Please do not duplicate or distribute.
To be presented at
Adoption-Centric Software Engineering (ACSE 2003),
one of the workshops at the International Conference on Software Engineering,
May 9, 2003, in Portland, Oregon USA

Two good reasons why new software processes are not adopted

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

Abstract

There are many reasons we do not adopt software engineering processes, including those associated with tools. This paper presents two of the most persuasive reasons, based on a literature review of 175 references.

The archetypal dimensions of adopting a new things are: attributes of the thing itself (classically relative advantage, compatibility, complexity, trialability, and observability), qualities of the adopters, the strength of opinion leaders in diffusion networks, characteristics of the change agent, and organizational factors. [16] In addition, other authors have identified environmental factors, too. Lopata cites seven of them. [7] All of these factor studies suffer several deficiencies: they are static and linear combinations, if combinations at all; there is no priority of factors; there is no time variation of the influence of the factors.

In conducting a literature search for a related paper [15], the author read over 175 references seeking to understand what drives software engineering process adoption. The author believes that many of the factors presented in the literature are actually dissatisfiers, that is, their absence will signal adoption impediments, but their presence is not a sufficient condition for adoption. Presented here are two satisfiers, that is, if the dissatisfiers are addressed then these models positively explain adoption.

1. Two good reasons

The basis of selection for these two reasons 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 times/conditions. Such a contingency style ("What is critical for adoption?" "It depends!") reveals far more than any set of 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.

1.1. The first model

This model is taken from Repenning [13]. The explanation of process adoption relies on Figure 1, below. The grammar of the diagram was first popularized in Senge [19], 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.

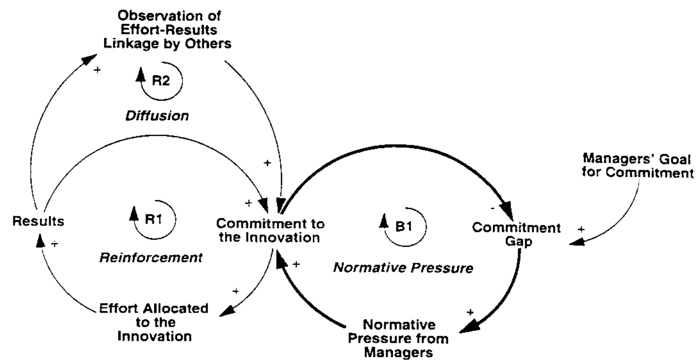

Figure 1. Arrangement of the dynamic forces of implementation. (from [13], pp. 109-127. Reprinted by permission of the Institute for Operations Research and the Management Sciences (INFORMS))

- 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 [14], 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 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.[1]

1.2. 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.
- Without the insight gained by using the model we are unlikely to succeed on intuition alone.

1.3. The second model

In her article, Markus [8] 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) [4], or the intended targets of the improvement (we humans) have some inherent reason to resist the implementation [17]. 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

[1] 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.)

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 [1,5,18].
2. Political: it's about power, who has it, and who loses and gains with the introduction of the new stuff.

In Table 1 Markus frames her insights in terms of resistance. Like any good theory, these three can be used to predict where to look for problems and solutions.

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 (normative) 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.

	People-Determined	System-Determined	Interaction Theory
Cause of resistance	Factors internal to people and groups	System factors such as technical excellence and ergonomics	Interaction of system and context of use
	Cognitive style Personality traits Human nature	Lack of user-friendliness Poor human factors Inadequate technical design or implementation	*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 [8], pp. 430-444. (c) 1983 ACM, Inc. Reprinted by permission.)

	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	Fix technical problems, resistance will disappear	Changing individuals and/or fixing technical features will have little effect on resistance
	Job rotation among resistors and nonresistors	Improve system efficiency Improve data entry	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[8], pp. 430-444. (c) 1983 ACM, Inc. Reprinted by permission.)

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 [2,10,11,12,20].

