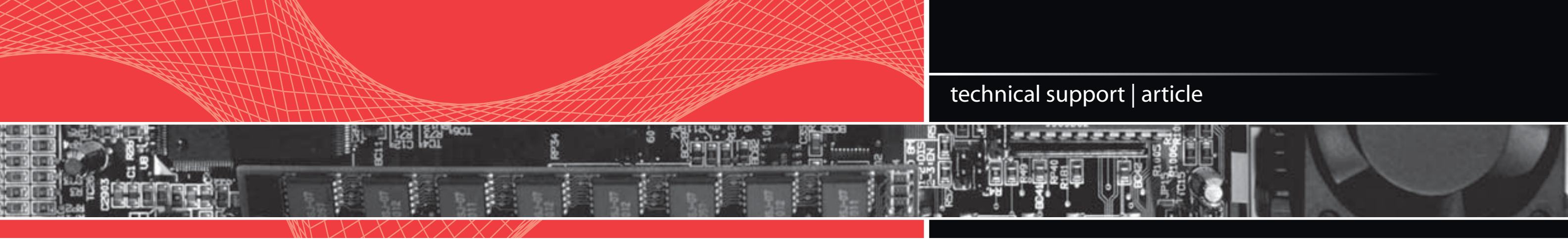

# HRO—The New TLA To Solve All Our Problems

By Lionel B. Dyck

HRO, the acronym for High Reliability Organization, is the next in a long line of acronyms that have come into the world of data processing in an attempt to improve the reliability, the availability, and the serviceability (also known as RAS) of the data processing environment. Since I started working with computers in the early 1970s I've heard about or been involved with RAS, TQM, CI, CQI, Six Sigma, Lean Six Sigma, ITIL, PBR, and now HRO. What HRO is and why you should care will be explained shortly. Will HRO, or indeed any of these process models, improve the availability and reliability of our information technology infrastructure? In short—NO. Not in and of itself. HRO, like all the other process models before it, is a mindset that we all should have when working in any environment where mistakes or failures can have a negative impact. In the financial services sector this impact could be the loss of millions of dollars, while in the health care services sector an impact could result in a patient (a friend, close relative, or even yourself) being killed or crippled in some way for life.

Is HRO the same as ITIL, Six Sigma, etc.? The short answer is yes in that it is an attempt to improve the quality of the product and the reliability of the processes that create that product. The other answer is no in that HRO is a culture rather than a set of predefined processes. The key to HRO is being "mindful" about your environment—what problems could arise and how can they be prevented.

HRO, or High Reliability Organization, started with studies during World War II in improving equipment design and maximizing human effectiveness. It got a boost from a study by Charles Perrow in 1984 about the accident at Three Mile Island which came to be known in the literature as the Normal Accident Theory (NAT). This theory is basically that accidents are going to happen; they are inevitable.

Then in 2001 Karl E. Weick and Kathleen M. Sutcliffe published a book entitled *Managing the Unexpected: Assuring High Performance in an Age of Complexity,* which seems to have become the go-to book to learn about High Reliability Organizations or HROs. The difference between NAT and HRO is that with an HRO the emphasis is on being prepared for accidents and how accidents are responded to when they occur. Responding to accidents is how an organization improves and eventually prevents accidents.

The book gives several examples of HRO type organizations that embody the five guiding principles described next. One of the more interesting ones is that of an aircraft carrier where you have a highly complex and stressful environment staffed by 19- and 20-year-olds being managed by officers who are a few years older. An accident there could result in the loss of a multi-million dollar aircraft and millions of dollars in damage to the ship. Yet they perform this job very well and with very few accidents. How do they do this? By being prepared, being well trained, having very clearly documented procedures, and everyone having the authority to speak up and report a potential problem. For example, if an airman notices that a wrench is missing from a toolbox, he is required to report it, at which point the entire flight deck operation may be shut down until the wrench is found. They also will periodically do a walk of the flight deck to check for any extraneous debris which could cause damage to a plane taking off or landing. Everyone from the Captain to the newest airman will gather on the flight deck and walk from one end to the other (from the bow to the stern) picking up everything from a loose screw to that missing wrench.

HRO has five guiding principles:

- **Preoccupation with Failure**

  Every failure is important, from the smallest mistake to the largest. Each and every failure and near failure is reviewed as a learning opportunity so that the failure can be prevented in the future or the response can be improved to reduce the impact of the failure. Everyone is encouraged to report any failures as well as near misses, including those that they cause and there are no repercussions (e.g. you're fired) unless the cause was an overt act of negligence. One thing to always keep in mind is what Mr. Murphy said many years ago slightly updated "If anything can go wrong it will, and at the worst possible time."

- **Reluctance to simplify interpretations**

  When you simplify you lose information, which is why an HRO tries to avoid simplifying. The more information that you have, the more you can learn from it and the more complete a set of instructions, the less chance of a mistake being made.

- **Sensitivity to operations**

  An HRO realizes that those closest to the action know the most about what is going on. Everyone is encouraged to speak up about anything that is abnormal or perceived to be a potential problem, realizing that those doing the work are most likely to see anomalies before anyone else.

- **Commitment to resilience**

  How fast are you in resolving a failure? An HRO will emphasize the Boy Scout Motto "Be Prepared" and will have walked through scenario after scenario and still be very aware that the next failure could very well be something no one imagined. Resilience requires that the HRO have the expertise to be able to improvise and to think quickly to resolve issues and thus relies heavily upon their experts (both in-house and vendors) to work together to resolve failures when they occur.

