), pp. 227-251. John Wiley, Chichester, England.
27. Zelkowitz, M. V. (1996, August). Software engineering technology infusion within NASA. *IEEE Transactions on Engineering Management, 43*(3), 250-61.
28. Redwine, S. T., Jr., Becker, L. G., Marmor-Squires, A. B., Martin, R. J., Nash, S. H., & Riddle, W. E. (1984, June). "DoD related software technology requirements, practices, and prospects for the future." (P-1788). Institute for Defense Analysis, Alexandria, VA.
29. Maier, F. H. (1998, Winter). New product diffusion models in innovation management -- a system dynamics perspective. *System Dynamics Review, 14*(4), 285-308.
30. Schumpeter, J. A. (1939). "Business cycles: A theoretical, historical, and statistical analysis of the capitalist process." McGraw-Hill, New York, NY.
31. McFeeley, B. (1996, February). "IDEAL: A user's guide for software process improvement." (CMU/SEI-96-HB-001). Software Engineering Institute, Pittsburgh, PA.
32. Caputo, K. (1998). "CMM implementation guide: Choreographing software process improvement." Addison-Wesley, Reading, MA.
33. Conner, D. (1993). "Managing at the speed of change: How resilient managers succeed and prosper where others fail." Villard Books, New York, NY.

34. Grady, R. B. , and Van Slack, T. (1994, July). Key lessons in achieving widespread inspection use. *IEEE Software, 11*, 46-57.
35. Leonard-Barton, D. (1988, October). Implementation as mutual adaptation of technology and organization. *Research Policy, 17*(5), 251-267.
36. Miller, W. R., and Rollnick, S. (1991). "Motivational interviewing: Preparing people to change addictive behavior." Guilford Press, New York, NY.
37. Tyre, M. J., and Orlikowski, W. J. (1994, July). The myth of continuous improvement. *Chemtech, 24*, 12-19.
38. Orlikowski, W. J. (1993, September). CASE tools as organizational change: Investigating incremental and radical changes in systems development. *MIS Quarterly, 17*(3), 309-40.
39. Lassila, K. S., and Brancheau, J. C. (1999, Fall). Adoption and utilization of commercial software packages: Exploring utilization equilibria, transitions, triggers, and tracks. *Journal of Management Information Systems, 16*(2), 63-90.
40. Fowler, P., & Rifkin, S. (1990, September). "Software engineering process group guide." (CMU/SEI-90-TR-24). Software Engineering Institute, Pittsburgh, PA.
41. Rai, A. (1995, May). External information source and channel effectiveness and the diffusion of CASE innovations: An empirical study. *European Journal of Information Systems, 4*(2), 93-102.
42. Zmud, R. W. (1984, June). An examination of "push-pull" theory applied to process innovation in knowledge work. *Management Science, 30*(6), 727-738.
43. Spilka, R. (1995, August). Communicating across organizational boundaries: A challenge for workplace professionals. *Technical Communication, 42*(3), 436.
44. Huff, S. L., and Munro, M. C. (1985). Information technology assessment and adoption: A field study. *MIS Quarterly,* 327-339.
45. McMaster, T. (2001). The illusion of diffusion in information systems research. *In* "Diffusing software product and process innovations (Proceedings of the IFIP TC8 WG8.6 fourth working conference on diffusing software product and process innovations)" (Ardis, Mark A. and Marcolin, Barbara L., Eds), Banff, Canada, pp. 67-86. Kluwer Academic Press, Boston.
46. Rogers, E. M. (1995). "Diffusion of innovations." (4th ed.) The Free Press, New York, NY.
47. Larsen, T. (2001). The phenomenon of diffusion: Red herrings and future promises. *In* "Diffusing software product and process innovations (Proceedings of the IFIP TC8 WG8.6 fourth working conference on diffusing software product and process innovations)" (Ardis, Mark A. and Marcolin, Barbara L., Eds), Banff, Canada, pp. 35-50. Kluwer Academic Press, Boston.
