Business-IT Strategies

Vol. 11, No. 11

CUTTER CONSORTIUM

After-Action Reviews in IT

by Vince Kellen, Senior Consultant, Cutter Consortium

Anyone who spends time supporting complex IT environments will experience a major outage at some point in his or her career. In many cases, the experience is traumatic, leaving a lasting and painful impression. Companies can avoid this and other potentially major incidents if they implement an after-action review (AAR) process. AAR processes are found in the military, medical communities, the safety engineering discipline, and are normally embedded in total quality management (TQM) frameworks. Despite the rather widespread adoption elsewhere, most IT shops do not make rigorous use of AARs. This needs to change. This *Executive Report* explains AARs in detail by describing the steps within an AAR process and discussing the cultural and organizational tricks and traps that IT executives need to consider when implementing an AAR process.

Executive Report

ABOUT CUTTER CONSORTIUM

Access to the Experts

Cutter Consortium is a truly unique IT advisory firm, comprising a group of more than 100 internationally recognized experts who have come together to offer content, consulting, and training to our clients. These experts are committed to delivering top-level, critical, and objective advice. They have done, and are doing, groundbreaking work in organizations worldwide, helping companies deal with issues in the core areas of software development and agile project management, enterprise architecture, business technology trends and strategies, innovation, enterprise risk management, metrics, and sourcing.

Cutter offers a different value proposition than other IT research firms: We give you *Access to the Experts*. You get practitioners' points of view, derived from hands-on experience with the same critical issues you are facing, not the perspective of a desk-bound analyst who can only make predictions and observations on what's happening in the marketplace. With Cutter Consortium, you get the best practices and lessons learned from the world's leading experts — experts who are implementing these techniques at companies like yours right now.

Cutter's clients are able to tap into its expertise in a variety of formats, including print and online advisory services and journals, mentoring, workshops, training, and consulting. And by customizing our information products, training, and consulting services, you get the solutions you need while staying within your budget.

Cutter Consortium's philosophy is that there is no single right solution for all enterprises, or all departments within one enterprise, or even all projects within a department. Cutter believes that the complexity of the business technology issues confronting corporations today demands multiple detailed perspectives from which a company can view its opportunities and risks in order to make the right strategic and tactical decisions. The simplistic pronouncements other analyst firms make do not take into account the unique situation of each organization. This is another reason to present the several sides to each issue: to enable clients to determine the course of action that best fits their unique situation.

Expert Consultants

Cutter Consortium products and services are provided by the top thinkers in IT today — a distinguished group of internationally recognized experts committed to providing top-level, critical, objective advice. They create all the written deliverables and perform all the consulting. That's why we say Cutter Consortium gives you *Access to the Experts*.

For more information, contact Cutter Consortium at +1 781 648 8700 or sales@cutter.com.

The Business-IT Strategies Advisory Service Executive Report *is published by the Cutter Consortium, 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA. Client Services: Tel: +1 781 641 9876; Fax: +1 781 648 8707; E-mail: service@cutter.com; Web site: www.cutter.com. Group Publisher: Kim Leonard, E-mail: kleonard@cutter.com. Managing Editor: Cindy Swain, E-mail: cswain@cutter.com. ISSN: 1530-3470.*

Rob Austin | Ron Blitstein | Christine Davis | Tom DeMarco | Lynne Ellyn | Jim Highsmith | Tim Lister | Lou Mazzucchelli | Ken Orr | Mark Seiden | Ed Yourdon

Cutter Business Technology Council

After-Action Reviews in IT

THIS MONTH'S AUTHOR

Vince Kellen
Senior Consultant, Cutter Consortium

IN THIS ISSUE

It's 2:12 am. The cell phone is ringing. What now? It's Amy, a data center technician. She wouldn't be calling me unless something is wrong. Very wrong. She tries to explain that the primary storage network is failing. And the path to the backup is failing, too. She can't tell what it is. Everything is coming down. In the middle of the call, Rick, who manages the storage systems, is calling. I put Amy on hold. Rick says it isn't the storage network at all. All the information says it's the database server having issues but from his perspective, it doesn't make sense. He is checking. As soon as he hangs up, Bill calls me. He's the manager of the database group. I tell him to wait a minute and tell Amy I have to hang up on her. Amy is upset and says she needs to get back to the manufacturing group since this is affecting their big processes for tomorrow and these seasonal processes represent tens of millions of dollars in revenue. I go back to Bill. He explains the situation. All his metrics say the system outage, which now appears to be widespread, isn't a database problem. He is agitated, saying everyone keeps blaming the database group. Just because he has the best metrics doesn't mean everyone has to blame him each time there is an issue. He thinks the problem is in the manufacturing application. After all, yesterday the manufacturing group put into production some significant vendor patches. In the middle of this, Lynn calls. She heads up the manufacturing group. She is furious because she says the database group switched some database drivers last night out without telling her, and she thinks the driver changes may be at fault. She says all their testing on the manufacturing changes checked out just fine. She is angry, too.

In the middle of this, I receive two SMS messages from my boss. The first says, "What the heck is going on?" The second says, "Manufacturing is nervous. Please call Pat. And then call me." I can tell he is not happy about this one. Heads are going to roll.

Anyone involved in the support of complex IT environments will experience a major outage like the scenario just described at some point in his or her career. In many cases, this experience is traumatic, leaving a lasting and painful impression on those involved or worse, precipitating significant job changes for some. Scapegoats are hunted down. Vendors are put through an inquisition. Management looks for the weakest link or who they can vote off the island to show everyone that the company is serious and, more important, to move the target off their backs. Lawyers start gearing up. Employees polish resumes.

Major incidents can turn into witch hunts, which would be the worst of all outcomes. Why? Because there is so much to learn from failure. Witch hunts push that knowledge underground or out of the organization. Witch hunts prevent the company from properly diagnosing its own errors and designing its own solutions, both of which are critical for preventing future failure.

AARs are best conceived as a tool useful to line employees and frontline managers, not just senior management.

In order to run IT operations smoothly, the IT group needs to have the cultural and organizational prerequisites before it can effectively learn from failure. Having these prerequisites in place helps the IT group build up knowledge — not only of the systems in question — but more important, of how to interact with each other in stressful moments. While major outages are often the catalyst for change, they aren't the best catalysts for change.

One of those organizational prerequisites, an after-action review (AAR), has been used by the US Army for more than two decades. The Army uses AARs to understand what happened during an engagement so that the individual soldier and others, including their leaders, can learn and corrections and improvements can be determined. After-action reports, which are short written summaries of key engagements or actions, have been in use for two millennia and have served useful not only for commanders of armies to learn what is going on, but for historians of war. The US military establishment picked up on the more thorough AARs in the 1980s, especially as technology facilitated data collection techniques.[1, 2] The healthcare industry requires root cause analysis, a component of an AAR, for significant events. For decades, the safety engineering discipline has used something akin to an after-action review in order to document actual accidents and near misses. A cardinal truth in safety engineering is that in order to learn how to design safety into equipment and processes, one has to understand something about near misses and prior accidents.

The purpose of an AAR process is to help staff collect, analyze, and synthesize the data regarding a major incident, develop a narrative and a causal model of what transpired, and develop recommendations on how to prevent similar failures or improve related practices. The AAR process is a critical component of knowledge management for improving products, processes, skills, and abilities. AARs do not need to be restricted to failures, but are as effective, if not more so, for understanding successes. After-action reviews are a principal means for an organization to assimilate difficult feedback about what hasn't worked.

AARs are best conceived as a tool useful to line employees and frontline managers, not just senior management. It is far better for individuals and teams to diagnose their own errors and implement their own solutions. When senior managers insist on continually doing that for the lower levels of the organization, they infantilize the organization and render it utterly incapable of maximal improvement. While self-directing the course of action from the top may allow the executives to stand in the limelight and gratify their egos, it also creates a culture where only mindless automatons or consummate sycophants prosper. In a top-down, directive culture that does not allow for bottom-up self-diagnosis and self-correction, the organization is at the mercy, for better or for worse, of the skills and capabilities of senior leadership alone. Such an organization will fail to develop or utilize the immense knowledge and passion of its workforce and given the right environmental stress, it will face crisis in the long term — if not extinction. AARs are an important feedback tool to help the top and the bottom of the organization share their thoughts about risky, difficult, and complex work.

