From: Jones, Capers <Capers@SPR.com>
To: 'Bob Bemer' <bbemer@bmrsoftware.com>
Date: Monday, June 15, 1998 11:57 AM
Subject: RE: Hidden, Encoded, and Encrypted Dates

Bob,

Thanks for your reply - the situation is more or less as I had imagined.

Here is a question from Congress that goes beyond my personal knowledge. Are you aware of any systems whose dates default to 1961? I've been told that possibly some old Wang or Burroughs systems might do this, and that octal numbering systems may be the reason.

I worked much of my career at IBM, and have little personal knowledge of other manufacturer's equipment other than recent stuff.

Thanks,
Capers JOnes

-----Original Message-----
From: Bob Bemer [mailto:bbemer@bmrsoftware.com]
Sent: Monday, June 15, 1998 1:06 PM
To: Jones, Capers
Subject: Re: Hidden, Encoded, and Encrypted Dates

Man, that was some talk! It frightened even me.
Very, very glad to have it, though. Super useful
data, untouched by Koskinen's dirty hands.
To reply to your query in a satisfactory, or at
least lengthy, way took an attachment.
-----Original Message-----
From: Jones, Capers <Capers@SPR.com>
To: 'bbemer@bmrsoftware.com' <bbemer@bmrsoftware.com>
Date: Friday, June 12, 1998 1:44 PM
Subject: Hidden, Encoded, and Encrypted Dates

>Bob,
>
>Last week I introduced a new talk on year 2000 contingency planning at
>the Data Dimensions conference in Texas. There are three forms of
dates
>that can be present in the MVS world and I'm not sure how you would
deal
>with them. The three are hidden dates, encrypted dates, and encoded
>dates where a date field is filled with other information, such as

>strings of 9's.
>
>How do you deal with these three date topics? I can see how you might
>handle explicit dates and perhaps indirect dates, but I'm uncertain as
>to how you would deal with something like an encrypted date.
> <<Y2KCont.ppt>>
>Best Regards,
>Capers Jones
>

Reply to Jun 12 Questions

You asked how we would/will handle hidden, encrypted and encoded dates -- assuming, quite correctly, that our method can handle explicit and indirect dates.

The answer is "as poorly and/or painstakingly as everyone else, with the burden on the customer". Our difference is that by using computer power we get most of the work done in very short time, leaving enough time to work on these conundrums and enigmas.

We don't do miracles, we don't read the original programmer's mind, and we don't heal bad code.

If by hidden date you mean something like incorporating it in an insurance policy number, like "BA6970321F08", then the question arises on whether you are going to expand the policy number fields like you would expand plain date fields. If you chose that path, you had better damn well make sure that ALL of your historical data is so changed, and send the customer a new copy of his policy with the new identifier, or else havoc. I don't think one wants to change these.

If by encoded date you mean something like flag dates or just plain illegal values for a date, I extract the following from our Website viability document:

- o When derailed for a decimal compare operation, check if either operand has a candidate value that one might expect a programmer to use to represent "infinity", or an unlikely date in the future. E.g. "991231" or "123199", or some other value known to or suspected by the user of this product. If it does appear, it's count is augmented. If such appears often, it's not a computed value but rather a literal (which you could have located in the source program as well). During the derail we can bigitize that to be equivalent to "20991231" or "12312099".
  A newcomer to the VERTEX 2000 (TM) method might ask "How then can a real date of "991231" be handled? Simple! When we make that change, and restore the changed test value, the test process is turned off!

\CALENS.TXT

From: Jones, Capers <Capers@SPR.com>
To: 'Bob Bemer' <bbemer@bmrsoftware.com>
Date: Wednesday, April 08, 1998 12:30 PM
Subject: RE: Answers to your doubts

Bob,

Thanks for your kind remarks about my last paper. Here is yet another, and this one also mentions your work. Are you now in the commercial market or still testing your product? p.12

Best Regards,
Capers Jones

-----Original Message-----
From: Bob Bemer [mailto:bbemer@bmrsoftware.com]
Sent: Wednesday, March 25, 1998 5:21 PM
To: gary.fisher@nist.gov
Cc: capers@SPR.com; Roleigh.Martin-1@tc.umn.edu; yourdon@cutter.com; tompetz@msn.com; jbemer@ford.com
Subject: Answers to your doubts

The attached covers the two points that you made. My taking an international position does not mean that I would ignore US opinion. Thanks to Capers Jones for a great paper, and I'd like to be able to work with him toward its goals. Be sure all of you see it. Really an eye-opener.

Bob Bemer
bbemer@bmrsoftware.com
http://www.bmrsoftware.com

THE EURO, THE YEAR 2000, AND AND THE SOFTWARE PERSONNEL SHORTAGE

April 8, 1998

Abstract

Software is a highly labor-intensive occupation that is currently entering into a period of significant labor shortage. The current software population of the United States is approximately 2,000,000 if software engineers, managers, and support personnel are included. The estimated shortfall of software engineering personnel in the United States has been placed between 100,000 and more than 300,000 jobs.