48. Geroski, P. A. (2000). Models of technology diffusion. *Research Policy, 29*,

603-625.
49. Mahajan, V., Muller, E., and Bass, F. M. (1990, January). New product diffusion models in marketing: A review and directions for research. *Journal of Marketing, 54*, 1-26.
50. Jaakkola, H. (1996). Comparison and analysis of diffusion models. *In* "Diffusion and adoption of information technology: Proceedings of the first IFIP WG 8.6 working conference on the diffusion and adoption of information technology" (Kautz, Karlheinz and Pries-Heje, Jan, Eds.), Oslo, Norway, pp. 65-82. Chapman & Hall, London.
51. Mahajan, V., and Peterson, R. A. (1985). "Models for innovation diffusion." Sage, Beverly Hills, CA.
52. Mahajan, V., Muller, E., & Wind, Y. (Eds.). (2000). "New-product diffusion models." Kluwer Academic, Boston.
53. Larsen, T. J., & McGuire, E. (Eds.). (1998). "Information systems innovation and diffusion: issues and directions." Idea, Hershey, PA.
54. Levine, L. (1994). "Diffusion, transfer and implementation of information technology (Proceedings of the IFIP TC8 Working Conference)" , Pittsburgh, PA, North-Holland, Amsterdam.
55. Kautz, K., & Pries-Heje, J. (1996). "Diffusion and adoption of information technology (Proceedings of the first IFIP WG 8.6 working conference on the diffusion and adoption of information technology)" , Oslo, Norway, Chapman & Hall, London.
56. McMaster, T. , Mumford, E., Swanson, E. B., Warboys, B., & Wastell, D. (1997). "Facilitating technology transfer through partnership: Learning from practice and research (Proceedings of the IFIP TC8 WG8.6 international working conference on diffusion, adoption and implementation of information technology)" , Ambleside, Cumbria, UK, Chapman & Hall, London.
57. Ardis, M. A. , & Marcolin, B. L. (2001). "Diffusing software product and process innovations (Proceedings of the IFIP TC8 WG8.6 fourth working conference on diffusing software product and process innovations)" , Banff, Canada, Kluwer Academic Press, Boston.
58. Fichman, R. G., and Kemerer, C. F. (1997, October). The assimilation of software process innovations: An organizational learning perspective. *Management Science, 43*(10), 1345-1363.
59. Fichman, R. G., and Kemerer, C. F. (1999, September). The illusory diffusion of innovation: An examination of assimilation gaps. *Information Systems Research, 10*(3), 255-275.
60. Bayer, J., and Melone, N. (1989, February). A critique of diffusion theory as a managerial framework for understanding adoption of software

engineering innovations. *Journal of Systems & Software, 9*(2), 161-166.
61. Moore, G. A. (1991). "Crossing the chasm: marketing and selling technology products to mainstream customers." HarperBusiness, New York, NY.
62. Tornatzky, L. G., and Fleischer, M. (1990). "The processes of technological innovation." Lexington Books, Lexington, MA.
63. Lyytinen, K. , & Damsgaard, J. (2001). What's wrong with the diffusion of innovation theory? A case of complex and networked technology. *In* "Diffusing software product and process innovations (Proceedings of the IFIP TC8 WG8.6 fourth working conference on diffusing software product and process innovations)" (Ardis, Mark A. and Marcolin, Barbara L., Eds), Banff, Canada, pp. 173-189. Kluwer Academic Press, Boston.
64. Attewell, P. (1992, February). Technology diffusion and organizational learning: The case of business computing. *Organization Science, 3*(1), 1-19.
65. Lange, R., McDade, S., and Oliva, T. A. (2001, March). Technological choice and network externalities: A catastrophe model analysis of firm software adoption for competing operating systems. *Structural Change & Economic Dynamics, 12* (1), 29-57.