AARs IN IT

While after-action reviews are now widely adopted in the military, the safety engineering discipline, and healthcare, use of robust AARs within IT remains quite spotty. Many, if not most, IT shops do not employ regular AARs. Instead, to learn from past failures (and successes), IT workers rely on personal history, oral tradition, case studies, or whatever they can learn in their college degree programs or seminars and conferences. "Adhocracy" rules.

IT shops can glean much from conducting their own AARs. IT systems are complex, and more important, idiosyncratic. No two firms have the same IT system configuration. IT system failures and successes are more often the byproduct of specific configuration nuances in each organization that are rare and hidden than of widely known systemic problems in equipment or software. Collecting and analyzing AAR data would not only help develop a deeper understanding of the firm's specific IT configuration and management issues, it would also help the firm improve its IT operations in a way competitors can't or won't. When IT employees learn how to do AARs, they shift their thinking about IT operations and start to understand the need for a frank, open, and systematic approach to the improvement and prevention of incidents.

IT operations are full of what safety engineers call "near misses." These are incidents that don't have a measurable business impact or haven't resulted in downtime or loss of data. Near misses, however, do represent opportunities to improve equipment, processes, and human abilities. For every major incident, hundreds, if not thousands, of near misses lie underneath. In this regard, major system outages are just the tip of an iceberg. All sorts of process, equipment, or skill gaps exist, hidden below management attention and usually attended to by line employees in an ad hoc fashion without raising concerns for fear of rocking the boat or out of ignorance. AARs can be used to glean insights from near misses as well as major incidents.

Cultural Preconditions

CIOs will be hard pressed to introduce an AAR process in an environment where system failures are fraught with fiery rhetoric, finger-pointing, infighting, blaming, and reprisals. In order to properly introduce an AAR process, the culture of the IT unit must be conducive. Left to evolve by themselves without more learned guidance, IT cultures — just like any business culture — will naturally become defensive and have difficulty dealing with tough situations where careers and reputations may be on the line. CIOs will need to attend to the IT culture. This lesson should not go unheeded. In the process of implementing its AAR processes, even the US Army learned the importance of the cultural preconditions. Without the preconditions, the AAR process will simply become a tool in the larger political game or, even worse, a highly regimented but meaningless drill. For the effective adoption of after-action reviews, I believe the organization has to take a strong stand on how it chooses to manage risks such as outages. Here are some ideas.

IT system failure is a shared risk. It isn't assigned to one specific person, and one person's career does not hinge on a system outage. IT systems are complex and require well-coordinated teams to be managed effectively. The whole team must share the risk. IT senior managers must be careful to ensure they are not perceived as singling out or blaming one person. Even the perception, false or not, that IT senior management plays the blame game will cause IT frontline employees and managers to do the same and engage in defensive CYA behavior. AARs require frankness, openness, and the exposing of personal and team vulnerabilities. IT senior managers need to treat AAR processes with great care.

If the cultural preconditions are not in place, CIOs can use an AAR process as a tool to help establish the right organizational culture. However, the CIO should be leveraging more than just the AAR to develop the right culture. I recommend a multilevel approach in which decision-making processes around hiring, promoting, performance appraisals, incentives, and pay are examined and altered to promote the right culture. The CIO needs to make sure the approaches to budgeting, organizational design and redesign, and overall project management are consistent with the desired cultural preconditions.

In short, the entire system of motivations for IT employees needs to be examined to ensure an AAR process will be effective. Like any other worker, IT workers tend to pay attention to how their job duties are assigned, how they are assessed and advanced in their career, and how IT senior managers handle budgets and project prioritization. Those IT workers hoping for a promotion are especially sensitive to these and other cultural considerations. AARs will be effective to the degree that the overall system managing IT worker motivations is consistent with the chief aim of an AAR: to learn difficult lessons from personally painful situations.

Organizational Preconditions

Essentially, an after-action review is a business process. The process has inputs, steps to perform, and outputs. AARs are more easily adopted by organizations that have a sense of and a respect for business process discipline. IT shops vary both in their need for and their use of business process discipline in their own work. I have seen chaotic IT shops where there is little sense of or appreciation for well-done business processes, and I have seen overly regimented IT shops where everyone served the almighty business process and no one dared do otherwise.

In all methodologies, frameworks, and collections of business processes, I am more interested in how these things illuminate knowledge versus constrain action. All frameworks need to help generate knowledge to improve things. When frameworks become constraining straitjackets, they fail to win over the minds and hearts of the employees and then fail to illuminate. AARs are of the same ilk. The organization needs to strike the right balance between knowledge gained from the endeavor and the effort required to get the knowledge. The right balance has to be set and perceived by line employees as appropriate, not just upper management.

Organizations that have a history of good business processes and quality improvement will easily adopt an AAR process if they haven't done so already. For less mature IT organizations, CIOs can also use an AAR process as a beginning point for introducing process discipline. As noted before, a multilevel approach is in order. Having well-run AAR processes but having little if any process discipline elsewhere in IT does no good. After all, the people who conduct AARs will make recommendations on how to improve many IT processes. For those with quality frameworks in place, an AAR process can be a critical component for failure diagnosis within that framework. When I last implemented an AAR process, I did so first knowing that a full-quality management framework would be following closely. I wanted to use the after-action review as a way of introducing staff to IT business process concepts before launching into full implementation of a quality framework.

AARs provide documentation and analysis of critical incidents and some number of near misses. They do not come close to capturing the total number of issues.

AARs provide documentation and analysis of critical incidents and some number of near misses. They do not come close to capturing the total number of issues, problems, or incidents that IT shops handle. IT shops that have good contact or call centers with incident (or case) tracking will have an advantage in implementing AARs. IT staff can analyze the data in the case management system, which may prove useful in understanding a complex incident. IT shops that use case management systems tend to produce reports listing top problems. IT staff can then use these lists of top categories of cases to understand the nature of some of the problems and to prioritize improvement efforts. Both streams of performance data, ongoing case management and AAR data, are valuable for improving IT operations.

As with all human change, multilinear simultaneity is the key. That is, one often has to do many things at once in order to effect lasting change.

HOW TO CONDUCT AN AAR

One can devise any number of ways to conduct an after-action review. As is typical in IT management, there are multiple correct answers to this problem. One approach I have used has the following basic steps:

1. **Initiate an AAR.**
2. **Appoint an AAR leader.**
3. **Appoint a senior management sponsor.**
4. **Collect data.**
   - ○ Review existing documentation, system logs, and any other relevant system artifact.
   - ○ Consult with prior AARs, if needed.
   - ○ Conduct interviews.
5. **Analyze and synthesize the data.**
   - ○ Appoint specialists needed to analyze and prepare any data.
   - ○ Conduct small or large group meetings.
   - ○ Begin work on key deliverables, including a timeline of events, a causal factor analysis, a common factor analysis, a blue sky scenario, and recommendations and followup actions.
6. **Review and share AAR documentation.**
   - ○ Produce deliverables.
   - ○ Conduct briefings with relevant groups.
   - ○ Make any adjustments in AAR deliverables.

Don't be alarmed with the document-centric flavor of this AAR approach. AAR documentation should be limited to only what is needed and what actually sheds light on the incident. In practice, many types of IT employees can be shown how to construct each of the documents and how to go through the AAR steps. The approach should be simple, lightweight, and the amount of insight generated should exceed the effort required to conduct the AAR. Figure 1 depicts an overall process for conducting an AAR. Not all organizations will or should follow this overall process as it is depicted. It is presented here as one possible example. The process needs to be tailored to individual circumstances such as working with an existing total quality management (TQM) framework or other governance processes and should match well with the cultural and organizational preconditions. Later in the report, I will explain Step 5 in more depth since this is the critical activity.

Initiating an AAR

In improving the quality of IT operations, it is best to leverage all levels of the organization, not just senior or frontline managers. In the process I am sketching out here, I prefer to let anyone within the IT function, no matter whom it is, ask for an after-action review. While more often than not, senior IT managers do call for AARs, I have frequently seen frontline employees see the need for an AAR, especially on a near miss. The requestor of the AAR can contact the CIO or an appropriate senior IT manager. If the senior IT leader concurs, then the AAR is initiated. I think it is wise to let concerns about system reliability or project success occasionally blow over to the top layers of IT management so that senior IT management stays in touch with line concerns. These senior IT managers have to handle the request, which may bypass layers of management, with care, being mindful of the cultural preconditions that AARs require.

