Designing Solutions for Organizations that are Increasingly Focused on Knowledge Work

Fujitsu North America, Platform Products Group

Keith D. Swenson

Abstract: System architects design systems to help an organization get its work done, but to do that effectively requires having an accurate understanding about how organizations work. Tradition leads the system architect to think of an office as a factory, a kind of machine, and that information systems are a kind of automation to make such work entirely automatic. As such automation succeeds, it changes the structure of the organization. Since workers have been freed from the routine, repeatable tasks, they now spend more time thinking and innovating. Companies are shifting dramatically to become knowledge worker environments. Knowledge workers are innovative, and require a dramatically different approach from automation. This paper explores the different technology options that are available today, and how best to use them. Innovative organizations are continually reinventing themselves. This individual experimentation is at odds with a system that constrains action to predefined patterns. Organizations are complex, and display antifragile qualities. Like other adaptive systems, organizations need a bit of change and stress in order to remain healthy. As more organizations become agile, there are new choices for system architects and a new generation of tools, such as adaptive case management, that support innovation and learning organizations.

Keywords: Case management; business process management; knowledge workers; organizations; process technology; innovation; learning organization; antifragile

1. Antifragile

Organizations hire knowledge workers so that they will think outside the box.[6] However, a Business Process Management (BPM) system used to automate work is often the box they are hired to think outside of. This paper will consider process technology that can be used for rapid implementation of business systems, as well as the needs and requirements they have, but before we do that, we need to step away and consider the nature of the organization. It is common to think of a business in terms of a factory model that consumes one kind of resource, and produces another. Such a model then simply needs automation. But this model is invalid for knowledge work. Before we can understand this directly, we need explore the concept of *antifragile*. This term was coined by Nassim Nicholas Taleb in his 2012 book "Antifragile: Things That Gain From Disorder."[1]

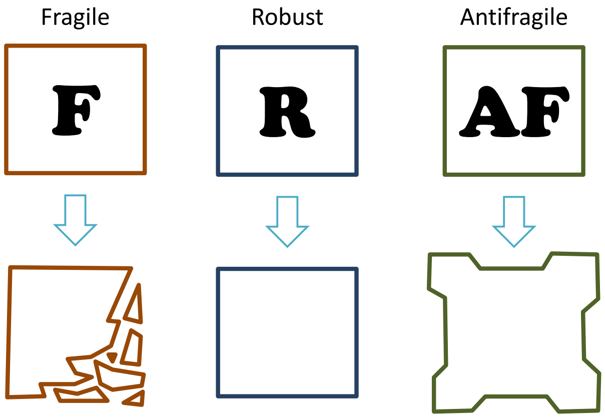

Fig. 1: Depiction of Fragility Spectrum

We all know the meaning of *fragile*: When you stress something that is fragile, it might break. What is the opposite of fragile? Most people will readily suggest that the opposite of fragile is *robust*. Something that is robust is something that, when you stress it, it does not break; it remains the same.

There exist things that are less fragile than robust. When subjected to stress, these things not only resist change, they actually grow and get stronger. They actually get better when subjected to stress, and remain better after the stress is removed.

This notion is strikingly non intuitive. It is common sense that everything around you eventually wears out and breaks down. It may break suddenly, like a porcelain teacup. Friction will cause the bearing of a wheel to wear down and fail. Wind on a canvas tarp will work the material and eventually rip the weak spots. We simply know at an intuitive level that stress always causes things to wear out.

Consider muscles. If you exercise, the result will be increased size and strength of the muscles used. To learn to play the piano, you practice. Reading a book on piano technique is not effective. Only by actually sitting at the keyboard and working through songs will you gain proficiency. To learn to play tennis, you have to get out on the court and start hitting balls.

Learning in general is antifragile. Quizzes and exams are purposeful stresses that help to prepare a student for when they will have to face real situations. Performing a fire drill is clearly an unwanted extra stress that takes people away from their main job, but the result will be an organization better prepared for such an emergency. A fire drill teaches these behaviors far faster and more effectively than any amount of textbook learning.

