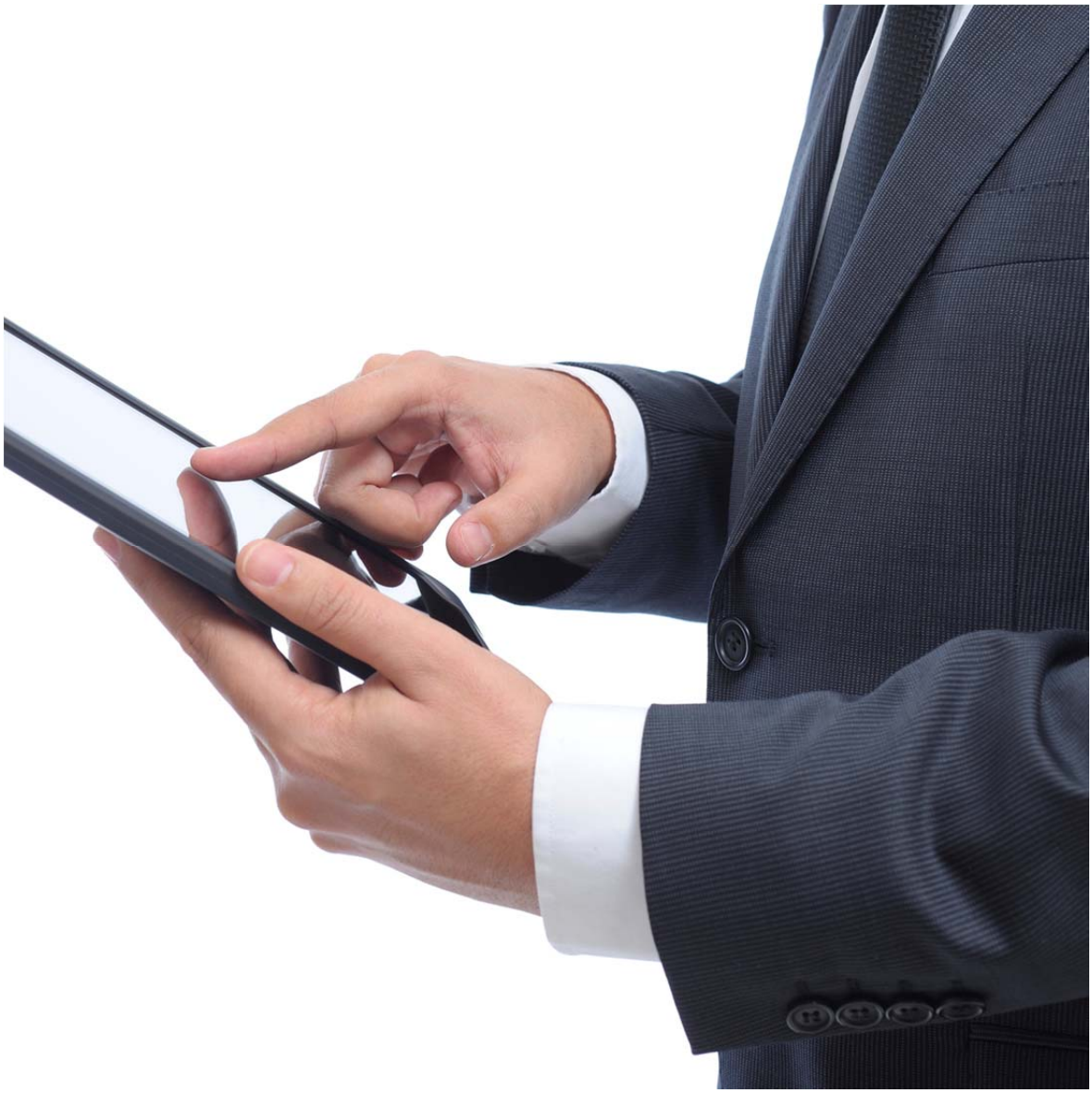
FUJITSU

White paper
State of the Art In Case Management

Keith D. Swenson
Document version 1.0
Date: March 2013



This page intentionally left blank

Table of Contents

The purpose of case management is quite simply to help knowledge workers get things done.

How did knowledge work come to be the center of attention for new IT adoption? Much of the focus for information technology deployment for the past two decades has been on automating or even eliminating less skilled jobs. This has been largely effective, and organizations today are able to do far more with fewer people. Workers today spend less of their time on routine tasks, and more of their time on things that really require thinking, than was possible just ten years ago.

The challenge today is how to support higher skilled modes of work: knowledge work. We can also call this kind of work "unpredictable work" because one cannot predict in advance the exact course of what will be done. It requires thinking in order to figure out what to do. The exact course of what needs to be done cannot be known in advance, and this is the central challenge to the traditional way of designing IT systems.

The name "case management" is used to talk about an approach that supports the knowledge worker, without requiring that the work be constrained to a set of pre-defined actions. This is technology for the rest of us: managers, decision makers, executives, doctors, lawyers, campaign managers, emergency responders, strategists, and many others who think for a living. It is for people who figure out what needs to be done, at the same time that they do it.

This is not only a technology trend. We are seeing a fundamental shift in our workforce, and in the ways they are managed. Not only are companies engaging their customers in new ways -- using social media, mobile computing devices, and social networks -- but managers are engaging workers in similarly transformed ways. The office is being transformed from an assembly line for the processing of forms, to far more agile and effective patterns for accomplishing organizational goals. This is enabled by recent technology developments, and at the same time demands new technology approaches.

Case management – A method or practice of coordinating work by organizing all of the relevant information into one place -- called a case. The case becomes the focal point for assessing the situation, initiating activities and processes, as well as keeping a history record of what has transpired.

Knowledge workers – People who have a high degree of expertise, education, or experience and the primary purpose of their job involves achieving goals in a way optimized by that knowledge.

Developer – Someone with skills in creating IT applications. A developer is a knowledge worker, but one with skill in the area of programming, and not primarily in the area that the application is produced for. We distinguish between the knowledge workers doing the work, and the developers who help create applications.