IT senior managers and CIOs should set goals for a certain number of AARs to be completed each year or each quarter. If there are no major incidents in a timeframe, senior management can focus the AAR process on near misses or smaller problems. By setting a target for a specific number of AARs, the management team can then align motivations and incentives, thus encouraging IT staff to embark on the AAR process. Setting a target shows that management wishes to have more, not fewer, AARs conducted. When incentives in the organization are aligned to reach this goal, IT employees begin to associate conducting AARs with accomplishing important goals. The natural negativity associated with AARs can be tempered with the positive aspects of accomplishing goals.

Since major incidents are the tip of the iceberg, the way to reduce the incident rate for major incidents is to begin to reduce the incident rate for near misses. Conducting a certain number of AARs in a time period helps ensure that you can review a sufficient number of near misses. AARs ought to have senior IT management sponsorship. I prefer AARs to have a senior IT manager be formally appointed a sponsor. This also signals to the

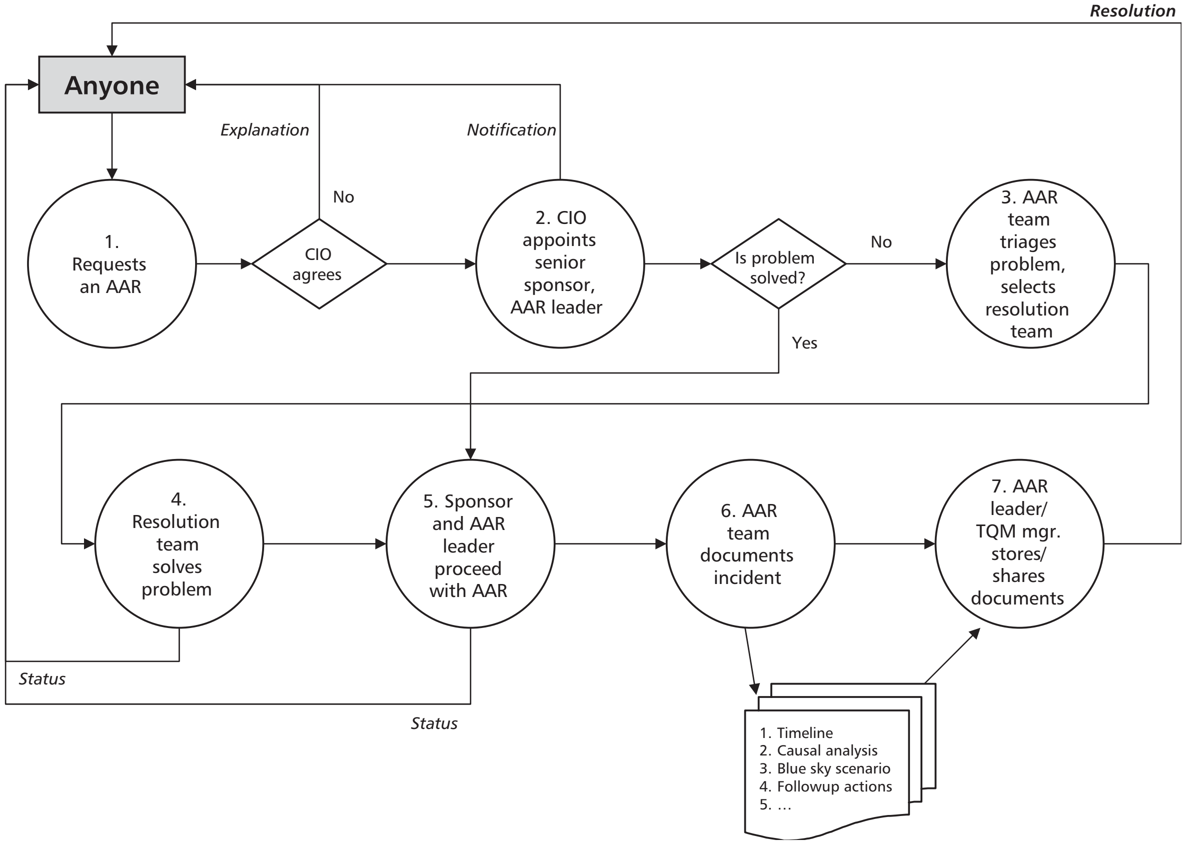
Resolution
Anyone
Explanation
Notification
No
1. Requests an AAR
CIO agrees
2. CIO appoints senior sponsor, AAR leader
Is problem solved?
No
Yes
3. AAR team triages problem, selects resolution team
4. Resolution team solves problem
5. Sponsor and AAR leader proceed with AAR
6. AAR team documents incident
7. AAR leader/ TQM mgr. stores/ shares documents
Status
Status
1. Timeline
2. Causal analysis
3. Blue sky scenario
4. Followup actions
5. …

Figure 1 — Overall after-action review business process.

IT organization that IT senior management will remain involved in the details of the AAR and the AAR process.

Once the sponsor is appointed, the leader of the AAR process should be appointed as well. This is far easier in larger IT staffs that can dedicate personnel to quality-improvement tasks. In the past, I found that we get the best results with someone who is trained in safety, quality, or system-reliability engineering and is fully dedicated to conducting after-action reviews and other related quality-improvement tasks. However, some IT shops do not have the size or available personnel to conduct AARs. In these environments, the person who leads an AAR must split his or her time between work duties and conducting the AAR; often, multiple employees have to learn how to conduct AARs so the burden doesn't fall entirely on one person. However the IT shop decides to do it, the leader will need to ensure that the AAR is conducted as the organization has defined it.

If needed, the AAR leader or the sponsor may wish to identify an outside facilitator to help.

The AAR leader will need to have access to the people and artifacts needed to complete the AAR and should have the visible backing of the AAR sponsor and IT senior management. The AAR leader will most likely be assigned long term to the AAR as followup actions are identified and monitored. If needed, the AAR leader or the sponsor may wish to identify an outside facilitator to help. The AAR leader should be someone empowered to accomplish such tasks.

After-action reviews should be initiated quickly — within a few days of the incident or its resolution. When organizations wait too long to start an AAR, memories fade and the quality of the data is reduced. The initial sense of urgency may have passed and people may actually settle more deeply into a superficial understanding of the incident that will be harder to address.

Collecting Data

Since incidents can vary significantly from each other, so can the data you will need to collect in an AAR. In addition, the way the organization collects data when first implementing an AAR process will be quite different from the way it collects data after doing it for a few years. Early on, data collection takes a bit more effort. As noted above, the best results occur when the IT shop has personnel dedicated to managing the data collection process. The types of data to be collected include:

- System logs, which contain date and time-stamped records of system events
- System performance data, such as network loads, CPU utilization, system I/O performance, and latency and error rates
- Change management and production control logs, containing information on which production environments have changed
- Lists of exceptions to documented change management procedures
- Any notes that employees keep regarding operational processes, such as exceptions, errors, unusual conditions, and ad hoc adjustments
- Existing architectural documents and diagrams depicting relationships between IT systems and subsystems
- User-activity listings or time-entry logs, which help the AAR leader and AAR team understand who was doing what before, during, and after an incident
- TQM documentation, such as business processes and procedures, which depict how work activities are performed
- Any prior AARs, which can shed light on the current incident
- Incident tracking data, such as help desk ticket information
- Prior analysis of incident data, including trending and frequency analyses
- Interviews with key IT employees or users to help those conducting the AAR understand the nature and effect of the incident

The data collection process is not a thing unto itself but is subservient to the need to produce the AAR documents. The AAR investigators should not be collecting data they don't need or engaging in a fishing expedition. The data needed to produce the documents is the data that the team should be collecting. Data collection, data analysis, and data synthesis will be iterative. As the AAR process unfolds, the team's theory about what transpired may change, thus causing the team to embark on more than one data collection session.

The first time the team goes through an AAR process, the team will find the data collection more difficult.

The investigators will be less familiar with the overall process and the documents it produces, thus causing more iterations of data collection. IT shops that have been through several after-action reviews will actually prepare their own logs and notes in anticipation of an AAR while an incident is unfolding and often in advance of an AAR being requested, knowing that the AAR process will be much quicker if they have their notes in order and handy.

For IT shops with a TQM framework in place, such as ITIL, ISO 9001:2000, COBIT, or Six Sigma, data collection will be easier because these frameworks require good documentation of processes and procedures and the establishment of processes and procedures to promote quality improvement. Moreover, each of these frameworks has a portion of their methodology dedicated to problem identification, resolution, and followups embedded within them. In effect, these TQM frameworks have something very close to an AAR already built in. What is the difference? An AAR is a more intense and team-oriented exercise designed to tackle more complex problems and can be easily added into the TQM framework.