66. Granstrand, O. (1991). Temporal diffusion and population dynamics: A systems model. *In* "Diffusion of technologies and social behavior" (N. Nakićenović, and A. Grübler, Eds.), pp. 247-263. Springer-Verlag, Berlin.
67. Glaziev, S. Yu., & Kaniosvki, Y. M. (1991). Diffusion of innovations under conditions of uncertainty: A stochastic approach. *In* "Diffusion of technologies and social behavior" (N. Nakićenović, and A. Grübler, Eds.), pp. 231-246. Springer-Verlag, Berlin.
68. Lyytinen, K. (1991). Penetration of information technology in organizations: A comparative study using stage models and transaction costs. *Scandinavian Journal of Information Systems, 3*.
69. Swanson, E. B., and Ramiller, N. C. (1997, September-October). The organizing vision in infromation systems innovation. *Organization Science, 8*(5), 458-474.
70. Chaddha, R. L., and Chitgopekar, S. S. (1971, Autumn). A "generalization" of the logistic curves and long-range forecasts (1966-1991) of residence telephones. *Bell Journal of Economics, 2* (2), pp. 542-560.
71. Linstone, H. A. (1991). Multiple perspectives on technological diffusion: Insights and lessons. *In* "Diffusion of technologies and social behavior" (N. Nakićenović, and A. Grübler, Eds.), pp. 53-92. Springer-Verlag, Berlin.
72. Checkland, P., & Scholes, J. (1990/1999). "Soft systems methodology in action." (30-year retrospective) John Wiley & Sons, West Sussex, England.
73. Checkland, P. (1993/1999). "Systems thinking, systems practice." (30-year retrospective) John Wiley & Sons, West Sussex, England.

74. Bijker, W. E., Hughes, T. P., & Pinch, T. J. (Eds.). (1987). "The social construction of technological systems: New directions in the sociology and history of technology." MIT Press, Cambridge, MA.
75. Kanter, R. M. (1983). "The change masters: Innovation & entrepreneurship in the American corporation." Simon & Schuster, New York, NY.
76. Easterbrook, S. (Ed.). (1992). "CSCW: Cooperation or conflict?" Springer-Verlag, London.
77. Rifkin, S. (2001, May/June). What makes measuring software so hard? *IEEE Software, 10*(3), 41-45.
78. Rifkin, S. (2001, July/August). Why software process innovations are not adopted. *IEEE Software, 10*(4), 110-112.
79. Bayer, J., & Melone, N. (1988, August). "Adoption of software engineering innovations in organizations." (CMU/SEI-88-TR-27). Software Engineering Institute, Pittsburgh, PA.
80. Jermier, J. M., Knight, D., & Nord, W. R. (1994). Resistance and power in organizations: Agency, subjectivity and the labour process. *In* "Resistance and power in organizations" (J. M. Jermier, D. Knight, and W. R. Nord, Eds.). Routledge, London.
81. Lawrence, P. , and Lorsch, J. (1967). "Organization and environment: Managing differentiation and integration." Irwin, Homewood, IL.
82. Tushman, M. L., and Anderson, P. (1986). Technological discontinuities and organizational environments. *Administrative Science Quarterly, 31*, 439-465.
83. Tushman, M. L., and Rosenkopf, L. (1992). Organizational determinants of technological change: Toward a sociology of technological evolution. *Organizational Behavior, 14*, 311-347.
83a Levitt, B. and J.G. March.(1988). Organizational Learning, *Annual Review of Sociology*, (14),319-340.
84. Hardgrave, B. C. (1997, April). Adopting object-oriented technology: Evolution or revolution? *Journal of Systems & Software, 37*(1), 19-25.
85. Bartlem, C. S., and Locke, E. A. (1981, July). The Coch and French study: a critique and reinterpretation. *Human Relations, 34*(7), 555-566.
86. Coch, L., and French, J. R. P. (1948). Overcoming resistance to change. *Human Relations, 1*(4), 512-532.
87. Dent, E. B., and Goldberg, S. G. (1999, March). Challenging "resistance to change". *Journal of Applied Behavioral Science, 35*(1), 25-41.