Knowledge Work

Knowledge worker productivity is the biggest of the 21st century management challenges. In the developed countries it is their first survival requirement. In no other way can the developed countries hope to maintain themselves, let alone to maintain their leadership and their standards of living.
- Peter F Drucker

Knowledge workers have a high degree of expertise, and the primary purpose of their jobs involves the creation, distribution, or application of knowledge.
- Thomas Davenport

Knowledge work is work that requires thinking. It requires not just skill, but also expertise. Knowledge workers make decisions like executives or managers. Knowledge work involves putting facts together like a detective. Knowledge work requires experience with the details of the situation in order to make the right plans. Knowledge workers are all around us: all levels of management; lawyers taking a case to court; judges presiding over those cases; elected representatives crafting new legislation; detectives following up on a crime; business people drawing up a new plan; stock traders cornering the market; product designers determining the features for a new product; marketing staff deciding a media campaign. All of these require specific insight into a situation in order to make the right decisions for success.

Peter F Drucker made the first reference to knowledge work in 1959. He calls attention to the uniqueness of each knowledge worker's job when he describes a knowledge worker loosely as "*someone who knows more about his or her job than anyone else in the organization.*" Each knowledge worker may need to do things different ways, depending upon the specifics of the situation. Ironically, information systems have focused in past on making everyone work in exactly the same way.

Drucker also reflects that knowledge, and by extension knowledge work, is constantly in flux. "*Knowledge is different from all other resources. It makes itself constantly obsolete, so that today's advanced knowledge is tomorrow's ignorance. And the knowledge that matters is subject to rapid and abrupt shifts-from pharmacology to genetics in the health-care industry, for example, or from PCs to the Internet in the computer industry.*" Jobs based on knowledge will change when the knowledge does. Processes that depend upon knowledge, and at the same time produce knowledge, have a compound dynamic which makes them especially difficult to manage.

The opposite kind of work, routine work, is predictable and repeatable. Routine work is something that is done a particular way, over and over. It is just a matter of time until all routine work becomes automated. Because knowledge work is not predictable, and therefore difficult to automate, we find that the work force has to shift to do more knowledge work. Across all industries, the percentage of routine work is diminishing and the percentage of knowledge work is growing. The working population is spending more time thinking, and needs more support for this.

Unpredictable Processes & Agility

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

Knowledge work is described as being unpredictable. Unpredictability does not mean simply that there is variation in the details of the work. The process of fulfilling an online book order may be considered predictable, even though the size of shipment and the destination vary significantly from order to order because these factors don't affect the pattern of things that must be done to complete the shipment. Even if the outcome is uncertain, such as bidding at an auction, the process might still be considered predictable with branches representing the various courses of action.

By unpredictable, we mean that the sequence of significant human acts is not knowable in advance, and the course may vary from case to case in new and unexpected ways. The course will depend greatly on details of the situation itself, and the details may change before the work is finished. More importantly, while many steps in the sequence might be normal routine steps, it might be impossible to know ahead of time which steps will be needed, and how many times. There is always the possibility of needing to do something that has never been done before.

Note also that any prediction that an organization makes on the course of events has to be based on actual information that they know at the time. Hindsight convinces us that if we had known more information up front, then we would have been able to predict the course of events. One of the limitations that organizations face is that is it not possible for them to be aware of everything which is knowable at that time. For instance, when a person arrives at an emergency room unconscious and without identification, their medical history surely exists somewhere, but without identification or knowing their normal care provider it remains at least temporarily inaccessible, even though the emergency care must be provided without delay. A theme we will return to many times is that the case manager must move forward even when they know they have only partial information, even when they know they will receive more information later that may change their plans. They cannot wait until they have all the facts to make a decision.

Consider the case of a hospital accepting a person for care who has been in a car accident. The patient has to be accepted and care has to start before any detail about their condition is known. They will start with some investigative procedures: vital signs, a blood test, examination of wounds, etc. Then, based on what they learn, they will prescribe an initial treatment, but that is not the end of it; they will continue to monitor. The signs of a blood clot in the brain may not show up for many hours. When it does, that may warrant emergency surgery but there are many factors that need to be considered due to the condition of the other wounds. The point is that nobody could say at the time of admission that this patient would need surgery; the information necessary to make such a prediction could not possibly be known.

In an unpredictable world, sometimes the best investments are those that minimize the importance of predictions.
- Gökçe Sargut and Rita Gunther McGrath

Typically knowledge workers will prepare for uncertainty. The hospital is prepared to respond to many types of emergencies. The hospital staff have learned and practiced what to do when particular situations arise, and have even tried to make certain procedures routine. The hospital staff work is not chaotic; it is not simply random tasks. The unpredictability comes from the way they are responsive to the situation. You might think that the only decision is to select the right treatment from a pallet of possible treatments, but the possible options themselves are constantly changing.

An unpredictable process is simply a process that has to be figured out; it emerges as the work is done. That is precisely what knowledge workers do. We call this practice of preparing for many possible courses agility. Knowledge work is everywhere that people determine what to do as they work: executives, managers, problem solvers, designers, etc.

No plan survives contact with the enemy.
- Helmuth von Moltke the Elder

Planning is essential, plans are worthless.
- Dwight D. Eisenhower

These military strategists point out that planning is important because you are forced to think about all the options, but any given resulting plan does not represent that value itself. The plan is just a construct used to organize the thinking, but most important is the thinking itself.

Discussions about case management are often derailed by those who feel that the purpose of an IT system is to eliminate thinking. This is the right thing for routine work – fastening a seat into a car on the assembly line should not be done differently every time. This kind of routine work does not require a lot of thinking, and success would be the complete automation of the work with a robot. Standard, traditional BPM is about automating those routine processes in the office that you want done the same way every time. There is not much thinking involved to create an account for a new customer. Many office activities from 10 years ago are automated today.

Automation is an important part of what IT systems do. But automation is impossible for knowledge work, because the work is not automatable. This is a deep philosophical truth that comes from the complexity of reality. Modern office systems are running up against the limit of what can be automated. What lies beyond is simply not automatable -- not because our systems are not powerful enough, nor because we don't know enough to automate them, but because the work itself simply does not repeat in routine patterns. There is a naive technologist viewpoint that believes that all office work is ultimately automatable. I call this the "enlightenment bias." Enlightenment age thinkers, like Newton and Descartes, introduced the idea that behind every physical phenomenon there is a simple formula that can be used to predict behavior. This works well in physics (Newton's laws of motion, Maxwell's equations) and in many other fields as well. Scientific management applied these ideas to isolate the precise sequence of actions that a worker has to do to accomplish a particular result. We are taught from early days that to accomplish something, you need to plan. If you fail to accomplish, it is because you did not plan well enough. Next time, spend more time on planning, and things will go better.