Things that actually get better as the result of stress are known as adaptive systems. Antifragility is a property that emerges from a complex adaptive system. Complex adaptive systems are all around us; ecosystems, biological systems, organizations, marketplaces, social networks, the economy, even our own muscles and brain. These do not behave like machines.

2. Machines

Complicated machine are made out of components, each providing a specific set of functions. Each part is made very precisely to perform that function. Machines that last are made as a hard and durable as possible. Parts fit together as perfectly as possible, with just the right gap to minimize friction and other degrading forces.

A good analogy for an idealized machine is a luxury mechanical watch. There are many gears, each made to fit precisely together with the other gears. The better watches have a jewel movement, which use a very hard stone at the pivot points. Built correctly, the watch will run for a very long time, and be very accurate.

It is rather obvious that machines are not adaptive systems. No matter how many times you run out of gas, the car does not get better about conserving fuel! (However the driver might.) Less obviously, we should understand that organizations are not machines, even though we like to think of them as machines.

Organizations can learn and flexibly adapt to situations. The roles that people play are not like the parts of a watch. People routinely fill in for others while they are away on vacation. The introduction of a new CEO, with a different management philosophy, can have the effect of redefining many jobs in the company, without any explicit orders being given. Organizations do not wear out; they may come to an end in many different ways, but they never simply wear out. Organizations routinely do many things that a machine could never do.

3. Stability

In many ways, the purpose of an IT system is to help support the stability of the organization. A well-functioning IT system will help smooth out the peaks and valleys of the business environment, and allow the personnel to perform more effectively. The organization gets more done because it uses its existing resources better.

Organizations do not achieve stability the same way that machines do. Remember, mechanical stability comes from designing parts very precisely and forming them from very hard materials. Even so, this stability is a temporary thing: the machine will eventually wear out.

An adaptive system achieves stability through *homeostasis*; this stability comes from a balance between different adapting forces. In an ecosystem, good weather may cause an increase in vegetation. In response the population of grazers might increase. Later, the population of predators might increase as well. The next year weather might be less productive, and grazing populations would be down, and so would the predators. These population proportions are not maintained by any central plan, but instead by a balance of different adaptive forces working off each other.

Thinking that adaptive systems should be treated like machines is a large part of what I call the *enlightenment bias*. This is a way of viewing the world using ideas from Descartes, Newton, and other Enlightenment philosophers who promoted the idea that behind every complicated phenomenon is a set of simple rules that define the behavior. These ideas were revolutionary at the time and led to a dramatic expansion in understanding of natural laws. These ideas were applied to management with the advent of Scientific Management where large complicated operations are seen to be decomposable into smaller, simpler steps that can be precisely and rigorously defined. Scientific Management is a part of our culture. We all learned that you should first plan, and then act. If you fail to act, then the fault can be attributed to poor planning. Plan better and you will act better in the future.

The ultimate expression of the enlightenment bias is in BPM systems where management attempts to define every possible detailed action that workers might take, and to find the optimal sequence of these actions. These system architects envision the organization as a kind of machine. They are trying to define very precise and very durable parts for that machine. This works for automating routine processes, but more and more organizations are turning to support for knowledge workers who do work that is anything but routine. At the level of knowledge workers, the organization is not a machine. It behaves more like an adaptive system, and applying machine principles can actually harm the organization.

"The only sustainable competitive advantage is an organization's ability to learn faster than the competition."
- Peter M. Senge[8]

4. Adaptive Systems Crave Stress

Adaptive systems not only respond well to stress, they actually need stress. This seems surprising when stated that way, but we already know of many examples around us.

If you don't use muscles, they atrophy; they shrink and become weaker. A large muscle uses resources, and that is a waste if a large muscle is not necessary. Growing and shrinking muscles are the balancing adaptive forces that allow the body to optimize resource usage. Yet if muscle strength declines too far, it is possible to be injured by something that a normally healthy person would not be hurt by. So exercise is an important part of remaining healthy.