88. Diamond, M. A. (1986, September). Resistance to change: A psychoanalytic critique of Argyris and Schon's contributions to organization theory and intervention. *Journal of Management Studies, 23*(5), 543-562.
89. Hirschheim, R., and Newman, M. (1988). Information systems and user resistance: Theory and practice. *Computer Journal, 31*(5), 398-408.

90. Krantz, J. (1999, March). Comment on "Challenging 'resistance to change'". *Journal of Applied Behavioral Science, 35*(1), 42-44.
91. Merron, K. (1993, Winter). Let's bury the term "resistance". *Organization Development Journal, 11*(4), 77-86.
92. Levine, L. (1997). An ecology of resistance. *In* "Facilitating technology transfer through partnership: Learning from practice and research (Proceedings of the IFIP TC8 WG8.6 International Working Conference on Diffusion, Adoption and Implementation of Information Technology)" (McMaster, Tom, Mumford, E., Swanson, E. B., Warboys, B., and Wastell, David, Eds.), Ambleside, Cumbria, UK, pp. 163-174. Chapman & Hall, London.
93. DiMaggio, P. J., and Powell, W. W. (April 1983). The Iron Cage revisited: Institutional isomorphism and collective rationality in organizational fields. *American Sociological Review, 48*(2), 147-160.
94. Pfeffer, J., and Sutton, R. I. (2000). "The knowing-doing gap: How smart companies turn knowledge into action." Harvard Business School Press, Boston, MA.
95. Peters, T., and Waterman, R. H., Jr. (1982). "In search of excellence: Lessons from America's best-run companies." Harper and Row, New York, NY.
96. Schumpeter, J. A. (1942). "Capitalism, socialism, and democracy." Harper & Row, New York, NY.
97. Garud, R., & Karnøe, P. (2001). Path creation as a process of mindful deviation. *In* "Path dependence and creation" (R. Garud, and P. Karnøe, Eds.), pp. 1-38. Lawrence Erlbaum Associates, Mahwah, NJ.
98. Ahire, S. L., and Ravicahndran, T. (2001, November). An innovation diffusion model of TQM implementation. *IEEE Transactions on Engineering Management, 48*(4), 445-464.
99. Roberts, T. L., Jr., Gibson, M. L., Fields, K. T., and Rainer, R. K., Jr. (1998, August). Factors that impact implementing a system development methodology. *IEEE Transactions on Software Engineering, 24*(8), 640-649.
100. Yadav, S. B., Shaw, N. G., Webb, L., and Sutcu, C. (2001, March). Comments on 'Factors that impact implementing a system development methodology'. *IEEE Transactions on Software Engineering, 27*(3), 279-281.
101. Roberts, T. L., Gibson, M. L., Rainer, R. K., and Fields, K. T. (2001, March). Response to 'Comments on factors that impact the implementation of a systems development methodology'. *IEEE Transactions on Software Engineering, 27*(3), 282-286.
102. Lopata, C. L. (June 1993). *The cooperative implementation of information technology: a process of mutual adaptation*. Unpublished doctoral dissertation, Drexel University, Philadelphia, PA.

103. Ramiller, N. C. (1994, March). Perceived compatibility of information technology innovations among secondary adopters: Toward a reassessment. *Journal of Engineering & Technology Management, 11*(1), 1-23.
104. Tornatzky, L. G., and Klein, K. J. (1982, February). Innovation characteristics and innovation-implementation: A meta-analysis of findings. *IEEE Transactions on Engineering Management, EM-29*(1), 28-45.
105. Swanson, E. B. (1994, September). Information systems innovation among organizations. *Management Science, 40*(9), 1069-1092.
106. Zmud, R. W. (1982, December). Diffusion of modern software practices: Influence of centralization and formalization. *Management Science, 28*(12), 1421-1431.