- **Deference to expertise**

  The HRO values expertise and encourages the training of its staff in all areas. When a failure occurs it is not those in the executive suites who resolve the failure but those closest to the failure. How many times during a failure has some executive or manager walked into the room, taken "control" of the situation and then proceeded to direct the recovery efforts, only to elongate the restoration process significantly? This will not happen in an HRO, although a member of management may "take control," it will not be to direct the action but to make sure that the experts have everything that they need to resolve the failure.

Looking at these one sees quickly that they are all common sense. Sadly, common sense is so very uncommon that we have to continue to remind everyone. Whether HROs are better or worse than other models is open to debate, but I can say that anything that takes someone who is complacent and makes them think about their job in a new way that improves both the quality of their job and their work product is good.

Many of the examples of HROs that are given in the literature are complex environments which experience a low level of regular change. This does not easily relate to the typical data center where change is the rule rather than the exception. While a stable HRO can have solid procedures that are well documented and tested, it is more of a challenge to an HRO where things change frequently. When the environment changes, the documented procedures must change and be verified before they are used again. This is something that is not always easy if a robust test environment is not available. When things change, the potential for failure increases and every scenario for anticipated failure must be revisited.

Those of us in the world of data processing know change very well. We are the ones who are making the changes and we are the ones who are also resolving the failures that occur when something does not go right. The longer we've done our job the more we've learned to look at a change and evaluate the potential success or failure of it. We've learned to review failures and have found ways to prevent them or to recover more quickly when they do occur. These are important traits that we need to continue to develop and to share with our peers. HROs then take these individual learnings and institutionalize them into the organization's culture.

Many of those working in the data center think nothing of doing a reboot of a server to resolve a problem. That may be the solution, but have they captured enough information to determine why the server needed to be rebooted? Can they report the incident to the vendor or developer with enough information that the issue can be prevented in the future? In some cases the pressure to restore service prevents the capturing of the necessary information to determine the true root cause of a failure. And when that happens we have to feel comfortable letting management know that unless we have the time to capture the necessary documentation to do a root cause analysis that the problem is likely to reoccur again and again. An HRO will factor that into any failure recovery model so that there are guidelines, such as no more than 10 minutes to perform data capture before doing a reboot.

An HRO, as mentioned above, encourages reporting of both failures and near misses. This is not violating the 5th Amendment (protection from self-incrimination), rather it is done in an environment where the individual feels comfortable to report issues without fear of losing their job. If this environment does not exist, failures and near misses will not be reported and thus the organization cannot learn from them and cannot prevent them from happening again and again. HROs view failures as a "window" into an organization's operation providing valuable learning opportunities. If your organization has a "blame first and fire" culture instead of an open reporting culture, this can be difficult to implement without full support from high in the organization, but you can do this anonymously to start with.

Does this take a CIO leading the charge to become an HRO? It wouldn't hurt, but this is something that each of you can influence from whatever position you have within your organization. If you are a systems programmer on the front line you can model the HRO behavior to your peers and to your management. Keep track of all failures and near misses and keep your management informed of what you've learned from them and how you've used what you've learned to prevent other failures. When you make a mistake take ownership of it and report it if you feel comfortable doing so. When one takes personal responsibility for accidental failures that they've caused, they demonstrate character. While you may not be rewarded (right away) you will demonstrate something that is key to an HRO and you will be noticed for the better in most cases.

Becoming an HRO takes time. It does not happen overnight and to do it right requires a commitment from senior leadership because there will be costs both in dollars and in resources. To do anything right requires that the right amount of resources be available, both in staffing (so you don't have burned out and exhausted staff working their normal shift and then working the midnight shift to install changes and then returning to the office at 8am for their normal shift again) and in keeping your software and hardware reasonably current so you aren't using outdated or unsupported software or hardware to operate your environment.

This has only touched the surface of HRO and I would encourage you to learn more about this concept. Many of you remember RAS (reliability, availability and serviceability) which was used by IBM many years ago to sell their software and hardware. That was an early HRO model. It is still valid as are TQM, CI, CQI, ITIL, Six Sigma, etc. Learning about any or all of these will give you new insights that you can use to do your job better and give your customers better service and support.

There are not a lot of resources on High Reliability Organizations that I've been able to find to date, but I strongly suspect that within the next year or two there will be many resources published. There is already a HRO annual seminar (next year in Normandy, France I understand) and it is moving from the academic world into the business world quickly.

In closing remember that you need to "be prepared" and that "if it can fail—it probably will fail and it will fail at the worst possible time."

Some useful resources are:

1. **HRO Has Prominent History** by Karlene H. Roberts, PhD http://www.apsf.org/resource_center/newsletter/2003/spring/hrohistory.htm
2. **Managing the Unexpected: Assuring High Performance in an Age of Complexity** by Karl E. Weick, Kathleen M. Sutcliffe. Published 2001.
3. **Beyond Normal Accidents and High Reliability Organizations: The Need for an Alternative Approach to Safety in Complex Systems** by Karen Marais, Nicolas Dulac,

and Nancy Leveson, MIT March 24, 2004
http://esd.mit.edu/symposium/pdfs/papers/marais-b.pdf

4. **Safety, Reliability, Stewardship, and Regret: Contributions to Dependable System Design from the Study of Highly Reliable Organizations** by Andrew Koehler, PhD, Statistical Sciences, D-1, Los Alamos National Laboratory 12/16/2005
http://ti.arc.nasa.gov/projects/ishem/Presentations/Koehler_High_Reliability.ppt
5. **5 Habits of Highly Reliable Organizations**
http://pf.fastcompany.com/magazine/58/chalktalk.html
6. **High Reliability Organizations Conferences**
http://www.highreliability.org/

**NaSPA member Lionel Dyck has been working with computers for over 34 years and has written many articles over the years for this magazine including several on his open source z/OS SMTP mailing utility XMITIP. You can learn more about his open source tools at http://www.lbdsoftware.com.**