If a forest is protected from fires, the undergrowth grows up, and makes the forest more susceptible to fires, and if a fire breaks out it is likely to do far more damage. The policy of preventing all fires in a forest has had the disastrous consequence of indirectly causing far larger and more damaging fires that are harder to recover from. In a very real sense, protecting a forest from fire makes it grow weaker. To maintain the strength of a forest, you need to have regular, modest sized forest fires.[7]

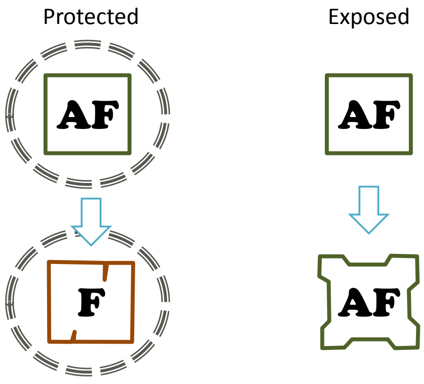

Fig. 2: Protected and Unprotected Antifragile Systems

Here is the surprising conclusion: adaptive systems need a certain amount of stress. If they are protected from all stress they become fragile.

This rule applies to organizations. Fire drills are required to prepare for a fire. Suspending all fire drills will cause the organization to be less prepared for fire. Emergency response teams that do not drill themselves on different simulations and scenarios in advance would find themselves ill-prepared to meet the next unexpected situation.

If a football team wants to win the championship, it does so by practicing and playing many practice games. Spending time sitting still, resting, and conserving muscle movement would decrease the chance of winning. Unlike a machine, a team is an adaptive system, which gains from being exercised, and is actually harmed if it does not exercise.

Such exercise must include variations. Olympic swimmer Michael Phelps was forced by his coach to swim in unexpected circumstances: sometimes with the lights turned off, sometimes woken up in the middle of the night without warning.[4] This intentional variation in training has been credited with his ability to win a gold medal in the Beijing Olympics even though his goggles cracked and filled with water. [3]

The central point of Taleb's book was that antifragile systems crave stress. While it is able to readily accommodate modest perturbations, if you protect the system from those changes, attempting to provide a static environment, then the system becomes fragile and dangerous.

Business teams crave stress as well. Running simulations and scenarios as a business team is a well-known way to improve team performance. Doing things differently allows the team to understand how to coordinate on the fly. Then, if a situation comes along where the team has to do things differently, they are more prepared to meet this challenge.

5. The Best Practice Conundrum

It is ironic that the very purpose of most business process management initiatives is to find the single best practice, and institute that best practice by forcing employees to follow it. Here we get to the central theme of this paper; enforcing a single best practice on an organization *can make it fragile*.

This can be hard to understand for a system architect who thinks of an organization as a machine. After all, a diesel truck will perform best at a particular speed in a particular gear. A truck driver wants to find that speed and gear and use it whenever the situation is favorable. But organizations are more like muscles than trucks.

A swim coach searching for the single best practice might conclude that an athlete swims best at 2:00 in the afternoon, after sleeping to 10am and when the pool temperature is 72 degrees. Instituting and enforcing this best practice might leave the swimmer exposed to the risks of performing poorly in competition if it is impossible to achieve the optimal preparation and environment.

The same thing happens in business teams. If a process is put into place that enforces that 'A' is done first, then 'B', and then 'C', the people working in the office come to expect it to always be this way. By acclimatizing to always having this pattern, the organization loses the ability to handle cases in any other order.

In a complex adaptive system constant stress is not to be mistaken as overreacting to noise but must be understood as environmental tuning information. We need to re-learn that in a complex world the notion of a single logical cause or a predictable outcome of an action is suspect. Constant, random stress is information that aligns the small anti-fragile system with the changes in its environment

This is not really a new idea. Management guru Tom Peters' 1988 book "Thriving on Chaos"[2] discusses organizations that thrive on the churn and turmoil around them. It is common to suggest that an organization needs occasional "shaking up" to keep it healthy. Agile software methodology works on understanding that software development is complex and unpredictable, and does not try to define everything perfectly in advance.