However, the year 2000 software problem and the Euro-currency conversion problem are both turning out to be much larger than anticipated, and are running late. If both of these massive software projects are included as well as the normal backlog of work that cannot be performed until personnel are available, the total shortfall can exceed 700,000 jobs. Since there is no reasonable way of filling 700,000 jobs, the only likely solution is to defer substantial volumes of normal software development and maintenance work.

Capers Jones, Chairman
Software Productivity Research, Inc.
1 New England Executive Park
Burlington, MA 01803-5005

Phone	781 273-0140 X-102
FAX	781 273-5176
Email	Capers@SPR.com
CompuServe	75430,231
Web	http://www.spr.com

INTRODUCTION

Software engineering is a highly labor-intensive occupation. In spite of the claims of various software technologies such as the object-oriented (OO) paradigm, computer aided software engineering (CASE), component-based development, and many others, software still requires very significant amounts of manual labor.

As computers and software continue to expand their roles in business, government, and military affairs the increase in demand for software has triggered a substantial shortage of skilled software personnel. Adding to the rising demand for software personnel are too additional factors that are making the shortage much more severe than it might otherwise be: the year 2000 software problem, and the unified European currency or Euro.

The Euro is scheduled to begin circulation on January 1, 1999 and all software applications that deal with currencies are scheduled to be ready by the end of 2002. Work on the year 2000 software problem has already started but many companies are far behind the curve of necessary accomplishment. Therefore date problems are starting to occur with software applications that look forward (such as credit card processing) and other problems will occur too.

The combined impact of the year 2000 and the Euro on top of the already tight labor market for software engineering personnel is creating significant problems for the United States and the industrialized nations of the world.

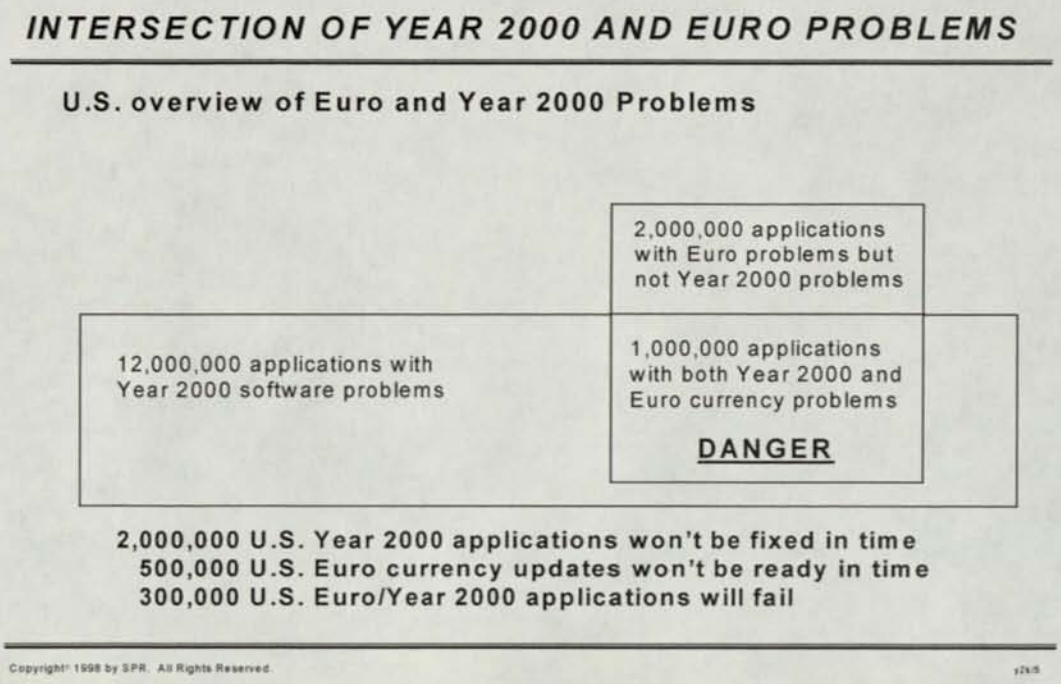
The Sizes of the Year 2000 and Euro Problems in the United States

There is considerable debate about the relative sizes of the software updates associated with the year 2000 and Euro currency problems. Overall the year 2000 problem is far larger and affects many more kinds of software, such as realtime and embedded applications. However within certain industries such as banks and retail chains, the Euro currency problem is at least as large as the year 2000 in terms of the numbers of software applications which require modification.

Based on the author's past analysis of the sizes of these two massive problems, it appears that for the United States about 12,000,000 applications must be updated to repair the year 2000 problem and about 3,000,000 applications must be updated to deal with the Euro. Both of these estimates are partly speculative and have a large margin of error.