This enlightenment bias causes a blind spot in the modern technologist, who believes that a plan is the essence of a program that can be run on the servers to bring about the completion of the process. Once again, this is true of routine work, but not of knowledge work. Before you go any further, you must become comfortable with the idea that knowledge work will never be automated. It is not just a matter of working out the details. Knowledge work is fundamentally different because it requires thinking. As the routine aspects of work become automated, you are left with people doing more and more of the knowledge work, but the knowledge work is never eliminated. For example, ERP systems have automated a large part of what accountants used to do, but you still need knowledgeable people to run the accounting department.

Case management does not attempt to replace thinking, but rather to provide a platform for thinkers to communicate and take action through.

Case Management Systems

Case management (CM) is an approach which has been used for years in the fields of health care, social care, the courts, and law enforcement but now we are seeing this same technique being used for all knowledge workers across all industries. We call it case management because all the work is structured around a case, which is simply a place where everything for the job is collected. It is not just a folder, but is also has an implicit or explicit goal. Sherlock Holmes has the goal to solve the crime, so he keeps all the relevant information in a case folder. The case is considered closed when the crime has been solved.

Today, that case is not a physical folder, but instead an information system structure that can be accessed by multiple people from multiple locations simultaneously. It is in some ways like an enterprise content management (ECM) system, except that while an ECM system is designed to be a permanent store of all information for an organization across all time, a case folder brings together all the things only for a particular goal. The contents are focused on relevance for a particular case.

A knowledge worker creates a case in order to accomplish a goal: recruit a new employee, plan a company event, run a conference, address a customer problem, treat a patient's symptoms, investigate a suspected terrorist, resolve a billing dispute, and any sort of non-trivial accomplishment. Part of achieving a goal is identifying and tracking sub-goals that will be steps to completing the case. Those sub-goals may depend uniquely upon every case.

There is always a case manager, a person who is responsible to see that the goal is completed, and that person is designated the case manager. The case manager must have complete freedom to create whatever sub-goals are needed to complete the case. The case management system helps in communicating goals to the performers, and to also facilitate the communications back.

For example, to treat a patient, we would put the results of all tests, and all other relevant information about the patient into a case folder. We also would associate specific people with the case in the roles like doctor, nurse, or other specialist so that they can access the information as they need to, and so that the information is protected from others not involved in the case. The case manager might be a doctor, who comes up with a unique treatment plan designed to meet the needs of this particular patient.

The case brings together all of the information resources that a case requires and it also tracks everything that has happened so that a clear recorded history is retained. This history can be analyzed to find patterns of success, and even in some cases a best practice process might be mined out of it.

This will all sound very similar to many other kinds of IT systems, so let me break out a spectrum of technologies in terms of their ability to handle different levels of predictability. When we do that, we will find that there are two types of case management: production case management and adaptive case management. These will be described in quite a bit of detail later. First, however, let's consider where they both fit in a spectrum of 7 different domains of work predictability.

Seven Domains of Predictability

Technology to support business can be seen on a spectrum of varying degrees of predictability. At one 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 Programming - 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) - Integration patterns between systems can be quite stable in the short term: months or years. Still the systems being integrated do change, and the business needs change as well. PDSI usually incorporates a development approach where the key interaction points are depicted on a process diagram. It is a technique that helps deal with the kinds of change that IT departments experience.

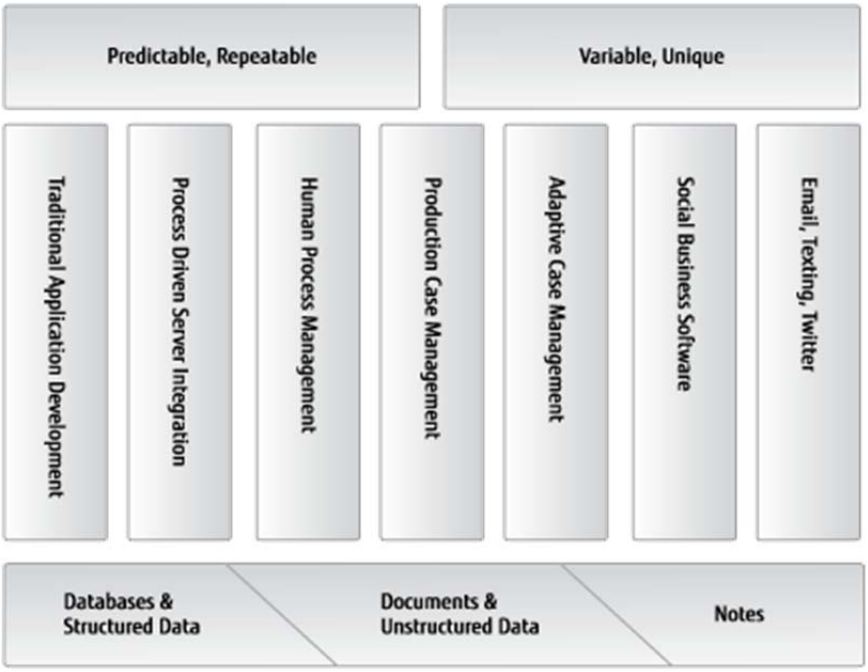
3. Human Process Management (HPM) - This style of integration usually involves forms routed through a set of people. The major shape of the process and the tasks may be well-understood but HPM has capabilities to handle the kinds of continual change that exists in an organization: people joining, people leaving, changing positions, going on vacation, or learning new skills. Routing a process through people can be far more complicated than routing a process through a set of servers. Not only do servers never go on vacation, when a job is submitted to a server, it either does it, or it does not. In the latter case, sending a reminder to a server makes no difference. Things like reminders, escalations, and skill matching are additional capabilities that HPM provides.

4. Production Case Management (PCM) - This is designed to handle situations where there is so much variation between individual cases that it is not possible to set out a single fixed process, and yet there is still a well-known set of actions that can be taken. The knowledge worker is actively involved in deciding the course of events for a case, but the range of actions and options is bounded and can be specified in advance.