The system we speak of here is the *organization*, not just the computer system. The computer system may or may not be adaptive on its own. Our goal is clearly to make the business run better, and that involves people as well as the computer system. Organizations are adaptive, and it is the role of IT systems to support that adaptiveness.

"A military force has no constant formation, water has no constant shape: the ability to gain victory by changing and adapting according to the opponent is called genius"
- Sun Tzu

6. Planning as Part of Work

Part of the reason for attempting to identify and isolate the single best practice is to eliminate the need to spend time planning what to do. If there is a fully elaborated best practice, then there is no need to waste time planning. Planning is viewed as a waste, and if planning can be eliminated, then workers can spend all the time doing productive work.

"No plan survives contact with the enemy."
- Helmuth von Moltke the Elder
"Planning is essential, plans are worthless."
- Dwight D. Eisenhower

The military is the place where you might expect to see the most rigorously defined and standardized modes of operation, but these respected leaders go out of their way to stress the importance of the planning activity itself. The importance is not just the end result–the plan–but the actual activity of planning itself is important.

Translated to modern terms, it is almost as if Eisenhower was saying that it is important to model your business processes, but when you are done you can throw the resulting models away. It is not the models that have value, but the activity of doing the modeling is worthwhile.

From this we can conclude that planning itself should not be eliminated, but in fact should be done as part of work. A best practice should be enforced without question, but instead as a guideline that might, or might not, be followed. There should be a point where the team sits down and evaluates whether the best practice is going to work in this case, and if not, to come up with an alternative. Planning needs to remain part of what the knowledge worker does.

7. Advances in Information Technology

System architects have a wide spectrum of technology available to support business. Each technology approach is useful for a different kind of business problem. In order to organize and make this easier to understand, I have organized them according to the degree of predictability of the business problem they solve. At the left end you have entirely predictable work which needs to be done exactly the same way every time: the process has always been done that way for years, and (probably) always will be done that way. The other end of the spectrum is complete unpredictability where there is no way to know from moment to moment what will have to be done next.

Predictability and repeatability go hand-in-hand. Any work which is repeated the same way thousands of times, is predictable by definition. Work that is not done the same way every time, that is frequently repeated, is consequently less predictable. Something that is done only once in history is the most unpredictable of all.

The approach to developing any system will depend on how much change the system will have to respond to over time. Extremely predictable, stable environments can benefit from powerful but inflexible approaches. As the anticipated amount of change rises, it becomes more important to use a technique which offers greater flexibility. More flexible approaches have less precision to exactly match the needs of the situation. The approach depends entirely on the amount of change.

Most job situations lie between the extremes of completely predictable/repeatable and not predictable/not repeatable. We can break the field into seven domains according to the technology that might be used to support workers:

1. Traditional Application Development is not a special process approach, but rather the absence of process technology. If work is very predictable and stable over time, one can use traditional development techniques (e.g. using any third generation language like Java, VB, PHP, etc.) to create a supporting application. The cost of development might be high, but the benefit of having very precise control of the capabilities will yield efficiency that over a large number of cases will pay back the up-front costs.

2. Process-Driven Server Integration (PDSI) is a form of programming that uses process models to provide flexibility to cope with changes in the distributed server environment. These process models deal with low level data; pick up a record from one place, transform it, and send the result to another place. The modeling is done by a programmer who understands data structures and transforms, although a business person might be able to review the model for conceptual correctness. Straight-thru-processing (STP) and Enterprise Application Integration (EAI) are other names for this category.

An example of a business problem in this domain is fulfilling the purchase of a cell phone: coordination is needed between one system to allocate a phone number, another system to set up an account, another system to order the phone, and yet another system to arrange for delivery. Ideally the process model automates this completely and quickly.