For IT issues, most of the data collection can occur through e-mail or one-on-one discussions. However, sometimes group meetings may be required to go through complicated data or for the AAR investigators to learn details about systems or processes that will help them in preparing the documentation.

Analyze and Synthesize the Data

Theories about what happened arise almost instantly after a system failure occurs. The theory-building process (actually, it is one of creating a theory, testing it, and discarding it) is critical for responding to major system failures. The AAR process may be useful for helping teams solve problems *in medias res* (i.e., in the middle of things). However, due to the often frantic pace that ensues in a major outage, AARs are best reserved until after the situation has been sufficiently resolved. However, the theory-building process that the teams engage in during a crisis and in reviewing a crisis after it has passed is the most valuable part of an AAR. This is where the team learns new things.

The AAR process, when done after an incident, generates yet another cycle of theory-building as to what transpired, which is necessary for the team to find improvements it may have missed earlier in the middle of things. Frequently, IT shops address symptoms and not causes in a major incident, choosing to address the cause at a later time. Also, teams that have undergone several AARs tend to do better at resolving system failures while they are occurring. Why? Because the team is often better at solving complex, multifactor problems after they have gone through several AARs. The team starts to build multiple theories of causation and then systematically looks for data to prove or disprove theories until a satisfactory course of action is established. The team gets good at doing this in a frank and open manner. The only difference between the theory-building process in the middle of a crisis and after the fact is that the after-action theory building can be more communal, can occur in a less stressful context, and can take as much time as needed. In an ideal world, the theory-building process in the middle of and after a major incident should be remarkably close. In reality, even just being partially close helps.

The theory-building process will vary widely depending on who and what was involved in the incident. Let's take a complex problem involving multiple levels of computing abstraction (network routers, operating systems, storage systems, and applications) and several IT functions (security, data center management, network support) as an example. In this kind of problem, all sorts of teams may be involved, including those who manage the system hardware (storage systems, servers, switches, routers, cables); those who manage operating systems and middleware integration tools (application servers, Web servers, Web services, and messaging software); those who manage databases; those who manage vendor applications; those who manage custom applications; and those who manage edge devices, including PCs, PDAs, and other workstations. Multiple hardware and software platforms may be involved, which will require the team to understand the interaction effects across the different hardware and software platforms.

Since the total body of knowledge to properly understand a major system failure can be quite large, the right people will need to participate in the theory-building process, including enterprise architects who possess a higher-level but more unified perspective and frontline IT support staff that has deeper and narrower implementation-specific knowledge about a specific hardware or software platform.

Each person will tend to have their own ideas about what happened, drawing from data they are most familiar with, which will be more limited than the sum total of information available. The purpose of the data analysis and synthesis process is for people to share their data and their insights and develop a theory about what happened with which the entire team can agree.

To get this shared understanding, I have found it is best to first assemble a chronology of events without regard to developing a theory of causation. People often work in the reverse. They build a theory of causation immediately based on their understanding of the sequence of events. Causality is deeply tied to temporality. Often, our only way of building a theory of what happened is through the knowledge of the order in which things happened. Teams and individuals typically infer causality (incorrectly) from the temporal sequence of events. The AAR team needs to reverse this process and nail down the sequence of events without too much regard for causality. The blame game can be better neutralized when all parties understand the entire timeline. As simple as a chronology is, I have seen too much resentment and anger caused by a simple misunderstanding of the sequence of events because a chronology was not developed and communicated.

Timeline

The entire team should be aware that the first data collection task is to jointly assemble an unambiguous chronology or timeline. A timeline can be simple with the following four attributes:

1. The date and time.

2. What system is involved in this step?

3. Who was involved with this step?

4. A description of the item or step.

Portions of a timeline are usually known and documented by the area responsible for a specific system. The AAR leader will aggregate the portions of the timeline into a master timeline and confirm with all parties the accuracy of the timeline. IT employees first begin to get a sense of the overall timeline during this review cycle and then begin, as we would expect, to modify their theories of causality to fit the data. Remember, it is very difficult to stop IT workers from building theories of causality. Rather than exhorting to staff to avoid prematurely building causal models, I do the opposite. I encourage the development of multiple, competing theories of causality immediately even within the data collection process. Having multiple competing theories spurs creative thinking and deepens data collection efforts.

Even at this stage in the process, and especially if we are in the middle of an incident, the team will construct a theory board, which is a simple list of major theories the team is considering that roughly explain the incident causes, which then guides data collection efforts. Some theories are promising; some are not. Some start out as promising and then become discarded. Others that were considered not relevant can suddenly become the dominant theory. In complex outages, this “theory jitter” is normal and should be encouraged, especially as new data comes in. The theory board is not a detailed causal analysis (see the following section). It is merely a convenient tool to allow IT staff to develop multiple models of what happened.

In IT work, we have plenty of system metrics, alerts, and logs that can keep those who like to collect data happy for quite some time. In diagnosing major system incidents, a plethora of data can be a curse. It tends to give rise to all sorts of false positives as teams examine data more closely and latch on to what they believe to be deviations in the data that are abnormal but are actually spurious, having no effect on the main outcome. Like a dog chasing its tail, this results in teams chasing their own data. Even worse, IT groups that assiduously collect and monitor data are frequently the best-run groups and in a major incident are often examined the most due to the abundance of monitoring data. The light will then shine brightest on the area that has a lesser probability of being a factor.

Once in a while, I have seen teams come to the opposite conclusion: start examining where they have less or no data. The takeaway for the data collection phase is clear. After-action review investigators should remember to examine systems and processes that produce little data.

Causal Factor Analysis

Also known as root cause analysis or problem analysis, causal factor analysis is designed to identify the effects (the problem, incident, or outcome) and the causes that led to the effect. Effects are usually clear, such as “65% of the user community could not access any network resources for two hours.” A major incident can have more than one effect, and it is often useful to depict these separately. Doing so helps the team probe deeper into details. An effect typically has multiple causes. Causes have a hierarchical relationship where multiple component causes contribute to a main cause. An incident can be depicted as a hierarchy of one or more effects, with each effect having multiple causes contributing to it.

Complex IT environments have incidents for which it may be exceedingly difficult to determine causality. Usually, multiple systems can fail in a cascading sequence of events for which it may be impossible to prove that one component of the architecture actually affected another. This is especially true for distributed software systems like complex Web services in which the combinations of possible interactions are much too large to examine, even partially. This is where human expertise proves valuable, especially group expertise.

To bring this expertise to bear, the AAR leader usually calls for one or more group meetings of the best informed minds to begin to describe the causal factors and their relationships with the effect(s). During these meetings, the team will frequently iterate between the data they have, the current model or theory of causality constructed, and the data they don't have. These meetings are the heart and soul of an AAR because it is during these meetings that the team generates new and valuable insights.

Describing cause and effect is deceptively simple. Let's use the introduction as an example. The confusion in the introduction is that the main effect is never stated; it is only implied. At 2:12, some system appears to be down and manufacturing processes appear to be in jeopardy. Many IT workers start with the system as an object of an effect, such as "the main database system was unavailable" or "the application server load balancer failed." One could argue that these are not effects at all but causes themselves. One can also argue perpetually about what is a cause and what is an effect.

Rather than get lost in such badly overused and understood terms as cause and effect, I tend to use the term "main outcome" to describe, from a business perspective, what happened to the business. In the discipline of root cause analysis, other terms such as problem, event, or incident are also used. Since I like to use the AAR process to document positive outcomes as well as negative ones, I prefer the emotionally neutral term "outcome" to describe the main effect.

In the example at the beginning of this report, the main outcome can be worded as "the manufacturing unit was unable to process $21 million in invoices, resulting in a significant business loss." I tend to use the term "factor" to refer to causes. A factor is any other item that we believe contributed to the main outcome. In the case of this example, a faulty database driver and an unauthorized change in the production environment are two factors, among others, that may contribute to the main outcome.

Factors can be apparent or substantive. Apparent factors are ones that can look like real causes but actually are not. For example, in analyzing our incident above regarding database drivers, Lynn believes the incorrectly applied database drivers are a factor. Further analysis may show that the database team followed all reasonable procedures in applying the updated drivers, but the updated drivers contained a defect within them that only surfaced due to unknown conditions present in the company's specific production environment. The vendor's testing and broader client-incident history revealed no problem with the drivers, and the database group's testing was conducted in a test environment that can't fully simulate the production environment. Lynn's factor, incorrect driver update by the database team, is apparent. The real factor, or the root cause, lies in how did the defect get built into the new database drivers or how can the database team uncover defects in the database driver that 3,400 other companies did not? In this case, the root cause may be beyond the control of this IT shop.

