



The Upside of Y2K

John B. Hubbs
AverStar, Inc.

Despite the potential catastrophes of failure, a long-term view of solving the Y2K problem provides many positive benefits. The opportunities should be seized for more objective communications within the commands and for improved processes within our software shops.

Information technology (IT) industry pundits and the press predict all types of calamities to befall humanity because of the year 2000 (Y2K) crisis. Some of the worst-case scenarios may manifest themselves, and technology's reputation will take a beating from those most injured. Despite the near-term downside of Y2K problems, the long-term view indicates many beneficial aspects to "the crisis."

The major problem IT faces is described in the popular press as a "bug" or a "glitch." These terms are commonly used by practitioners for relatively minor problems that can be solved with minimal effort. The truth is that information technologists created the Y2K problem by conscious decisions; that is, we *designed* a crisis. But crises frequently have positive aspects when viewed from a long-range perspective.

Other articles have addressed the technical issues Department of Defense commanders and information technologists face. This article details the positive aspects of solving Y2K problems. When we look back on this era 10 or 20 years from now, we will appreciate what the millennium has done to transform our industry.

The common thread of the upside of Y2K is a focus on process. Whether a five-step Y2K process or the Capability Maturity Model (CMM), many software engineering processes will be enhanced by the year 2001. The positive aspects of Y2K can be grouped into four major areas:

- Improved structure.
- Heightened software awareness.
- Enforced accountability.
- Industrial maturity.

Improved Structure

After an inventory of systems (also a long-term benefit), the next step in

addressing an organization's Y2K situation is the decision to retire applications not worth renovating. Many of these older systems should have been retired many years ago. But we are reluctant to pull the plug on these stalwart applications. Building in the "millennium bug" when storage was a premium created unplanned obsolescence. After 1999, we will no longer need to maintain systems for which the functions will be delivered with more contemporary software. The cost to maintain applications will be reduced through

- fewer lines of code to support.
- a refreshed understanding of the remaining programs.

Organizations that started their Y2K activities a year or two ago took advantage of the opportunity to thoroughly review their systems. One health-care claims processing software shop that carefully examined all its applications in 1994 reported a reduction of more than 10 percent in lines of code compiled upon completion of their Y2K project. Organizations that have completed a re-engineering with their Y2K renovation also report less cycle time to execute the more efficient applications.

Shops that started later had to employ fully automated techniques without a comprehensive review to optimize systems before renovation. These shops see little code reduction and, when using windowing techniques, may experience a small increase. Regardless of the amount of reengineering, additional benefits to the renovation process include expansion of the regression testing suite.

On the hardware side, many commands experienced a proliferation of platforms over the past several years. Improved infrastructure provides many of these shops with the opportunity to standardize on a processor base that

should consolidate future maintenance costs. As commanders stabilize their platforms, they will also be using the Y2K process as an opportunity to reduce the multiplicity of operating systems and to benefit from economies of scale.

Heightened Software Awareness

Perhaps one of the most profound and subtle positive effects of the Y2K problem is the new level of dialog between senior managers and information technologists. Over the past 25 years, data processing has grown from an ancillary function to a ubiquitous necessity of modern life. Many managers in government and business, however, continue to see IT as a "back room" support function. The abstract nature of software once facilitated a manager's ability to relegate our profession to a support role.

With the millennium approaching, managers are more acutely aware of the role of software to the continued successful operation of the organization. Senior commanders and boards of directors now are aware of software risk management, project staffing, task status, metrics, and contingency planning. Although upper-level management need not be involved in the details of IT over the long range, their concerns about the Y2K project bodes well for the future of our profession. Their increased awareness of the processes required to create and maintain software should reduce the number of arbitrary, uninformed decisions in the future and foster better communications with technologists. Programmers at Johns Hopkins University report that their board of directors has a Y2K oversight committee that tracks progress and issues on a regular basis. Nontechnical managers at all levels will have a better

appreciation for software and the extensive testing required to get it right.

In addition to systems “owned” by a command, interfaces with other applications have increased with advances in telecommunications. With each command serving its own needs, we have unintentionally woven a complex web of interfacing systems. The Y2K issue forces us to realize that the interfaces are as important as the home-grown systems. As information technology practitioners, we should not let this opportunity pass us by to inform our managers.