3. Human Process Management (HPM) uses a process map showing actions assigned to people. Humans are in many ways less predictable than servers. In PDSI, the process will decide which server to handle a task, send the request and 99.95% of the time it will be done. However, for humans, it is not uncommon for a task to be assigned to one person, then reassign to another, then forgotten about until a reminder is sent, and then finally complete by someone else who works with the assignee. Human process management is designed for the idea that one can't state in advance exactly who will do the task.

Usually this technology has strong support for roles which is a way of directing the assignment to a structure that can be easily changed from day to day, without having to edit the process. Humans need a task list in order to decide which task to do next, deadlines to indicate that something has been sitting too long, and reminders to actively prompt for something that is late. Escalation is a feature that allows a task to be automatically reassigned if it takes too long. It is worth noting that none of these features are needed in the PDSI domain: if a server fails to respond, then sending a reminder will have no effect.

A good example of HPM is expense report approvals: there are a number of people involved who do different tasks like approving. There are reminders if people are slow, and tasks can be reassigned if someone changes position.

4. Production Case Management (PCM) is for when the process itself cannot be completely defined in advance, but the set of possible actions one might want to do can be defined. The user has to decide, based on experience, what the right next thing to do, and to pick that from a menu of possible actions available at that time.

It is called *production* case management, because it is designed for high volume situations. Like Human PM there is a separation between the people who determine the possible actions (the developers) and the people who use the actions (the users).

A good example of a business problem needing PCM is a help desk or customer support center. There are a set of possible actions, such as refund the customer, order a replacement, escalate to development, etc. The customer support agent is there primarily to answer questions, but if warranted, can call one of these steps into play. The agent is a knowledge worker who learns the specific trouble modes that people are likely to encounter, and learns to determine the right course of future action for each case.

5. Adaptive Case Management (ACM) - not only are the processes unpredictable, but even the actions that need to be done are not known in advance. Knowledge workers can actually create new goals that have never been needed before.[5] An ACM system allows knowledge workers to experiment with new ways of working. It does not constrain the workers to any given business pre-defined process. The process can be changed by any participant, and changing the process is a natural part of everyday activity.

When we say that the process can be changed by any knowledge worker, it is necessarily understood that no special skills or knowledge must be necessary for making these changes. Users must be not only be *allowed*, but also *able* to make those changes. This requirement rules out most of the more formal ways of modeling processes which require specialized training.[13] The process must be expressed in a way that a completely untrained knowledge worker can modify at will. One might think of this not as process modeling, but instead *planning*. Knowledge workers don't work on pre-planned units of work, but instead planning itself is part of doing the job.

An example needing ACM is that of a doctor defining a treatment plan for a patient with a particular set of symptoms and history. The treatment plan is a kind of process, but the doctor cannot wait for a programmer to implement it. In an ACM system, the doctor is able to create the treatment plan directly and without IT help.

6. Social Business Software (SBS) is the last category between ACM and email: it is a domain that has very little or no explicit process support, however there is a greater amount of collaboration and time sequencing than manual email. This domain has people collaborating on permanent artifacts, and often using network connections to control access. This is collaborative software, and it includes basic document management systems without a fixed plan. There may be representations of goals, but they are created on the fly and discarded after use.

7. Email, Telephone, Texting is on the right end of the spectrum, and has no process support at all, no permanent structures, simply communication. This is the default that many current processes are forced to use, but this approach puts the greatest burden on the user, and yields the least amount of analytic data to monitor and improve processes.