5. Adaptive Case Management (ACM) - These cases vary so much that knowledge workers are constantly striving for innovative approaches to meet the needs of new cases. The knowledge worker is involved not only in the case, and picking a predefined action, but is actually helping invent the actions that can be taken. However, there is still enough predictability that a given knowledge worker may want to re-use a process from before, and may want to share and discuss process plans with others.

6. Social Business Software (SBS) - 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 - There is no process 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.



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 once the entire project is completed and the fully-tested software is installed. At the other end of the spectrum is email, or phone calls, which need no preparation at all, and can be used immediately without delay. PCM and ACM lie between these extremes; both require a certain amount of upfront investment, but still leave quite a bit of flexibility to the case manager.

If there is investment up front, then dealing with change means some kind of cycle of improvement. PDSI and HPM have fully defined processes which are regularly updated in what we call the BPM lifecycle. PCM has a less fully-defined process model; the case manager picks from a menu of actions in order to meet the needs of the case. The possible actions are predefined and can change only with the involvement of a developer. ACM takes this a step further: the case manager not only chooses the actions but also creates entirely new actions on the fly as needed. You can think of all of these as having a cycle of improvement that get progressively smaller and faster across the spectrum, and when we get all the way to ACM, the entire improvement cycle is performed by a knowledge worker, making the cycle of improvement fast and effective.

Compromise and Transience

The different domains of predictability represent different levels of tradeoff between well-defined representations of the work to be done and the cost of designing and implementing this work. On the left you have very high-value or high-volume situations which can support the high cost of the traditional approaches. As you move to the right, the value or volume decreases and therefore less expensive methods must be used to remain profitable. This is often viewed as a compromise: a poorer solution is being offered because we cannot afford to implement a better solution. The equation rests on the cost of implementation. As the cost of implementing a particular technology decreases, this will tip the balance, and may change the size of the domains.

The enlightenment bias will tempt many technologists to view the domains to the right as simply temporary compromise positions. They will readily acknowledge that while it may not be cost-effective today to make a better implementation, technology is getting more and more powerful and eventually everything will belong in a single category. Remember that the enlightenment bias promotes the idea that all work patterns depend ultimately on a number of rules that could be made explicit. This leads to the belief that while it may be cost-prohibitive today to uncover those rules or to implement an application that addresses those patterns, we can expect that as the power of programming languages increases, the cost of implementation decreases, and we might eventually get to the point that such an approach will handle all business behavior.

The blind spot is in thinking that the world is static. In reality the world of the knowledge worker is always changing. New research results invalidate old techniques every day. A doctor learns a dozen new facts every day that are relevant to their practice – and it is not the same dozen facts across all doctors. Knowledge is always in flux, and knowledge workers are specialists in their particular realm. There can never be a time that complete set of working rules can be made explicit. It will always be the case that as soon as a set of rules are specified, there will be changes necessary.

Researching the condition of a particular patient is hard work, and can require a doctor to incorporate recent news from many different sources. We call this thinking. While that thinking may be difficult, to then externalize the resulting rules and to explain them to a developer is an insurmountable barrier to implementation. The doctor would have to spend half his time explaining changes to the developer, and only half time actually doing medicine. As development techniques get stronger, it will not be the time of development that will be that principal cost, but rather time necessary for the knowledge worker to explain what needs to be developed to the developer.

No matter how powerful the programming techniques get, the domains on the right end of the chart will always be out of reach of them as long as the knowledge worker and the developer are two different people.

It is not just medicine, but in all knowledge worker realms that the flux of knowledge is the barrier to implementing fixed, stable applications. Lawyers and judges have to deal with any manner of unprecedented eventualities and they can't wait for a developer to make the changes. A product manager releasing a product must respond quickly to moves in the marketplace. Understand that it is this flux of knowledge that mandates the technology to be driven by the knowledge worker directly. There must not be a developer separate from the worker, and there must not be any need for developer skills that the knowledge worker would not have. You will also see that this requirement of being self-driven is the divider between the two main types of case management. The idea of information technology being self-driven by the knowledge worker can lead people to the fear that the result will be chaos. As we will see in the next section, stability comes not from rigidity but instead by another property entirely.

The Meaning of Adaptive

To understand the difference between PCM and ACM, you need to understand what we mean by adaptive. Whenever you hear about an adaptive system, you should think about muscles. If you want to increase the size or strength of a muscle, you exercise it. The use of a muscle triggers a response to build the muscle. Conversely, lack of use causes muscle atrophy.

Adaptiveness is not simply that muscles can be adjusted in size. Instead it is more about the ability of the muscle to self-modify to fit the situation; the ability to sense a need, and to respond to it in a kind of feedback loop. Organizations are naturally adaptive if you have experienced people in management and they are getting accurate information about the situation.

Homeostasis is the idea that an adaptive system responds to external changes in such as way as to keep certain aspects constant. Your body maintains a constant temperature by various mechanisms that respond to temperature change. A retail store that can detect the increasing popularity of an item will order larger quantities in order to keep the item available for sale. An organization learns to respond, and since it is adaptive on all levels, each level learns their own part of the solution, in parallel. An adaptive system, if constructed correctly, will automatically adjust to changes and achieve stability in a new configuration in the face of external change.

Examples of Adaptive Systems

Human Body - Not just the human body, but all life forms are adaptive. While DNA specifies how to form all the various muscles, it does not include complete specifics on size. Instead a feedback loop – growth responding to exercise – is used to discover the optimal size. Each muscle is built with the ability to measure the amount of use and respond by growing, or shrinking, appropriately. This simple mechanism eliminates any necessity to predict the size up front.

Adaptive systems optimize themselves. Take, for example, someone who suffers the tragedy of losing the use of their legs. The muscles that are no longer used will reduce in size, while the muscles in the arms will increase to accommodate the increased use.

There are many such systems in the body. The skin responds to light exposure by varying the amount of pigment at precisely the points that received the exposure. This saves one from having to figure out in advance which parts of the skin should be more and less pigmented. Body temperature is maintained at a homeostatic constant through a number of mechanisms including sweat glands and shivering.

The Brain - this complex adaptive system allows a child to learn the language that they hear spoken around them by trying and practicing to improve skill. We study subjects in order to learn them. The concept of learning is again something that refers only to an adaptive system.