Sometimes the team gets lost in terminology, phenomenology, and ontology (i.e., in differences in how people use words and how they conceptualize subjective and objective experiences).

Substantive factors, or root causes, are manageable by the group. They are identifiable. Since the defect in the database driver isn't directly manageable by our doomed IT shop, the team will need to identify other factors within their control. For example, is there something in this IT shop's production environment that isn't quite right, which is contributing to the database driver problem? Is the nuance in the production environment likely to generate additional problems? Further research and discussion may reveal the answers and the root cause.

Sometimes the team gets lost in terminology, phenomenology, and ontology (i.e., in differences in how people use words and how they conceptualize subjective and objective experiences). While the philosopher or scientist in me would prefer a clear and precise definition of all things, the pragmatist in me realizes that the consensus of the group is more important — imperfect as that consensus is. It is through this consensus-building process that new insights will come to light and collective corrective action will be pursued. I recommend that the group feel free to intelligently compromise on terms and meanings in order to get to a shared understanding of the causes. Fortunately, most IT failures are not so philosophically knotty.

For completeness, below is a depiction of the causal factors contained with the example in the introduction (see Figure 2). This is a fishbone diagram in which the main outcome (i.e., incident, problem, or effect) is the "spine" in the diagram and the factors (i.e., causes) are the "bones" connecting to the main outcome. Other visualization techniques can be used. A fishbone diagram is a hierarchy and can be also represented as an inverted tree diagram.

Factors typically come in flavors. In IT, factors cluster around the following concepts:

- Equipment
- Software
- Data inputs
- Procedures and processes
- Human communication and coordination
- Skills and education
- Workplace design
- Social and cultural

A common problem with causal factor analysis is that the AAR team prematurely accepts and overestimates a factor in one of these categories (e.g., education or training) without deeper probing. To perform a good analysis and synthesis of the data, the team should have sufficient cognitive diversity and be composed of individuals who can challenge each others' thinking. The team should be good at continually asking, "Why did that happen?" to each description of a cause. This helps protect against premature agreement to a superficial causal analysis. Other parts of the methodology help in this regard, including the common factor analysis and the blue sky scenario. Root cause analysis can get somewhat involved. Some of the ideas in this report are drawn from Bjorn Andersen and Tom Fagerhaug's excellent book on the subject.[3]

Common Factor Analysis

After the team has settled on a theory of what happened and documented that theory in a causal diagram, such as a fishbone diagram or other visualizations, the next step in the process is to identify common factors, or systemic causes. Common factors are those causes that appear throughout a causal analysis. For example, if an IT shop has inadequate staff development and training processes, it is likely that a lack of training will show up in many different places, including software engineering, system administration, production control, and so on. In a complex incident, insufficient skill

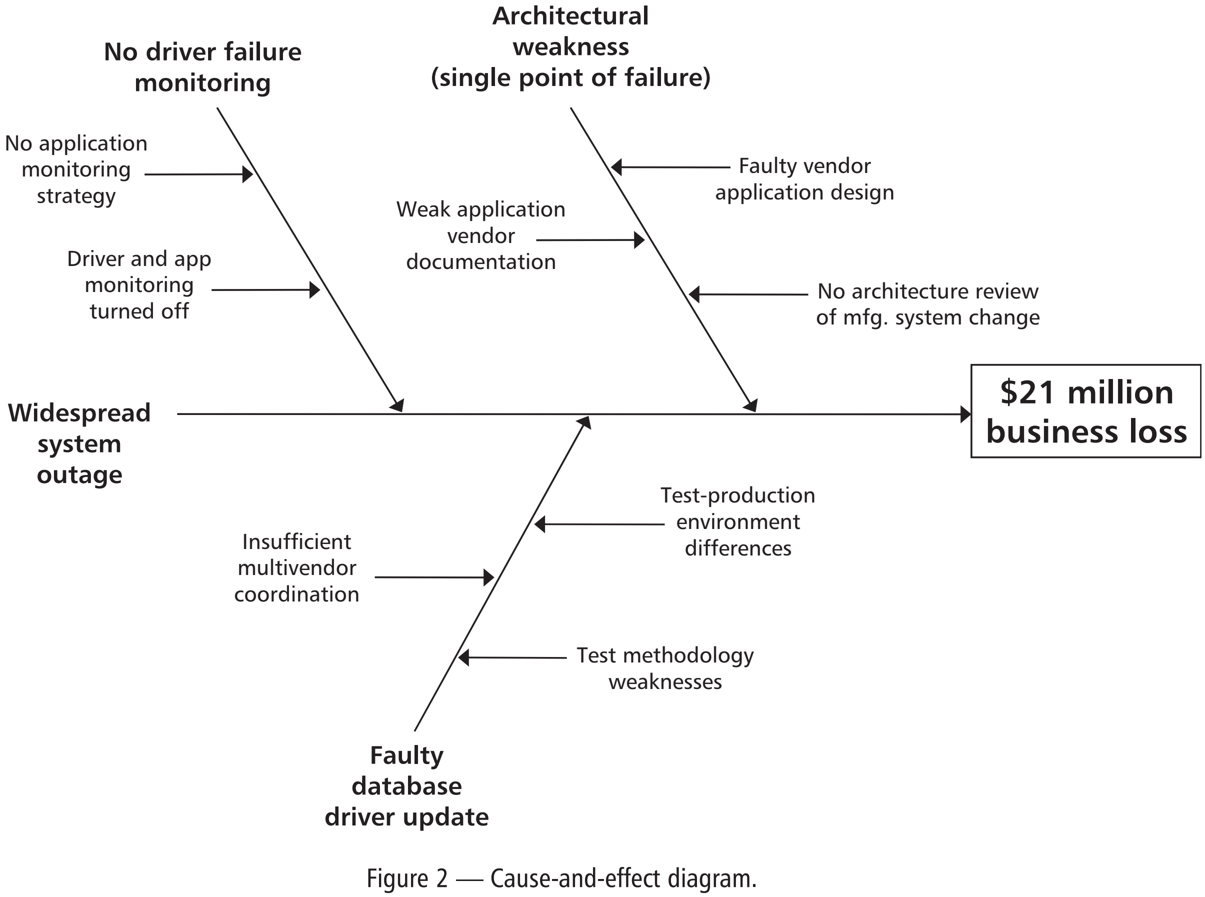
No driver failure monitoring
No application monitoring strategy
Driver and app monitoring turned off
Architectural weakness (single point of failure)
Faulty vendor application design
Weak application vendor documentation
No architecture review of mfg. system change
Widespread system outage
$21 million business loss
Test-production environment differences
Insufficient multivendor coordination
Test methodology weaknesses
Faulty database driver update

Figure 2 — Cause-and-effect diagram.

development is likely to show up as a cause in multiple places within the causal diagram.

The AAR team should then examine the causal model and identify these common factors. Common factors are slightly different than root causes. Some root causes are common (shared by multiple effects or multiple other causes), but some are not and are isolated or narrow in scope. Common causes are likely to be ones related to common IT processes, that if improved might generate improvements outside of the current incident. For example, improving staff development and training in response to a specific incident will certainly help prevent future similar incidents but improving staff development and training processes that can improve all sorts of other staff skills will have much broader benefits.

In this regard, common factor analysis can be considered systemic analysis, where the AAR team stands back and looks at the overall system of how IT work is done and identifies linkages and improvements that can go beyond the initially narrow scope of the incident. Rather than thinking about IT as a collection of independent parts, systemic analysis conceptualizes the many relationships and interactions, designed or emergent, across all the parts.

Obviously, the more experienced and skilled IT veterans will be better at common or systemic factor analysis. This is precisely why it is important to get the right type of experience and talent on the AAR team. Not only will these more experienced and conceptual IT employees add to the quality of the AAR, less experienced IT employees will learn from them along the way.

Blue Sky Scenarios

Not commonly discussed or done, I find blue sky scenarios immensely helpful. What are blue sky scenarios? These are make-believe versions of the incident that the team puts together in which there are no or nearly no constraints: all factors are manageable, money and time are not constrained and are plentiful, and expertise is high for all IT staff. We start off a blue sky scenario brainstorming session with the following question to the team: “In a perfect world, what would have happened?”