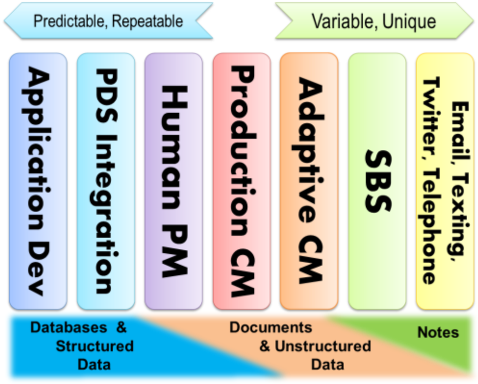

Fig. 3: The seven domains of predictability

Increasingly, managers are finding that it stems from the "second-order" organizational capabilities that foster rapid adaptation. Instead of being really good at doing some particular thing, companies must be really good at learning how to do new things.

- Martin Reeves and Mike Deimler [19]

8. Knowledge Work in Learning Organizations

From this extensive line-up of process technology, which one is appropriate for an antifragile organization? Each domain of predictability is distinguished by the amount of investment into preparation that must be done before you start production work. Traditional programming requires a large development project and is useful only after the entire project is completed and the fully-tested software is installed. At the other end of the spectrum is email, testing, or phone calls need no preparation at all, and can be used immediately without delay. It stands to reason the amount of up-front investment one is willing to make, is correlated to the predictability of the business problem. When you make a large investment in a fixed process, then the cost of changing that process is also higher, which makes it even more important to get your prediction correct the first time.

In every domain the desire to improve the process is expressed continuous cycle of improvement: model, implement, perform, measure, analyze, and cycle around to improve the model, but each domain realizes this cycle with dramatically different scales. In traditional application development the cycle may take 6 to 12 months. PDSI and HPM the cycle time may be weeks to months. However, in ACM, because the end users can modify their own plans, the cycle time for process change can be very quick: measured in days. It is the fast process improvement lifecycle, and the relatively loose controls, that allow ACM to be appropriate for a learning organization.

The left-most three (TD, PDSI, HPM) all require that the process be known in advance and coded in a process diagram, leaving little or no room for innovation. An innovative organization is strong because it reinvents itself, which means that the individual knowledge workers are finding new working patterns. PCM gives the worker more choice to choose the course at run time, but there is no way to be innovative about working patterns, and to learn new paths. ACM is the first domain (from the left) that incorporates planning into the work. This planning activity is where new, innovative working patterns are invented and instituted.

ACM allows variation in working patterns to be introduced at run time by any participant. This means that when a knowledge worker sees an opportunity to try something new, they can do so immediately. All of the technologies to the left of ACM constrain user actions to those that are pre-defined. Generally this constraint is justified as protecting the user from making mistakes and deviating from the best practice. However, section 4 above explains that protection of an organization tends to make it fragile and dangerous. For knowledge workers, protecting them from mistakes prevents an organization from learning. For unpredictable processes, instead of trying to make fail-safe systems, you get better results with safe-fail systems [18] that allow people to try innovative new actions without risking catastrophic failure. The organization would work more like an ecosystem.

"Nature loves small error, humans don't -- hence when you rely on human judgment you are at the mercy of a mental bias that disfavors antifragility."
- Nassim Nicholas Taleb [1]

Allowing knowledge workers to plan their own processes for each case is an approach that is very hard to accept by those who view an organization as a machine that operates on a set of simple principles. It runs contrary to the idea that there is a single best way to do something, and our goal is to find the one best way and make sure that everyone does it.

The Workflow Management Coalition runs the "Excellence in Case Management Awards" program to recognize excellent uses of PCM and ACM today. [16,17]

Regardless of the challenge, a system architect must come to see an organization as an antifragile system, which craves stress to remain healthy, in order to be successful at designing a system to support knowledge workers. Some organizations understand this today. Upper management must come to realize that, as Sun Tzu put it, their organization has no constant formation, and they must stop putting in place fixed, rigid processes, or they will find themselves left behind by those whose organizations are like water, shifting to meet each new challenge.

"The future is uncertain—but this uncertainty is at the very heart of human creativity"
- Ilya Prigogine

9. SUMMARY

Fragile: the quality that when disturbed has a propensity to break. Stressing a fragile object reduces or destroys its value.