The concept of 'practice' pertains exclusively to adaptive mechanisms. Want to learn the piano? Then sit down at the keys and practice, practice, practice. Want to learn to play tennis? Start hitting the ball and practice the right moves. Practice only works because the system is adaptive.

Ecosystems - The diversity of different organisms that thrive in differing conditions form an adaptive network, each organism dominating different aspects of the ecosystem when the conditions permit. The forest as a whole is extremely robust due to the adaptive nature of biodiversity.

Enterprise as an Adaptive System

Human organizations have always been naturally adaptive. The day-to-day decisions are decentralized and delegated to front-line workers. Different divisions compete for scarce resources and good management will shift resources as needed. There may be a centralized view and control at a very high level, but generally this is very much abstracted away from the details of day-to-day operations. Various parts of the organization are sensing and responding to their situations. There is a nested, recursive aspect of this, so that as you get to smaller parts of the organization, the sensing and responding are more finely tuned and detailed.

Organizations are constantly changing and responding to that change. When a person leaves a position, the jobs of dozens of others will change. When an individual is promoted, many people will change their own behavior in response.

Yet the organization is stable. Adaptiveness does not cause constant fluctuation in the organization as a whole. In fact, it is well known that it is difficult to change an organization once it is in place. Adaptiveness presents a kind of homeostasis that allows an organization to keeps its character and form over the years even though people within the organization are constantly coming and going.

An adaptive system is one that effectively senses what is needed, and automatically responds. The knowledge worker is part of that sensing and responding. Without the ability for the knowledge worker to self-modify the system, to adapt to the situation, the ability to sense and respond at a system level is lost.

Adaptive systems have evolved to conquer complexity. System thinking is the approach to try to understand how things influence one another within a whole. System theory attempts to understand self-regulating systems, which is achieved through some form of feedback.

Stability and Fragility

The goal of information systems management has been to make sure that systems are very stable and reliable because we want our organizations to be stable. The opposite of stability is chaos, a steady random thrashing that is hard for workers and management to deal with. One of the biggest fears of IT management is that if the system is not precisely defined ahead of time, then chaos will result.

If you build a machine, and you want it to be very stable, you create it from very sturdy parts. Stability comes from strength. Parts should fit smoothly together, with a precisely specified gap that allows for slipping without any rattling or undue wear. Given that you define things precisely enough and that the materials are sturdy enough, your machine should run for a very long time. The central idea is that a stable system must be composed of even more stable parts.

The traditional view of an organization is as if it was a machine with precisely defined parts doing precisely defined functions. This is the point of scientific management which attempts to break each function into sub-functions that can be rigidly and precisely defined. The idea is that if you can do a thorough job of this, you will have a stable organization because you have very stable parts.

But organizations are not machines, and people are not solid chunks of metal that can hold a precise shape. Organizations are complex adaptive systems and they do not follow the laws of complicated machines. This is once again the enlightenment bias getting in the way of understanding. An ecosystem is not stable because of a precisely defined arrangement of specific parts but because there is a large variety of different organisms that adapt themselves. Biological systems, like a tree, do not have a precisely defined form, but instead adapt to their specific situation while growing. The tree is stable, but no two trees are the same and no detailed design existed before.

Probably the hardest concept to grasp about case management is that stable organizations can result at the high level even though there is no enforcement of a rigidly defined design at the detail level. This bears saying a second time: even though knowledge workers are all working in differing patterns that may appear chaotic, the result can still be a very stable organization.

There is evidence that rigidly defined systems are bad for an organization; they can make the organization fragile. Nassim Nicholas Taleb presents this concept best in his book called “Antifragile.” The term “antifragile” is a property very similar to “adaptive.” Systems that are antifragile actually gain from disorder. In a way, these complex systems actually need a bit of external perturbations to keep them healthy; remove those perturbations and the system becomes fragile. If you protect your muscles from being used, then your muscles become weak, and you can be hurt by a situation that would not affect a healthy person. If you protect a forest from fires, the forest becomes weak through accumulated tinder, and a single fire can have much more devastating effects. Similarly, if an IT system is used to make an organization’s work uniform and done in exactly the same way, then the workers in that organization lose the ability to respond to unanticipated events. IT systems designed around the idea that they represent the real, true (and static) operating rules of the organization tend to enforce those static rules when the organization would naturally be changing and responding to the environment. Taleb would say that the organization becomes fragile and liable to break when external pressures change.

Complexity scientists are still advancing our understanding in this area, but for the purpose of this white paper you need only to understand that organizations are not like machines, and a stable organization does not depend upon the stability and uniformity of what knowledge workers do. The best illustration is this: good customer support is not defined by doing the exact same thing for every customer. Instead, good customer support is defined as doing the best thing appropriate to that particular situation. Case management is a technology designed to allow knowledge workers to do the most appropriate thing given the situation, and it must have the flexibility to allow the knowledge workers to decide, and sometimes to define, what the most appropriate thing is.

Two Kinds of Case Management Technology

In the marketplace today we are seeing two distinct approaches to providing CM technology for people: Adaptive Case Management (ACM) and Production Case Management (PCM). It is not a matter of one being better than the other. These are two different approaches suitable for different kinds of knowledge work.

PCM in A Nutshell

Production Case Management (PCM) is an approach to supporting knowledge workers which is programmed by specially-trained technical people (programmers) to produce a case management application. That application is deployed for use by knowledge workers to get their work done. The application offers collections of operations that the knowledge worker can select to use or not use depending on the specific needs of the case.

A PCM application is used when there is a certain amount of unpredictability in the work, and some flexibility is needed, but necessary actions are regular enough or the volume of work large enough to make identifying and codifying regular patterns valuable. A worker using PCM will be involved in selecting the actions toward the outcome of a particular case, but will not be responsible for the kinds of actions that might be available.

ACM in A Nutshell

Adaptive Case Management (ACM) is an approach to support knowledge workers who need the most flexibility to handle their cases. ACM allows the knowledge workers themselves to create and modify all aspects of a case at any time. There is no distinction between design time and run time: the designing and running are done at the same time by the same people.