The blue sky scenario is good for the team to expose each other’s assumptions about what a perfect solution might look like. Each IT person has a slightly, if not radically different, conception about a perfect solution. Discussion of those differences can yield some interesting insights about potential solutions. But there are deeper psychological reasons for conducting a blue sky scenario.

In most incidents, the mood of the team is colored by the magnitude of the problem and the personal and professional implications. In this tense mood, employees are likely to attend heavily to the perceived threat at hand (the incident), collect a lot of data to help explain the event, and then build a model that fits the collected data. The more stressful the incident, the more ruthlessly people employ this approach. While this is good for data collection and problem analysis purposes because it tends to develop a wealth of data and lots of detailed thought, it is detrimental to the synthesis and solutions development part of the process, which requires innovative and creative thinking.

> **The blue sky scenario is good for the team to expose each other’s assumptions about what a perfect solution might look like.**

Essential to creativity and innovation is establishing the right emotional tone or mood, which needs to be relatively risk-free and positive. When our minds are not threatened, we are more likely to use our own cognitive processes to solve problems rather than extensively examining external “threat” data. A blue sky scenario helps foster this mood change. When all constraints are relaxed, the team quickly shifts its thinking to what could be done to make things better. And the idea that money, time, and expertise are not limited forces the team to think sometimes wild and crazy, but optimistic, thoughts.

I have found that some IT workers have difficulty truly accepting the blue sky mentality. IT workers have been so focused on living with and managing constraints that they will instinctively reject blue sky thinking as wishful if not dangerous. Sometimes I have had to coach these workers into accepting the idea that blue sky thinking is safe, productive, and helpful. IT workers know that implementing changes costs money and takes time. From here, many IT workers have taken that aversion and applied it back to their thought processes about solutions. They assume that *simply talking about implementing changes also costs money and takes time* and hence avoid the discussions. Another reason for this aversion to discussing blue sky scenarios may lie in how the IT unit makes implementation and architectural decisions. If the decision-making process is defensive and conversational, IT employees will learn that discussing what can be perceived as "wild" system changes is uncomfortable or risky and will naturally avoid discussing and hence thinking about innovative ideas.

Senior IT management will need to ensure that the overall IT culture regarding discussion and deciding of system changes is congruent with a good AAR process that can find creative and innovative solutions. Establishing a creative mood or climate is essential.

The blue sky scenario is designed to give the AAR team the permission and the right mood to think creatively in a way that can challenge any assumption.

The blue sky scenario also encourages the team to conceptualize the problem at boundary conditions, even if the boundary is insane or unrealistic. For example, believing that database drivers can be adequately tested to deal with rare, idiosyncratic production environments will spur thinking about how that might actually be done, especially if the vendor relationship can be leveraged. This might force the team to reconsider an unmanageable factor as one that can be managed. *While this line of reasoning may be ultimately unproductive, it is even more unproductive to prematurely kill this line of reasoning.* Who knows what creative and ridiculously inexpensive solution may follow next? It is free to think; it is expensive to implement. The more the team thinks creatively, the less expensive future implementations will be. The blue sky scenario is designed to give the AAR team the permission and the right mood to think creatively in a way that can challenge any assumption. After the team exhausts the blue sky scenario, they may need to revise their causal model and common factors. For this reason, I tend to think of the timeline, the causal factor analysis, the common factor analysis, and the blue sky scenario as four deliverables completed iteratively rather than serially.

Recommendations and Followup Actions

While the intellectual heart and soul of the AAR is in the factor analysis, the body of the AAR is in the followup actions. While it might be fun to intellectually ponder the complexities of an IT incident and brainstorm about crazy solutions, it takes work to turn the results of that insight into actual improvements. Making sure the AAR team has people who excel in project and task management is essential, especially for this part of the AAR.

After the team has settled on the causal model and common factors, the team will then need to come up with recommendations and tasks that make improvements. Followup actions should be measurable and verifiable, assigned to individuals or teams, have due dates on them, and be periodically reviewed for completion. While the after-action review will produce recommendations and followup actions, the IT organization needs to have in place a process for ensuring that followup actions are monitored. Again, having a TQM framework in place helps makes this so.

Normally, followup actions flow freely from the prior work and are not problematic. If the incident is complex and the outcome serious, many eyes outside of IT will be focused on the followup actions as well. These stakeholders may not trust the IT organization or carry a very different model of what transpired and thus, what ought to be done.

If the followup actions recommended are challenging and expose critical weaknesses and the overall corporate or IT culture is defensive, even senior IT management may wish to color or cover up the followup actions or portions of the AAR. Even in very open and trusting environments, valid but out-of-the-box recommendations will require some explanation and persuasion. If the followup actions carry significant budget impacts and costs, additional decision makers may be involved who are outside of IT.

In this case, how the other business units and the IT unit get along is critical. A beautifully conceived and executed AAR process placed within a very dysfunctional IT-business relationship will only be able to handle smaller issues that involve resources exclusively at the control of the IT senior managers. Anything larger will require collaboration with key decision makers outside of IT. This leads us to an interesting question. While the CIO and IT senior managers do need to attend to IT and business unit relationships independent of an AAR process, should the AAR process be opened up to include business unit staff to facilitate their understanding and acceptance of recommendations?

To answer this question, one probably only needs to examine the nature of trust within the corporate culture. The AAR process requires a high degree of trust between members of the AAR team — the different units with IT and between all the layers of

IT management, including the CIO, the IT senior managers, frontline managers, and IT employees. If you include business unit members within the process, you will have to ensure that the same level of trust exists across the IT-business unit boundary. If it does not, the AAR process will erode in the long term as more IT employees see evidence of the lack of trust and begin to clam up, spin things, and engage in CYA tactics.

Complicating matters further is the fact that complex IT systems have an incredible amount of process coordination and knowledge sharing across the IT-business unit gap. A major incident is likely to have many factors relating to this process and knowledge coordination. The company will be able to implement difficult and demanding followup actions only to the degree that the overall corporate culture and management practices allow. If the organization as a whole has a strong TQM framework, the AAR process will most likely be able to produce and manage difficult followup actions. If it does not or if other cultural impediments exist, the CIO will need to help the organization remove the impediments. While IT may manage the technical infrastructure by itself, it takes a village to be able to diagnose and fix complex IT problems.

Review and Share

Once the after-action review is complete and recommendations and followup actions are moving forward, what's next? On the one hand, the AAR has served its purpose. On the other hand, the AAR may have value as an historical document that can educate others. As the AAR is shared, senior managers should remember that the AAR should not be used as a tool for an individual performance review. As soon as employees perceive that the AAR can negatively affect their performance review, all employees will learn to color their comments and data to ensure a good performance review, potentially damaging the AAR. While an AAR may indicate specific individual performance review issues, the AAR itself should not be the tool to address those issues. IT managers have other tools at their disposal including the normal performance evaluation processes. When sharing the AAR more widely, IT senior management should always be on the lookout for signs that employees perceive the AAR process to be detrimental to their careers. View the risk identified in an AAR process as a risk that the entire team shares, not one individual.

The more significant the incident, the more likely the IT shop will find value in sharing the AAR documentation. I recommend briefing those areas of IT (and relevant business units) that hadn't been previously privy to the process. While some of these units might not be directly involved in the incident or its followup actions, many units will be discussing the incident among themselves. Briefing these units gives them the same set of facts. In addition, as the documents and findings are shared, occasionally people will point out corrections and omissions. The AAR team can then alter the documentation as needed.

In the realm of IT, AARs do have a practical shelf life. Technology changes, systems come and go, and many of the causal theories and recommendations contained within the older AARs are no longer directly relevant. In that case, the AAR becomes a part of the repository of artifacts that has less day-to-day practical value while still retaining importance from an audit, legal, or historical perspective. Some of the AARs may turn into case studies and be used as teaching aids when developing IT staff skills.

The purpose of the AAR is to put into practice improved patterns of activity: processes, procedures, and ways of handling IT hardware and software. The value of the AAR is not in the documents but what from the documentation has been put into practice and has altered the minds and behaviors of IT staff.