Robust: the quality that when disturbed it remains the same. Stressing a robust object has no effect on it at all.

Antifragile: the quality that when disturbed it improves. Stressing an antifragile object actually makes it more valuable. Antifragility is a quality that emerges from an adaptive system. While it sounds crazy, there are adaptive systems all around us, and a human organization is one of those.

Not only do adaptive systems respond well to stress, they actually degrade when all stress is removed. Like muscles that need exercise, an organization needs a certain amount of variation in order to remain healthy.

Seven technologies for handling business problems were outlined organized according to predictability of the business problems. The more predictable, the more one is willing to invest in development of the business process.

Adaptive case management is an approach to supporting knowledge workers that does not constrain the working patterns to a predefined best practice. Instead, it allows knowledge workers to evaluate what the options are in this case, and to plan a course of action that might be unique for this case. It then focuses on communications about the plan, and in support of the plan.

Scientific managers and system architects who view their organization as having a fixed form with a single best mode of operation may find this approach uncomfortable. Experienced managers already know that knowledge workers are not simple gears in a clock, but are instead capable of the most if they can be allowed to experiment and find new innovative approaches to work. Case management is an approach to finally bridge this gap between business and IT for the support of knowledge workers.

10. References

[1] Nassim Nicholas Taleb, "Antifragile: Things That Gain From Disorder.", Random House, November 27, 2012

[2] Tom Peters, "Thriving on Chaos", Harper Paperbacks, 1987

[3] Scott Eblin "What Leaders Can Learn from Michael Phelps About Preparation" http://eblingroup.com/2012/08/what-leaders-can-learn-from-michael-phelps-about-preparation.html

[4] Sanyin Siang, "You Can Only Win in Sports, or Anywhere Else, if You're Ready for Chaos", Forbes Magazine, 8/14/2012

[5] Keith D Swenson, "Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done" Meghan-Kiffer, April 2010

[6] Peter F. Drucker, Management Challenges for the 21st Century. New York: HarperCollins, 1999

[7] Alex Kirby, "Why forests need fires", BBC News, December 2002, http://news.bbc.co.uk/2/hi/science/nature/1747062.stm

[8] Peter Senge, "The Fifth Discipline: The Art & Practice of The Learning Organization" Doubleday; Revised & Updated edition (March 21, 2006)

[9] Michael White, Case Management: Combining Knowledge with Process, BPTrends, July 2009.

[10] Wil M.P. van der Aalst, Mathias Weske, Dolf Grunbauer, Case handling: a new paradigm for business process support, Data & Knowledge Engineering 53 (2005) 129–162

[11] Sandy Kemsely, "It's Not About BPM vs. ACM, It's About A Spectrum Of Process Functionality", www.column2.com , 2011.

[12] Keith D Swenson, "Seven Domains of Predictability", http://social-biz.org/2013/06/10/seven-domains-of-predictability/ 2013

[13] Keith D Swenson, "BPMN is incompatible with ACM" ACM 2012 1st International Workshop on Adaptive Case Management and other non-workflow approaches to BPM, BPM2012 Tallinn, Estonia, September 3, 2012, pp 85-88

[14] Hamid R. Motahari-Nezhad, Keith D Swenson, "Adaptive Case Management: Overview and Research Challenges" unpublished, submitted to IEEE Conference on Business Informatics, July 2013

[15] Keith D Swenson, "Designing for an Innovative Learning Organization" unpublished, accepted to The 17th IEEE International EDOC Conference, Vancouver, BC, September, 2013

[16] Layna Fischer, "How Knowledge Workers Get Things Done", Future Strategies, 2012

[17] Layna Fischer, "Taming the Unpredictable", Future Strategies, 2011

[18] David Snowden, "Safe-fail or Fail-safe", http://cognitive-edge.com/blog/entry/4501/safe-fail-or-fail-safe/ Sept 2006

[19] Martin Reeves, Mike Deimler, "Adaptability: The New Competitive Advantage" Harvard Business Review, July, 2011