This approach is used by knowledge workers who have unique expertise in an area. They don't have a lot of time to transfer this specific knowledge to a programmer, and it is too expensive to hire a programmer for one-off cases. ACM offers a Do-It-Yourself (DIY) approach to process programming. The worker using ACM is responsible not only for the outcome of a case, but also for how the handling of that kind of case improves over time.

It is important to distinguish between these two. If you need ACM and try to substitute PCM, you will be very disappointed, and vice versa. Both are useful in relatively unpredictable situations, but there is one huge distinction that makes all the difference: PCM requires and can benefit from a software developer, while ACM must not. Instead ACM is exercised by the knowledge workers directly as they work. Put another way, PCM must separate the development time from the run time, while ACM must unify these. This may not seem like a big difference, but it affects the very core of what knowledge workers can do and how they achieve results. The difference lies in how predictable the work is.

Comparing ACM and PCM

When we talk about a case management system being adaptive, the complete system includes the case managers as well. Humans are not excluded from the feedback loop. We talk about a good ACM system facilitating what the professional wants (needs) to do. Professionals (case managers) play active roles in adapting the system to their needs. We can think of this as being self-modification because there is no need for a software developer or process analyst: the professional can adapt the system as needed to meet the constantly changing requirements. For example, when a doctor gets the idea for a new treatment plan, they can institute that new plan without involving a software expert.

There is a feedback loop within the non-adaptive PCM system but it is slower, requiring many weeks or months to get a new release out. To add a new option to a PCM system requires a developer, and that developer requires a precise description of the change that is needed. This much slower cycle means that a PCM system cannot adapt as quickly as an ACM system.

ACM is used for what I would call innovative knowledge workers: inventors, creative people, executives, managers, innovators, business entrepreneurs, media producers, doctors, lawyers, etc. These are people who really do need to decide their course of action every day, and the course of action might be to do things that have never been done before. A board of directors does not have a menu of options to pick from when it comes to actions to take. Someone responsible for the merger of two companies will not have a system with all possible actions pre-programmed. A doctor responsible for the survival of a patient may prescribe radical and untested treatment if it seems like the only option.

PCM is for environments where the number of knowledge workers is high, and the courses of action on a given case are sufficiently predictable to justify the cost of developing a dedicated application. Also, those knowledge workers are less responsible for evolution of work to fit new contingencies. For example, while a doctor might be in a position to prescribe a radical treatment, there are many others who work in a health care facility who should not have that flexibility. The routine care of a patient may still be too unpredictable for a fixed process, may still require the judgment of a nurse or clinician, and still the options available may be restricted to a set of known actions.

There are some strong similarities between PCM and ACM:

- At run time the most important concept is that of a "case" which is primarily a folder to collect all the information around the case, accepting essentially any format of document
- Knowledge workers use their own expertise to control the advance of the case from state to state
- The resulting case folder represents a system of record for the work that was done

The biggest distinction between PCM and ACM is that PCM is not adaptive, and this can be seen by three qualities of PCM:

- The programmers uses formalism such as modeling or programming to put the application together
- It uses a standard application development lifecycle: the application is constructed, then tested, then deployed to non-programmers
- After deployment there is limited ability for the case manager to alter the structure of the application itself

ACM Spotter's Guide

More products today claim to have ACM capabilities. Do they have what it takes? Or are they simply jumping on a bandwagon? It is a buyer-beware world. Here are some criteria to help distinguish the two, and avoid dishonest representations. The requirements are broken into three levels: All ACM Systems MUST have level 1. Many will have features at level 2, and only some advanced ones will have features at level 3.

I cannot stress enough that the capabilities described below must be made available in a way that the actual case manager can use them. The case manager is a professional, like a doctor or a lawyer, and not a programmer. When vendor makes a claim that a system has a capability, you need to dig below the surface, and make sure that it is available to the end user without programming. The capability must be both safe and easy enough to use without specific training. ACM supports knowledge workers, and having to go to a programmer will be too big a barrier. Having to request help from an administrator, for example to define the members of the case team, is a similar barrier. These capabilities need to be immediately available to the actual case manager.

Level 1 - Required of All ACM Systems

Team: It must organize the work of the knowledge worker (called a case manager) and others involved in the case (called the case team). The case team is not predefined, and one of the jobs of the case manager is to identify people who are appropriate to help on the case.

Folder: The central concept is a case folder that is used to collect together all of the information and artifacts that might be needed by the case manager. There is no predefined collection of information; part of the job of the case team is to locate and collect the necessary information for the case, and put it in the folder.

Goals: To drive the work to completion, the case team communicates about goals (also called tasks) and intended and actual timelines for performing activities toward those goals. None of the goals are fixed; any goal can be modified, added, or removed by the case manager without special skills.

History: The system keeps time-stamped records of every change that happens in and around the case.

Security: The case manager is in complete control of who is authorized to access the case folder. Unauthorized users cannot access the case folder.

Communication: The system must facilitate communication between case members through multiple channels: telephone, email, instant message, desktop conference, and other technologies.

Adaptive: The case manager can make new cases that build on structures of previous cases through some form of copy or reuse of the earlier cases without needing any special skills. Over time the case manager will adapt the system to their own style of working without needing the help of any specialist.

Reporting: A large variety of user configurable reports exists for communicating status, things that have happened within a given case, and aggregate information across cases.

Level 2 - Common Extensions in ACM Systems

Business Entities: Structured data with metadata behind it. Form capability to display and allow entry of business data.

Data Interchange: Ability to access systems and internalize this within the business entity format for reuse, and the ability to transform the internalized data to a form to send to an external system.

Business Rules: Generalized rules mechanism that tests the business data for specific conditions that can either trigger activities or affect the state of resources.

Resource State Model: Files and other resources can have a state model associated with them. Transitions between states can affect and be affected by other resource state changes.

Granular Access Control: Fine ability to control who can access what parts of the case information through the use of roles into which people can be assigned.

Sensors and Triggers: The ability to monitor external data / state, and the ability to trigger reactions that change the state of internal resources or goals.

Conformance Guiding: The ability to match the activities of a current state with a reference process and provide a visual difference highlighting extra or missing activities, order disparity.

Level 3 - Advanced Extensions for ACM Systems

Process Mining: The ability to analyze the history records and to generate process maps showing the patterns of behavior across many cases.

Social Mining: The ability to scan networks of connections between people in order to determine skills and/or preferences of people to guide assignment of activities.