OTHER AAR TOOLS

The various industries that perform root cause analysis, after-action reviews, or failure mode and effects analysis (FMEA) have developed a collection of useful tools that IT shops can use in their AAR process. These tools span the range of the process, including data collection, problem understanding, root cause determination, and development of followup actions. Some of these tools include:

- **Histograms.** All IT shops should be familiar with histograms, which simply plot some kind of performance data overtime. Network and data center engineers have long used histograms to understand system performance. Invariably, an IT AAR process examines or develops histograms in order to understand the problem better.
- **Pareto analysis.** This is a simple technique that lists the top occurrences of an event or incident by a category, with the most frequent occurrence first. The display is often a bar chart, sorted by frequency, from the most frequently appearing category to the last. Software engineers have used Pareto analysis to list

software defects found in systems by the type of software defect. This type of analysis lets an AAR team understand the frequency of problems or events and can help the team understand the problem better.

- **Flowcharts.** These depict a sequence of steps. (Figure 1 is an example of a flowchart.) AAR teams will often look at or construct their own flow charts to depict a sequence of actions.
- **Business process diagrams.** These are also a kind of flowchart but are useful for depicting business processes that may involve non-IT staff. AAR teams will look at or construct business process diagrams for those problems that cross the IT-business unit boundary.
- **Performance/importance matrix.** This is a simple grid with two axes. The vertical axis usually depicts the performance of a system, activity, or an event (poor to excellent), and the horizontal axis depicts the importance (low to high). Anything can be charted in this matrix. The lower right quadrant of this matrix is the most important one. It contains those items that are most important but have the least performance level. The AAR team will use this tool to help visualize any number of systems, activities, skills, or events. This in turn lets them better prioritize their efforts or gain a better understanding of overall performance of the IT areas in question.
- **Brainstorming.** Often overlooked and underdeveloped, brainstorming techniques are critical for finding inventive solutions to difficult problems. Brainstorming can be done verbally together as a team, or via several written techniques (to prevent domination of the conversation by a few). Brainstorming can be unstructured, where anyone can jump in, or structured to more carefully control participant involvement. A critical component for brainstorming to solve complex IT problems is to prevent the group from prematurely discarding an idea. An AAR team needs someone skilled at facilitating brainstorming sessions especially for constructing a blue sky scenario.
- **Is/is not list.** When diagnosing a critical system failure, IT teams invariably generate is/is not lists, as shown in Table 1.
- **Ranking.** When presented with a list of possible causes or a list of theories of causation, AAR teams can "vote" for the likelihood that one of the causes or one of the theories is the true explanation. Again, this is useful for generating discussion about what everyone believes to be a causal factor or what no one believes is a causal factor. This is a good tool for understanding dominant beliefs and outliers. I have seen simple ranking promote good discussions about causes, with typically a "lone wolf" arguing for a cause that the team is not considering. Normally,

Table 1 — Example of an Item in an Is/Is Not List

Question	Is	Is Not	Distinctions
What occurs?	Application server prematurely drops database connections resulting in a Web page error.	Application server maintains all other connections to other systems.	Only the database connection is in question.
Where?	The payroll system is affected.	No other systems affected.	Obvious!
When?	Randomly. No pattern detected.	No testing scripts can reproduce the event.	The timing of the event may not be related to user system use patterns.
Extent?	Not sure. It looks like all payroll users can be affected. All payroll application servers are involved.	This does not appear to be limited to one user, a class of users, or a limited set of servers.	
Who is involved?	Payroll functional users and IT are involved in this problem.	No other groups of people should be involved yet.	No other systems appear to affect or are affected by this problem.

data collection serves as the validation mechanism for final inclusion or exclusion of a cause.

- **Paired comparisons.** If the relationships between items in the ranking list are more complicated and seem to defy ranking, it is useful to do pair-wise voting on all the factors. Think of it as a sports tournament in which every team (a factor) plays every other team. If you have four factors in question, have the AAR team compare two factors. Assign a score of 1 to the winning factor (the more likely factor) and a score of 0 to the losing factor (the less likely factor). Continue until all items have been compared with each other and then rank order the factors.

All of these tools are just that. They do not replace the hard work an AAR team must do in order to develop a good theory of what happened. These and other tools are valuable for generating discussions, insights, and ensuring that the team isn't overlooking possible causes or is biased in its approach.

CONCLUSIONS

After-action reviews should be considered a piece in a larger puzzle of IT operational improvement. While important, an AAR process is intertwined with the IT culture and with other IT processes. AARs are effortful and infrequently done. Other IT work is done far more frequently. If the AAR process is not implemented as part of a larger plan for changing or maintaining an effective IT culture, most of the benefits of an AAR will be lost.

A CIO can potentially use an AAR process as a lead element in a TQM or culture change plan, but it must be followed up with — and embedded within — a multilevel organizational development plan. The AAR process by itself won't generate all the cultural and organizational change needed to keep an AAR process alive and healthy. Too often, we have seen IT shops become callous and indifferent to things like an AAR or a TQM framework. Conversely, I have seen organizations keep these frameworks alive and self-sustaining. The framework is not the solution. The integration of the framework into the culture is.

I believe it is imperative for all IT shops to have an equivalent to an AAR process, even for highly outsourced environments. The information and knowledge that flows between the inhouse or outsourced IT and the business unit can prove valuable for improving current operations and finding service improvements that can make a difference to the company and its customers. The AAR process needs to fit into a broader culture of openness, frankness, and an ability to discuss difficult issues in group settings while maintaining productive human relationships. Companies that can effectively use an AAR process may be endowed with both cultural and operational advantages that will let them prevent or at least handle an IT crisis gracefully, quickly, and effectively.

ENDNOTES

[1] From an interview with Brigadier General Harold W. Nelson (retired), former Chief of Military History for the US Army.

[2] Headquarters Department of the Army. "A Leader's Guide to After-Action Reviews." Training Circular 25-20, 30 September 1993 (http://35.8.109.2/resources/TC25-20AARs.pdf).

[3] Anderson, Bjorn, and Tom Fagerhaug. *Root Cause Analysis: Simplified Tools and Techniques.* 2nd edition. ASQ Quality Press, 2006.

ABOUT THE AUTHOR

Vince Kellen is a Senior Consultant with Cutter's Business-IT Strategies and Business Intelligence practices. Mr. Kellen's 25-year experience involves a rare combination of IT operations management, strategic consulting, and entrepreneurialism. He previously served as VP for Information Services (CIO) at DePaul University, where he won *CIO* magazine's coveted Top 100 award in 2007. Mr. Kellen also served as a partner with strategy consulting firms, where he helped *Fortune* 500 and midsized companies with business and IT strategies, IT organizational development, customer experience management, customer relationship management (CRM), and analytics.

A national and international speaker on CRM, customer experience management, and business and IT strategy issues, Mr. Kellen has authored four books on database technology and more than 120 articles and presentations on IT and business strategy topics. He holds a master's degree from DePaul's College of Computing and Digital Media and is currently completing his PhD in computer science at DePaul. Mr. Kellen was also an adjunct faculty member at DePaul for 10 years, where he helped launch DePaul's graduate program in e-commerce — one of the nation's first graduate programs concentrating on e-commerce — and designed and taught graduate courses in enterprise architecture, CRM technologies, and portals. He can be reached at vkellen@cutter.com.

Consulting

Business-IT Strategies

Ensure IT Is Creating True Value

Cutter Consortium offers advice and guidance from world-renowned consultants. The Consortium features a faculty whose expertise and credentials are unmatched by any other service provider.

Moreover, unlike many other consulting firms that use senior partners to sell a job but then assign junior staff to actually perform the work at the customer's site, Cutter has no junior staff and deploys only its expert Senior Consultants, Fellows, and Technical Coaches on every assignment. Cutter's expert practitioners have considerable management, technical, and domain-specific experience assisting *Fortune* 500 and other organizations with everything from IT strategic planning and organizational development to enterprise architecture, program management, data management strategies, benchmarking and measurement, and more.

In addition, Cutter does not rely on off-the-shelf solutions but instead customizes every solution to meet each client's unique needs based on the client organization's business drivers, culture, technology history, and budget.

The Consortium's great strength is that it can draw on its more than 150 best-in-class consultants to assemble the ideal team for your organization, tackling any challenge that might arise and offering a complete solution from assessment through implementation.

Value Project Portfolio Management

Every IT project should deliver business value. The challenge is that traditional project management and PMOs focus on "on time" and "on budget," not achievement of business value. Cutter's Value Project Portfolio Management (Value-PPM) approach ensures organizations achieve all three goals.

Value-PPM is compatible with agile project management and development methodologies, while supporting a detailed PMO process. Value-PPM takes an investment perspective — justification, prioritization, and monitoring — to give visibility to projects as they are implemented, with respect to cost, schedule, and performance. It enables companies to answer key questions, such as, "Are all projects on track for achieving business value?" and "Which projects require management intervention?"