Enforced Accountability

Partly because our work is viewed as a “black art,” we have been allowed slip-pages and nonperformance that would not be acceptable in other parts of the military. Software project failures and cost increases are not as publicized in the popular press as much as hardware fiascoes. Software projects have been given more time and money despite

- The shortage of documented requirements.
- Demands to meet an often arbitrary deadline.
- Confusion about the process.
- A changing technologic base.
- Frequent reorganizations.

Yet, this “new industry” has fostered unprecedented growth in an economy that was all but given up as lifeless 20 years ago when we feared being overrun by the Japanese. Most of this success was achieved by heroes who sacrificed a good share of their personal life to bring order out of chaos.

With an unchangeable due date that cannot be missed, we will be held responsible for our performance. The processes that must be enforced to be Y2K successful have grabbed the attention of senior officers (assuming they know they need a disciplined software process). Failure is not an option without bringing down the enterprise—be it a command, a business, or a government agency. Technologists will be fully accountable for Y2K success or failure.

Industrial Maturity

Any new industry experiences growing pains. The opportunities of the indus-

trial revolution at the end of the last century gave rise to labor abuses and monopoly powers. Through painful experience, a free society regulated itself to create a positive, constructive environment in which the then new technologies could flourish. Electricity was as misunderstood and ubiquitous at the turn of the last century as software is at the turn of the millennium.

Thus, the millennium change is our epiphany—the hidden art of software engineering will come out of obscurity to be seen and to serve in a more open and visible manner. This coming out will be accompanied by a renewed focus on the processes by which software is conceived and delivered. Those organizations that handle Y2K successfully will be seen as more progressive and able to deal with future challenges. Those who fail will join the manufacturers of muzzleloaders or will be relegated to forever catching up until they are overtaken by new market or regulatory forces.

The rapid growth of information technologies has rendered established techniques out of date. The cry of the hands-on technologist is the need to “maintain my skills.” Everyone wants to work on new development projects using the latest technology. As soon as a new technique catches on, the “old” skills are no longer desirable.

Only a few years ago, COBOL programmers were concerned about two aspects of their jobs: being relegated to maintenance tasks and not staying current with the latest technology. Today, these same people are being recalled from retirement with attractive salaries to perform maintenance on the old mission-critical systems. Skills in COBOL and the older information technologies enjoy a new-found respect that will last many years.

Our industry has grown so rapidly that we have not taken the time to step back and see what we have wrought. Individual commands focus on their system needs and develop interfaces with other systems for even more efficiencies. A similar approach to address Y2K is being taken. Each organization has looked at its system inventory and

(I hope) taken appropriate action. Only late in the game has it become apparent that the Y2K issues of our partners (suppliers of data as well as software) are as important as our internal problems. The worst-case scenarios of a global data meltdown are based on an understanding of the intricate, interlocking webs of applications that were built long before the Advanced Research Project Agency conceived the Internet. If any of these scenarios become real, we will fully realize for the first time how pervasive software has become.

Another subtle change in attitude centers on motivation. Nature’s primary motivators are desire and fear. As society exploited computational skills, management and practitioners were motivated mostly by desires—to beat the competition (or enemy), to code a more sophisticated routine, or to engineer another massive system. The motivator for the Y2K problem, however, is fear. For commands actively working on Y2K problems, management should be fully involved—failure is not an option. Fear of failure is more tangible now than it ever has been in this industry. The new motivator is another basis for maturation of our profession.

Conclusion

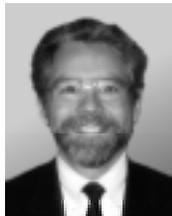
The Y2K issues and fears we face today are substantial. When we look back on the Y2K problem in 2015, we will see it as a blessing that we designed into an infant industry. Many more positives than those outlined here will be visible from the perspective of hindsight.

With an accurate inventory and improved regression test suite, programmers will be able to maintain systems more efficiently than in the past. As a result of the intense focus on Y2K projects, senior commanders and executives will have a better appreciation of the configuration management and quality assurance processes necessary to develop and support systems. With better processes, project managers can estimate costs and schedules more accurately and be more accountable for the results. Thus, a more mature industry

that has survived a dilemma of its own making may be seen as a valuable asset.