107. DeCanio, S. J., Dibble, C., and Amir-Atefi, K. (2000, October). The importance of organizational structure for the adoption of innovations. *Management Science, 46*(10), 1285-1299.
108. Edmunds, A. , and Morris, A. (2000, February). The problem of information overload in business organisations: A review of the literature. *International Journal of Information Management, 20*(1), 17-28.
109. Fan, M., Stallaert, J., and Whinston, A. B. (2000, March). The adoption and design methodologies of component-based enterprise systems. *European Journal of Information Systems, 9*(1), 25-35.
110. Premkumar, G., and Potter, M. (1995, May/August). Adoption of computer aided software engineering (CASE) technology: An innovation adoption perspective. *Data Base for Advances in Information Systems, 26*(2-3), 105-23.
111. Rai, A., and Howard, G. S. (1994, March). Propagating CASE usage for software development: An empirical investigation of key organizational correlates. *Omega (Oxford), 22*(2), 133-147.
112. Rai, A., and Howard, G. S. (1993, Summer). An organizational context for CASE innovation. *Information Resources Management Journal, 6*(3), 21-34.
113. Ravichandran, T. (2000, Summer). Swiftness and intensity of administrative innovation adoption: An empirical study of TQM in information systems. *Decision Sciences, 31*(3), 691-724.
114. Ravichandran, T., and Rai, A. (2000, September). Quality management in systems development: An organizational system perspective. *MIS Quarterly, 24*(3), 381-6+405-10.
115. Wolfe, R. A. (1994, May). Organizational innovation: Review, critique and suggested research directions. *Journal of Management Studies, 31*(3), 405-431.
116. Chau, P. Y. K., and Tam, K. Y. (1997, March). Factors affecting the adoption of open systems: An exploratory study. *MIS Quarterly, 21*(1), 1-20.
117. Chiasson, M. W., and Lovato, C. Y. (2001, Summer). Factors influencing the formation of a user's perceptions and use of a DSS software innovation.

Data Base for Advances in Information Systems, 32(3), 16-35.

118. Cooper, R. B., and Zmud, R. W. (1990, February). Information technology implementation research: A technological diffusion approach. *Management Science, 36*(2), 123-139.
119. Fichman, R. G., and Kemerer, C. F. (1993, Winter). Adoption of software engineering process innovations: The case of object orientation. *Sloan Management Review, 34*(2), 7-22.
120. Fowler, P., & Levine, L. (1993, December). "Technology transition push: A case study of rate monotonic analysis (Part 1)." (CMU/SEI-93-TR-29). Software Engineering Institute, Pittsburgh, PA.
121. Fowler, P., & Levine, L. (1995, April [Note that the date from the edition ID is 1993]). "Technology transition pull: A case study of rate monotonic analysis (Part 2)." (CMU/SEI-93-TR-030). Software Engineering Institute, Pittsburgh, PA.
122. Rai, A., and Patnayakuni, R. (1996, Fall). A structural model for CASE adoption behavior. *Journal of Management Information Systems, 13*(2), 205-34.
123. Marshall, T. E., Byrd, T. A., Gardiner, L. R., and Rainer, R. K. Jr. (2000, July-September). Technology acceptance and performance: An investigation into requisite knowledge. *Information Resources Management Journal, 13*(3), 33-45.
124. Iivari, J. (1996, October). Why are CASE tools not used? *Communications of the ACM, 39*(10), 94-103.
125. Iivari, J. (1995, August). Factors affecting perceptions of CASE effectiveness. *European Journal of Information Systems, 4*(3), 143-158.
126. Karahanna, E., Straub, D. W., and Chervany, N. L. (1999, June). Information technology adoption across time: A cross-sectional comparison of pre-adoption and post-adoption beliefs. *MIS Quarterly, 23*(2), 183-213.
127. Myers, W. (1998, April). Why software developers refuse to improve. *IEEE Computer,* 110-112.
128. Zmud, R. W. (1983, June). The effectiveness of external information channels in facilitating innovation within software development groups. *MIS Quarterly, 16*, 43-58.