Federated Case Folders: The ability to link folders in one case system to folders in another system, and synchronize content in both directions

Ontology / Taxonomy: Represented in standardized form and able to exchange with knowledge management systems.

Resource Sharing: A given goal or resource could be shared between multiple cases.

What An ACM System does NOT have

The hardest thing about recognizing a good ACM system is that it is not about what it has but what it does not have. To use an automobile analogy: how do you define a roadster? It is a simple, two-seat car with plenty of power for a visceral driving experience. A roadster is more about what it does not have than what it has. It does not seat more than 2 people. It does not have a huge trunk. It does not have a large towing capacity. It does not have a lot of creature-comfort features. It does not have 14 cup-holders. The dashboard is minimal. If it had a dashboard like the cockpit of an airplane, it could never be a roadster. With manual transmission, the roadster is enjoyed for the direct feel of the road, the enjoyment of driving. So you can not define a roadster by listing all the things it has with the view that more things is better.

Adaptive Case Management is designed to be used by the knowledge worker directly; it will not have many of the more sophisticated features that require a programmer. As you consider usability, think of a doctor using it; or a lawyer using it; or a detective. Here are some things that it must not have, or at the least it must be able to be completely functional without:

Programming Language: The users are not programmers. There needs to be customization, but it must not require programming skills to accomplish this.

Graphical Modeling: Experience has shown that professionals (like doctors) don't have the time or desire to learn such graphical modeling. The ACMS must function without any graphical modeling.

Web Services Integration: This is a capability that is used by programmers, but the users of an ACMS are not programmers. Thus, the ACMS has to work without any API style web service integration.

Of course, a system might have these capabilities, so that a programmer might occasionally help customize for a particular reason. The key here is that these capabilities are not central to the capabilities of the system. Any system might have APIs for extending it, and that is OK as long as those APIs are not central to the features that are needed in the first place. If, however, you see a vendor claiming that these are important features of the system, you should be very suspicious that it is not really an ACMS.

Description of PCM

PCM is used when the number of knowledge workers doing the same job is large and the domain well known, but the process is not entirely predictable. Because the PCM application is developed by programmers, it can make use of more traditional mechanisms for data integration: structured information can be read from some sources, transformed, and written to other destinations. The sources and destinations can be web services or applications with an API. Like a typical development model, once the application is coded, the design rationale behind a particular transformation is not included in the final produced code because it is not needed.

Because a developer creates applications with PCM, it tends to look a lot more like traditional programming and human process management (BPM) than ACM does. A PCM system will often have a graphical modeling language. The Object Management Group (OMG) is developing a standard modeling language called Case Management Modeling Notation (CMMN). The casual observer will see that it looks quite similar to Business Process Modeling Notation (BPMN), but in fact the two languages work on different principles. While BPMN strives to be a procedural language, the goal for CMMN is often described as trying to be more declarative, focusing on business entities and artifacts more than the process as a whole. At the time of writing, the CMMN specification has not been released to the public, but it is known to be based on earlier work done at CORDYS and IBM, and might be assumed to be similar to those earlier works.

There are a lot of services businesses which can make use of PCM. You might use PCM for telecom repairmen. These people need to visit the site, determine what the problem is, and then prescribe a resolution from a menu of well-known operations. It is hard to represent what these people do as a traditional HPM process because it is not predictable enough for a pre-defined fixed process. The process unfolds at run time because the first resolution might not work, and that tells the repairman more about the situation, possibly leading to further action. Yet the repairman is not in a

position to invent entirely new procedures. The phone/TV system is big and complex, and therefore the repairman's options are necessarily restricted to operations that are well-known not to cause a problem with the operation of the network.

Another example is auto service: the car is brought in. There is a set of things to examine. There are decisions about what to repair or replace. Maybe parts have to be ordered. Maybe components need to be sent to a shop for specialized repair. Perhaps the car needs to stay the night. With luck it all ends with the car being driven home.

PCM is most useful when the number of nearly identical offices is large, and the knowledge worker is a professional but not necessarily an owner of the process, and the process itself is not predictable enough to specify every step in advance.

Can One System Be Both ACM and PCM?

The smart money is betting that in the next year, many systems will be presented as supporting both ACM and PCM. Is this possible? Can a vehicle be both a car and an airplane? Yes it can, but it is neither a good car, nor a good airplane. Can a vehicle be both a car and a boat? Yes it can, but it will be neither a good car nor a good boat. Because cars, boats, and airplanes have distinct needs and requirements, there is no chance to meet perfectly multiple needs at the same time.

To be adaptive, the system must be programmed by the knowledge worker, which necessarily means that traditional programming skill must not be required. Involving a developer would be a barrier to getting the job done.

The Do-It-Yourself (DIY) aspect of ACM puts some significant constraints on capabilities. Consider other DIY products you might see in a store. They must be designed to be put together with no particular skill. An amateur DIY kit will always be more limited than the parts that a professional will use. The professional products that require a professional for installation can be higher quality for the same price, because they don't need to be designed to be fool-proof.

There are amateur (DIY) kits, and there are professional kits, but one kit is never both amateur and professional. A real artist will never use a paint-by-number kit, and a paint-by-number kit will never produce real art.

I have argued that using a two-dimensional graphical process diagram of any sort (i.e. BPMN) is a tool that professionals can use. However, such a language does not provide the kind of fool-proof utility that a DIY process system needs. It is easy to drop a few shapes on the canvas and have something that is syntactically incorrect. The syntax rules are hidden and must be learned independently. The knowledge worker is too busy with their profession to learn the intricate details of the syntax of BPMN.

For an adaptive system, the right format for a process for a knowledge worker is a simple list of goals. Anyone can add a goal to the list, and there is no possibility to invalidate the syntax. There are no hidden rules to violate.

It seems overly glorified to talk about a simple list of goals as a process. However, this simple approach allows a case manager to complete cases effectively and repeatedly in the face of changing demands of the situation.

A programmer designing a PCM process for an organization will tend to want to use many sophisticated capabilities to do things, such as determining when a goal is complete. Or in hiding particular potential goals until prerequisite goals have been completed. All of these are effective in making the PCM application effective in the organization and, because PCM is used in high-volume situations, any small increase in effectiveness is multiplied by the number of cases.