As information technology survives the millennium, the perception of our industry will move from that of an art to a science. As we struggle with our Y2K problems through the next few years, let us not lose sight of the many potentially beneficial aspects that improved processes will provide us. In the future, commanders and practitioners alike will be able to focus more on technical issues than process problems. ♦

About the Author



John B. Hubbs designed the Civilian Health and Medical Program for the Uniformed Services health-care claims processing system deployed with the 7th Army in Heidelberg, Germany. While supporting the Social Security Administration, he directed the contractor's CMM-based process improvements and lead the design for a

major development project that received Vice President Gore's National Performance Review Award in June 1997. He currently performs independent verification and validation for the Health Care Financing Administration.

AverStar, Inc.
3120 Timanus Lane
Baltimore, MD 21244
Voice: 410-944-3190
Fax: 410-944-3868
E-mail: jhubbs@averstar.com
Internet: <http://www.averstar.com>

Coming Events

Call for Papers: 1999 Acquisition Research Symposium (ARS)

Dates: June 21-23, 1999

Location: Doubletree Hotel, Rockville, Md.

Sponsor: Deputy Undersecretary of Defense for Acquisition Reform.

Co-hosts: Defense Systems Management College (DSMC) and the National Contract Management Association, Washington, D.C. chapter.

Theme: "Acquisition Reform – A Revolution in Business Affairs"

Topics: Acquisition Reform Successes and Lessons Learned, Business Process Reengineering and Benchmarking, Commercial Applications in Government, Competitive Acquisition Strategies, Cost and Resource Management, Federal Acquisition and the Political Process, Industrial Base/Civil/Military Integration, International Acquisition Issues, Leveraging Technology in Acquisition, Management Decision and Information Support Tools, Organization and Cultural Change, Outsourcing and Privatization.

Three camera-ready copies of paper due: Feb. 26, 1999 (obtain guidelines from Joan L. Sable)

Submit to: Joan L. Sable, DSMC Program Chairwoman, ARS 99, 9820 Belvoir Road, Suite 3, Fort Belvoir, VA 22060-5565

Voice: 703-805-5406 DSN 655-5406

E-mail: ars99@dsmc.dsm.mil

"Making the Move to COCOMO II" Seminars

Dates: March, May, and September 1999

Locations: Los Angeles, Calif., San Francisco, Calif., Dallas, Texas, Chicago, Ill., Boston, Mass., Washington, D.C., depending on date.

Instructor: Donald J. Reifer

Voice: 310-530-4493

Fax: 310-530-4297

E-mail: info@reifer.com

Internet: <http://www.reifer.com>

1999 IEEE Aerospace Conference

Dates: March 6-13, 1999

Location: Snowmass at Aspen, Colo.

Sponsor: Aerospace and Electronics Systems Society of the Institute of Electrical and Electronics Engineers (IEEE).

Topic: The internationally attended IEEE Aerospace Conference is organized to promote interdisciplinary understanding of aerospace systems, their underlying science and technology, and their applications to government and commercial endeavors.

Contact: Mike Johnson or Beth Leitereg

Voice: 702-784-6485

Fax: 702-787-1342

E-mail: jmj@ee.unr.edu

Registration: registration@aeroconf.org

Internet: <http://www.aeroconf.org>

Call for Papers and Workshops: Ninth International Conference on Software Quality

Dates: Oct. 4-6, 1999

Location: Royal Sonesta Hotel, Cambridge, Mass.

Sponsor: American Society for Quality Software Division

Suggested Topics: Software Quality Management, Software Processes, Software Project Management, Software Metrics, Measurement and Analytical Methods, Software Inspection, Testing, Verification and Validation, Software Audits, Software Configuration Management, Standards, Quality Philosophies and Principles, Organizational and Interpersonal Techniques, Problem-Solving Tools and Processes, and Professional Conduct and Ethics.

Abstracts, proposals for workshops, and panel sessions due: March 8, 1999

Contact: John Pustaver, Conference Chairman, 4 Kendall Road, Sudbury, MA 01776

E-mail: pustaver@swquality.com