129. Sagie, A., Elizur, D., and Greenbaum, C. W. (1985, April). Job experience, persuasion strategy and resistance to change: An experimental study. *Journal of Occupational Behaviour, 6*(2), 157-162.
130. Kanter, R. M., and Corn, R. I. (1994). Do cultural differences make a business difference? Contextual factors affecting cross-cultural relationship success. *Journal of Management Development, 13*(2), 5-23.
131. Kanter, R. M. (1989, Summer). Swimming in newstreams: Mastering innovation dilemmas. *California Management Review, 31*(4), 45-69.
132. Kanter, R. M., North, J., Richardson, L., Ingols, C., and Zolner, J. (1991,

March). Engines of progress: designing and running entrepreneurial vehicles in established companies: Raytheon's new product center, 1969-1989. *Journal of Business Venturing, 6*(2), 145-163.
133. Kanter, R. M., and Fonvielle, W. H. (1987, January). When to persist and when to give up. *Management Review, 76*(1), 14-15.
134. Kanter, R. M. (1988, March). Change masters vs. change stiflers. *Executive Excellence, 5*(3), 12-13.
135. Kanter, R. M., North, J., Bernstein, A. P., and Williamson, A. (1990, November). Engines of progress: Designing and running entrepreneurial vehicles in established companies. *Journal of Business Venturing, 5*(6), 415-27.
136. Kanter, R. M. (1987, May). Managing traumatic change: Avoiding the 'unlucky 13'. *Management Review, 76*(5), 23-24.
137. Kanter, R. M. (1991, January/February). Championing change: An interview with Bell Atlantic's CEO Raymond Smith. *Harvard Business Review, 69*(1), 118-130.
138. Kanter, R. M. (1990, November/December). Thinking across boundaries. *Harvard Business Review, 68*(6), 9.
139. Kanter, R. M. (1988, January). Change masters: Playing a new game. *Executive Excellence, 5*(1), 8-9.
140. Kanter, R. M. (1991, May/June). Transcending business boundaries: 12,000 world managers view change. *Harvard Business Review, 69*(3), 151-164.
141. Kanter, R. M., and Richardson, L. (1991, May). Engines of progress: Designing and running entrepreneurial vehicles in established companies - the Enter-Prize Program at Ohio Bell, 1985-1990. *Journal of Business Venturing, 6*(3), 209-229.
142. Kanter, R. M., Richardson, L., North, J., and Morgan, E. (1991, January). Engines of progress: Designing and running entrepreneurial vehicles in established companies; the new venture process at Eastman Kodak, 1983-1989. *Journal of Business Venturing, 6*(1), 63-82.
143. Kanter, R. M., and Parkes, C. (1991, June). In search of a single culture. *Business (London),* 58-66.
144. Kanter, R. M. (1991, July/August). Change: Where to begin. *Harvard Business Review, 69*(4), 8.
145. Kanter, R. M. (1992, January/February). Discipline! *Harvard Business Review, 70*(1), 7.
146. Kanter, R. M. (1992, Mar/April). Six certainties for CEOs. *Harvard Business Review, 70*(2), 7.
147. Kanter, R. M. (1992, May/June). Ourselves versus ourselves. *Harvard Business Review, 70*(3), 8.
148. Stein, B. A., and Kanter, R. M. (1993, Winter). Leadership for change: The

rest of the story. *Frontiers of Health Services Management, 10*(2 Page: 28).
149. Kanter, R. M., and Stein, B. A. (1992, November). New models, but where's the process? -- Transforming Organizations edited by Thomas A. Kochan and Michael Useem. *Contemporary Sociology, 21*(6), 758.
150. Kanter, R. M. (1993, April). Mastering change. *Executive Excellence, 10*(4), 11.
151. Stein, B. A., and Kanter, R. M. (1993, November). Why good people do bad things: A retrospective on the Hubble fiasco. *The Academy of Management Executive, 7*(4), 58.