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 [6], 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 [21], 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)

1.4. Advantages of the second model

Like the first model, this one incorporates other theories [9], 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. [3]

2. Implications

As designers of processes and tools that we want adopted by others, we should understand that there is only so much power in the technical content of our processes and tools. As change agents, that is, implementers of processes and tools, we should understand that the contours of the process and tool are basically dissatisfiers, factors to be overcome, and that we should turn our attention towards the human aspects, especially the collective aspects, of implementation. Power and how our focus shifts over the period of adoption trump technical features every time!

Several examples may help to illustrate this dichotomy. Imagine a software engineering tool that aids component reuse by keeping track of in which programs/classes the components are used and in which version or variant. While this seems innocuous enough, because it appears to be a central repository it would have to fit into an enterprise that is centrally organized. Trying to fit a central repository of component use into an organization that is decentralized would be a challenge, even though we might all agree that the tool is inherently useful. That is, while it has technical merits it cannot be implemented in certain kinds of organizations. Or, imagine a tool that finds errors in static text and is free (such as `lint`). Clearly this is useful, has relative advantage. But it upsets the power structure because it points out defects. What programmer would want to have a list of his/her defects that could be used against him/her – particularly at the height of a career? Yet most advice about implementation says to recruit the opinion leaders, the professionals who are respected precisely because they are at the height of their careers!

What to do? Realize that many tools and processes are point solutions, meant to be inserted into a much, much larger context, one that may not be hospitable. Therefore, point solutions need to be integrated from the start into their larger environments and tested for value in that context.

3. Acknowledgements

The paper this one was based on has benefited from improvements suggested by Eric Busboom, Ray Fleming, Suzanne Garcia, Robert Glass, Watts Humphrey, Philip Johnson, John Kunz, Ray Levitt, Steve Ornburn, Mark Paulk, Shari Lawrence Pfleeger, and John Tittle. I am especially grateful to Marvin Zelkowitz for letting me express some thoughts that had been brewing for a long time.

4. References

[1] 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.

[2] Burrell, G., and Morgan, G. (1979). "Sociological paradigms and organizational analysis." Heinemann, Portsmouth, NH.

[3] 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.

[4] Fitzgerald, B. (1996, January). Formalized systems development methodologies: A critical perspective. *Information Systems Journal, 6*(1), 3-23.

[5] Harris, M. (1996, March). Organizational politics, strategic change and the

evaluation of CAD. *Journal of Information Technology, 11*(1), 51-8.

[6] Kaplan, A. (1964). "The conduct of inquiry: Methodology for behavioral science." Chandler Pub. Co., San Francisco.

[7] Lopata, C. L. (June 1993). *The cooperative implementation of information technology: a process of mutual adaptation*. Unpublished doctoral dissertation, Drexel University, Philadelphia, PA.

[8] Markus, M. L. (1983, June). Power, politics, and MIS implementation. *Communications of the ACM, 26*(8), 430-444 .

[9] 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.

[10] 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.

[11] Meyerson, D., and Martin, J. (1987, November). Cultural change: An integration of three different views. *Journal of Management Studies, 24*(6), 623-647.

[12] 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.

[13] Repenning, N. P. (2002, March-April). A simulation-based approach to understanding the dynamics of innovation implementation. *Organization Science, 13*(2), 109-127.

[14] Richardson, G. P. (1991). "Feedback thought in social science and systems theory." University of Pennsylvania Press, Philadelphia, PA.

[15] Rifkin, S. (2003). Why software processes are not adopted. *In* "Advances in Computers" (M. Zelkowitz, Ed.), vol. 54 ed.. Elsevier, New York.

[16] Rogers, E. M. (1995). "Diffusion of innovations." (4th ed.) The Free Press, New York, NY.

[17] Roth, G., and Kleiner, A. (2000). "Car launch: The human side of managing change." Oxford University Press, New York, NY.

[18] 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.

[19] Senge, P. M. (1990). "The fifth discipline: The art & practice of the learning organization." Currency Doubleday, New York, NY.

[20] 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.

[21] Weick, K. (1979). "The social psychology of organizing." (2nd) Wiley, New York.