Cutter's Bob Benson and Tom Bugnitz can assist your organization in developing a Value-PPM approach, yielding:

- Comprehensive project management reporting capability
- Complete view of all projects throughout the lifecycle, tailored to both business and IT management
- Rules-driven "dashboards"
- Assessment of business impact and benefits
- Detailed review and rollup of status of each project by business unit and enterprise
- Business-based prioritization and risk assessment

This Value-PPM process will ensure that your IT investments are coordinated within the business unit and selected based on their merits in supporting business goals. It will ensure that IT investments conform to the appropriate standards (e.g., enterprise architecture and security). And it will ensure you maximize the value of the business units' total IT expenditures through a formal IT investment management process.

Infrastructure Portfolio Management

Infrastructure Portfolio Management is used to control and manage infrastructures, to answer the overall question, "Are we getting sufficient results and value from our investment in infrastructure?" Of particular importance for companies with multiple geographies, both domestic and global, are the standard tools and templates that we apply to ensure a common method for assessing and managing infrastructure for both centrally managed and locally administered infrastructures.

Bob Benson and Tom Bugnitz provide specific tools and processes, depending on your needs, to accomplish:

- A complete inventory of the infrastructure services that IT provides the business, including the support of new projects,

Cutter Consortium
37 Broadway, Suite 1
Arlington, MA 02474, USA

Tel: +1 781 648 8700
Fax: +1 781 648 1950
Web: www.cutter.com
E-Mail: sales@cutter.com

existing applications, user services, and management services
- Thorough assessments of the following:
  - — Strategic alignment with both business and IT strategic intentions
  - — Service level and quality, including reliability and accuracy
  - — Costs
  - — Technical risks
  - — Security risks
- IT investment strategies
- Security technical and business risk assessments
- An assessment of governance practices for making IT investment decisions

Financial Management of Information Technology

Bob Benson and Tom Bugnitz present a comprehensive set of tools for managing all the financial aspects of IT and educate your organization in how to apply IT Financial Management information so it can successfully manage corporate and IT capital and operating budgets. We equip the business executive, CIO, IT executive, IT financial manager, and CFO with the tools and processes they need to answer the following questions:

1. Where are we spending IT dollars? That is, what are the line items in the IT service portfolios, and how much are we spending on each one?
2. Are there IT dollars being spent on low-business-impact activities that could be better spent elsewhere in IT, or even elsewhere in the company?
3. What are the technical and business risk profiles of new IT investments? What are the risk profiles of ongoing (legacy) activities?
4. Do our budget and management processes put our financial resources in the areas with the highest business impact?
5. Are we managing all our resources as well as we can?
6. Wherever we spend money, do we know what we are spending, and are we doing it with thought?
7. Do we have the right management processes to control IT costs, plan IT activities, and choose the best IT investments?

The specific tools, assessments, and processes applied are dependent on your needs. Portfolio Management, covering applications, infrastructures, services, and projects, is applied extensively.

Strategy Meeting Facilitation

Cutter Senior Consultants and Fellows are experts at facilitating IT strategy sessions. They bring to the process a deep knowledge of business and IT strategy and many frameworks that can be applied, as well as extensive, hands-on experience assisting companies with this process. Their business expertise enables them to point out where a firm may be experiencing difficulties that are entirely normal and to be expected, and the likely timetable for resolution, as well as cases where the issues are more problematic and how they might best be resolved.

Prior to the event, Cutter's Senior Consultant will interview key participants individually and work with the executive team to plan the format, approach, and agenda for the session. Cutter will facilitate the session and summarize the group's conclusions, combined with our recommendations, into an action plan.

Optimization Review

The Optimization Review, typically two days, is designed to identify the most promising opportunities for change by identifying what separates a company from the norm. Areas in which an organization's capabilities are better or worse than the average represent terrific areas of opportunity. The successful processes/approaches that are already working in the organization can be propagated. And areas where an organization is performing at a below-average level are, by definition, the easiest to fix, because the average firms in this marketplace that are performing better in this area provide proof that improvement can be achieved.

This unique approach, developed by Cutter Fellows Tom DeMarco and Tim Lister, is designed to finesse change resistance and has proven repeatedly to be both a tremendous morale builder and an extremely effective change mechanism. The process creates shared goals among the participating factions, generating excitement about the opportunities that emerge.

The Consortium team will help the participants identify strengths and the directions in which these will lead the firm, a commonality of purpose, challenges the firm faces, and ways of addressing these. Mr. DeMarco and Mr. Lister present their observations and conclusions to all the participants, including management, at the end of Day 2. A final report sums up observations and recommended actions.

In-Depth Business-IT Assessment

Cutter can provide a high-level assessment of your organization's framework for achieving business excellence. We'll evaluate the internal process planning, structure, and management of key business areas that directly or indirectly affect business results. We will identify specific areas of opportunity for improvement and make recommendations to help address any gaps or weaknesses in the key areas of focus.

Specific subjects to be reviewed and discussed can include strategic planning, business/product structure, leadership structure and organization, risk management, portfolio management, metrics/goal management, standards and documentation, program management, customer interface, business process structure, system engineering process, software development process, human resources development and management, technology transition, and enterprise architecture.

The key deliverables of the Assessment are a summary report with an action plan and the presentation of these findings to the client executive team. Clients receive:

- A totally objective review, including gap analysis, of their current situation by a team of experts
- Recommendations regarding what's needed in a governance model, strategic plan, business processes, enterprise architecture approach, etc., that all parties can support
- Support, if requested, "drilling down" and tackling domain-specific needs, such as enterprise architecture

About the Practice

Business-IT Strategies Practice

The Business-IT Strategies Practice area focuses on the intersection of business and IT. Through the subscription-based Advisory Service, the Business-IT Strategies team of Senior Consultants guides companies to optimize their IT investments by ensuring they validate business requirements prior to making investments in technology, technology acquisition strategies, and day-to-day management of technology.

Consulting and training services within this practice area are customized to meet your needs; they cover assignments such as harnessing IT as a competitive weapon through sound business-IT alignment, developing an IT strategic plan, and reorganizing and transforming your IT department.

The Business-IT Strategies Practice guides you to identify the IT investments that make the most sense for your business, avoid those that fail to support your business objectives, and position your enterprise so it can leverage IT for competitive advantage.

Products and Services Available from the Business-IT Strategies Practice

- The Business-IT Strategies Advisory Service
- Consulting
- Inhouse Workshops
- Mentoring
- Research Reports

Other Cutter Consortium Practices

Cutter Consortium aligns its products and services into the nine practice areas below. Each of these practices includes a subscription-based periodical service, plus consulting and training services.

- Agile Product & Project Management
- Business Intelligence
- Business-IT Strategies
- Business Technology Trends & Impacts
- Enterprise Architecture
- Innovation & Enterprise Agility
- IT Management
- Measurement and Benchmarking Strategies
- Enterprise Risk Management & Governance
- Social Networking
- Sourcing & Vendor Relationships

Senior Consultant Team

The Cutter Consortium Business-IT Strategies Senior Consultant team includes seasoned experts in the business technology arena. Several are former CIOs; many have served as business management consultants; others have served as professors at prestigious universities. Collectively, the Senior Consultants on the Business-IT Strategies team have decades of experience both inside corporate IT and business groups, and working with organizations in a consulting capacity. The team includes:

- Stephen J. Andriole
- Robert D. Austin
- Steve Barnett
- Robert J. Benson
- John Berry
- Steve Bradley
- Thomas L. Bugnitz
- David J. Caruso
- David R. Caruso
- Robina Chatham
- Eric K. Clemons
- Mark Cotteleer
- Christine Davis
- Michael Enright
- Daniel Hjorth
- Maxwell Hughes
- Vince Kellen
- María Luisa Kun
- Steven Kursh
- Tim Lister
- Michael C. Mah
- Julio César Margáin
- Ciaran Murphy
- San Murugesan
- Rogelio Oliva
- Ken Orr
- Wojciech Ozimek
- Pat Patrick
- Patricia Patrick
- Jerry Peterson
- Robert Phaal
- Gabriele Piccoli
- David N. Rasmussen
- Kenneth Rau
- Ricardo Rendón
- Alexandre Rodrigues
- Jorge Ronchese
- Michael Rosen
- Mike Sisco
- Borys Stokalski
- Rob Thomsett
- William Ulrich
- Jim Watson