152. Kanter, R. M. (1996, December 2). Can giants dance in cyberspace? *Forbes (ASAP),* 247-248.
153. Kanter, R. M. (1999, May/June). From spare change to real change. *Harvard Business Review, 77*(3), 122-132.
154. Kanter, R. M. (2000, May/June). The enduring skills of change leaders. *Ivey Business Journal, 64*(5), 31-36.
155. Kanter, R. M. (1997). "Frontiers of management." Harvard Business School, Boston, MA.
156. Kanter, R. M., Stein, B. A., & Jick, T. D. (compiled by). (1992). "The challenge of organizational change: How companies experience it and leaders guide it." Free Press, New York, NY.
157. Kanter, R. M. (1989, December). "Improving the acceptance and use of new technology: Organizational and inter-organizational challenges." (Working Paper 90-043). Division of Research, Harvard Business School.
158. Kanter, R. M. (1989). "When giants learn to dance." Simon & Schuster, New York, NY.
159. Venkatesh, V., Smith, R. H., and Morris, M. G. (2000, March). Why don't men ever stop to ask for directions? Gender, social influence, and their role in technology acceptance and usage behavior. *MIS Quarterly, 24*(2), 115-18+131-6.
160. Ives, B., and Olson, M. H. (1984, May). User involvement and MIS success: A review of research. *Management Science, 30*(5), 586-603.
161. Tait, P., and Vessey, I. (1988, March). The effect of use involvement on system success: A contingency approach. *MIS Quarterly,* 91-108.
162. Buxton, J. N., and Malcolm, R. (1991, January). Software technology transfer. *Software Engineering Journal, 6*(1), 17-23.
163. Jurison, J. (2000, October-December). Perceived value and technology adoption across four end user groups. *Journal of End User Computing, 12*(4), 21-28.
164. Williams, L. R., and Rao, K. (1997). Information technology adoption: Using classical adoption models to predict AEI software implementations. *Journal of Business Logistics, 18*(2), 43-54.

165. Swanson, E. B. (1988). "Information system implementation: Bridging the gap between design and utilization." Irwin, Homewood, IL.
166. Kuilboer, J. P., and Ashrafi, N. (1999). Software process improvement deployment: An empirical perspective. *Journal of Information Technology Management, 10*(3-4), 35-47.
167. Kunnathur, A. S., Ahmed, M. U., and Charles, R. J. S. (1996, January). Expert systems adoption: An analytical study of managerial issues and concerns. *Information Management, 30*(1), 15-25.
168. Maansaari, J., and Iivari, J. (1999, July). The evolution of CASE usage in Finland between 1993 and 1996. *Information & Management, 36*(1), 37-53.
169. Yang, H.-L. (1999). Adoption and implementation of CASE tools in Taiwan. *Information & Management, 35*(2), 89-112.
170. Nelson, A. C., and Rottman, J. (1996). Before and after CASE adoption. *Information & Management, 31*, 193-202.
171. Shim, S. J. (1998). Characteristics and adoption of generic financial expert systems: A case study of failure. *Journal of Information Technology Management, 9*(3), 43-51.
172. Wartik, S., and Davis, T. (1999, April). A phased reuse adoption model. *Journal of Systems & Software, 46*(1), 13-23.
173. Zigurs, I., DeSanctis, G., and Billingsley, J. (1991, Spring). Adoption patterns and attitudinal development in computer-supported meetings: An exploratory study with SAMM. *Journal of Management Information Systems, 7*(4), 51-70.
174. Byrd, T. A. , and Marshall, T. E. (1997). Relating information technology investment to organizational performance: A causal model analysis. *Omega, International Journal of Management Science, 25*(1), 43-56.
175. Cale, E. G. , and Curely, K. F. (1987). Measuring implementation outcome: Beyond success and failure. *Information & Management, 13*(5), 245-253.