Can't you have both? Have the programmer make the basic process, and then allow the case manager to modify it? This is not possible because in the act of designing a program, the programmer makes many assumptions which are not apparent in the resulting program. It is dangerous to make even a small change in such a program without careful study of all the assumptions behind the program.

While a BPMN diagram is powerful for a process expert to use, it does not make clear the underlying assumptions that went into that program, nor does it provide a way to safeguard those important assumptions while allowing other things to be changed in controllable ways. Consider process variables; they are not visible in the diagram, nor is it clear exactly all the ways that a process variable is used. Changing the way a variable is manipulated in one part of the diagram might have dangerous consequences somewhere else. Only after careful understanding of the complete diagram can one make a safe change to it, and such effort is not something that a case manager can put into it.

If a system offers a powerful language for a process expert to make a PCM application, you will eliminate the adaptive capability and it cannot simultaneously be an ACM system. The two kinds of technology are distinct.

Organizational Directions in the Future

Two approaches to knowledge work imply that there are two kinds of knowledge workers distinguished by "responsibility."

Knowledge worker for hire – someone is trained in a specific field and learns to be an expert but has little or no ownership of the overall process. A car mechanic must make accurate suggestions on how to repair the car but does not take responsibility for the repair shop business and must work within the constraints set by others.

Knowledge worker with responsibility – someone who can plan and be responsible for the course of events. This is defined by Peter Drucker as someone "knowing more about their job than anyone else in the organization." These are the workers who handle the wicked problems and have to think outside of the box, e.g. a board member or someone responsible for mergers and acquisitions.

This difference seems to be a determinant for whether you use PCM or ACM.

As a result of these technological developments, new forms of organizations are appearing and winning against older companies that cannot move as quickly. The traditional way to structure an organization was to separate the brains from the brawn: centralized designers and planners would determine the right way to get a job done and the workers would perform the work as designed. The Industrial Age brought factories that were expensive to build, but could mass produce inexpensively and in quantities that made up for the expense of building the factory. In the business office, policies and procedures are put together for the same reason a factory is built: communication is expensive and the coordination cost of a large organization increases exponentially unless standard, centrally planned ways are drawn up. Changes to the plan are expensive because communication is expensive.

This same rationale does not stand up when communications gets very powerful and very cheap which is exactly what social technology is doing. You might be able to move in ways never dreamed possible a few years ago. Imagine a potential customer who needs a small alteration in a product being able to contact exactly the right people almost instantly and propose and confirm a change.

Pull platforms are emerging as a response to growing uncertainty. They seek to expand the opportunity for creativity by local participants dealing with immediate needs.
-John Hagel, John Seeley Brown, Lang Davison

In their book "The Power of Pull" John Hagel et al. call this a pull-oriented organization. A push organization is one where market research is done centrally, products and planned and produced, and then pushed out through the distribution channel where they sell. A pull organization instead allows the front line to work directly with the customer, determine exactly what they want, and then leverage the rest of the organization in an agile manner. Some companies today propose different product designs to customers and get them to vote. The winning design is then produced and sold to a waiting audience. It is almost as if you can sell the product before it is designed. In all of these experimental approaches, agility is achieved by distributing control away from the center, and leveraging powerful new communications media to make people in the field more flexible than ever possible before.

These shifts in the workplace make case management more important than ever. It gives flexibility to the users to evolve their own workplace. ACM is not deployed in the form of applications or solutions in the way that traditional IT approaches for pre-planned work support is. Instead, it is deployed in a plain vanilla form, for immediate use by knowledge workers. Over time, case managers build up patterns within the cases they work on. They reuse those patterns as they deem worthwhile. Over time, the organization gets more efficient, not because it has a centrally planned and enforced work process, but because the different parts of the system can adapt as needed.

Today, nearly 40% of the workforce can be called knowledge workers and that number is growing. Never before has there been a systematic approach to making managers, executives, and other thinkers more effective at getting things done. The current knowledge workers will need to adopt this or risk being replaced by the new people entering the workforce. As Drucker says, knowledge worker productivity is the biggest of the 21st century management challenges, and it is likely that case management will be a part of addressing that challenge in the next decade.

Author

Keith Swenson is Vice President of Research and Development at Fujitsu North America Inc. He is known for having been a pioneer in collaboration software and web services, and has helped the development of many workflow and BPM standards. He is currently the Chairman of the Workflow Management Coalition. In the past, he led development of collaboration software MS2, Netscape, Ashton Tate and Fujitsu. In 2004 he was awarded the Marvin L. Manheim Award for outstanding contributions in the field of workflow. His recent book "Mastering the Unpredictable" has introduced and defined the field of adaptive case management.

He blogs at http://social-biz.org/

About Fujitsu America

Fujitsu America, Inc., is a leading ICT solutions provider for organizations in the U.S., Canada and the Caribbean. Fujitsu enables clients to meet their business objectives through integrated offerings and solutions, including consulting, systems integration, managed services, outsourcing and cloud services for infrastructure, platforms and applications; data center and field services; and server, storage, software and mobile/tablet technologies.

For more information, please visit: http://solutions.us.fujitsu.com/ and http://twitter.com/fujitsuamerica

About Fujitsu

Fujitsu is the leading Japanese information and communication technology (ICT) company offering a full range of technology products, solutions and services. Approximately 170,000 Fujitsu people support customers in more than 100 countries. We use our experience and the power of ICT to shape the future of society with our customers. Fujitsu Limited (TSE:6702) reported consolidated revenues of 4.4 trillion yen (US$47 billion) for the fiscal year ended March 31, 2013. For more information, please see http://www.fujitsu.com.

FUJITSU AMERICA, INC.
Address: 1250 East Arques Avenue Sunnyvale, CA 94085-3470, U.S.A.
Telephone: 800 831 3183 or 408 746 6000
Website: http://solutions.us.fujitsu.com
Contact Form: http://solutions.us.fujitsu.com/contact

Have a question? Email us at: AskFujitsu@us.fujitsu.com

FUJITSU LIMITED
Shiodome City Center, 1-5-2 Higashi-Shimbashi
Minato-ku, Tokyo 105-7123, JAPAN
Telephone: +81-3-6252-2220
www.fujitsu.com

FPC65-6749-01 05/13.
13.0515