The most serious concern is the set of applications that need to be repaired for both the year 2000 and the Euro simultaneously. These applications are at serious risk due to the fact that the two sets of changes may interfere with each other. Also a problem called "bad fix injection" needs to be considered. Each time an application is updated, there is a small but significant chance that the update may contain a new error. The U.S. average is about a 7% bad-fix injection rate. When concurrent updates are taking place, such as with the Euro and the year 2000, it may raise the bad fix injection rate above 15%.

The following illustration gives an approximate overview of the magnitude of these two massive problems for the United States.



Software Occupation Groups and Specialization

In 1994 and 1995 the author's company, Software Productivity Research (SPR), was commissioned by AT&T to carry out a study of software specialization within large software organizations. Some of the enterprises that participated included AT&T itself, the U.S. Air Force, Texas Instruments, and IBM. These enterprises were visited and participated in detailed surveys. Some of the results of this study were published in the author's book, "Patterns of Software Systems Failure and Success" (Jones 1995). See also the author's more recent book, "The Year 2000 Software Problem – Quantifying the Costs and Assessing the Consequences" (Jones 1997).

The study revealed several facts that were suspected, but difficult to pin down. First, the software industry like every other profession is now moving toward a wide variety of specialists. Second, the existence of so many different occupations and specialties within the overall software community makes demographic studies very difficult. For example, many of the technical personnel who write embedded software are electrical or mechanical engineers by training, and some refuse to call themselves programmers or software engineers even though their work is actually that of creating software. Third, personnel records are so ambiguous that it is not easy to determine who is involved in software and who is not in typical companies and government organizations.

Table 1: Software Specialties Within the United States in 1998

Software Occupation Groups	Number Employed	Percent
Programmer/analyst	400,000	19.67%
Programmer (maintenance)	350,000	17.21%
Programmer (development)	275,000	13.52%
Project manager (1st level)	225,000	11.06%
Software engineer (systems)	200,000	9.83%
Testing specialist	125,000	6.15%
Systems analyst	100,000	4.92%
Software engineer (realtime)	75,000	3.69%
Software technical writer	75,000	3.69%
Software engineer (embedded)	70,000	3.44%
Data administration specialist	50,000	2.46%
Project manager (2nd level)	35,000	1.72%
Software Quality Assurance	25,000	1.23%
Configuration control specialist	15,000	0.74%
Performance specialists	7,500	0.37%
Project manager (3rd level)	5,000	0.25%
Software Architect	1,500	0.07%
SUBTOTAL	2,034,000	100.00%

Support Occupations	Number Employed	Percent
Software sales specialist	105,000	30.04%
Customer support specialist	80,000	22.89%
Systems administration	50,000	14.31%
Software mgt. consultant	45,000	12.88%
Software education specialist	30,000	8.58%
Software librarians	15,000	4.29%
Process auditors/assessors	7,500	2.15%
Process improvement specialist	5,000	1.43%
Measurement specialist	3,500	1.00%
Software marketing specialist	3,000	0.86%
Cost estimating specialist	2,000	0.57%
Cost estimating specialist	2,000	0.57%
Human factors specialist	1,000	0.29%
Certified function point counter	500	0.14%
SUBTOTAL	349,500	100.00%
TOTAL	2,383,500	100.00%

A full analysis of all possible software specialties would total to more than 50 occupations. For this article, only the more common occupations will be considered. The results illustrated in Table 1 are derived from extrapolating the populations of our

clients up to national levels. This means that the data in Table 1 has a large margin of error, but no other source of data includes all of the more important kinds of specialties.

Table 1 shows two sets of occupations. The first set are the personnel who are involved in the creation and maintenance of software projects. The second set are the support personnel who do not build or maintain software, but who handle sales of software packages, customer support, process improvement, and other indirect activities. The percentages in Table 1 are against each of the two sets separately, rather than against the combined total.

When all of the major specialties and occupations are included, the U.S. total of software employment is much larger than if only "programmers" or "software engineers" are included.

U.S. Personnel Shortages Due to the Year 2000, the Euro, and Normal Backlogs

As software applications expanded in number and utility, there has long been a "backlog" of applications awaiting development or maintenance but unable to start due to the lack of personnel.

These backlogs are somewhat ambiguous and difficult to enumerate, but a general rule of thumb is that a normal corporate software backlog exceeds the supply of available software personnel by between 10% and perhaps 25% based on company and industry variations. That is, staffing would have to be between 10% and 25% larger than it actually is if all applications awaiting development were to be worked on simultaneously. In extreme cases, the backlog can actually approach 100% which would imply needing twice the available personnel.

The year 2000 problem and the Euro currency problem have added major unplanned elements to normal backlogs. Neither the year 2000 nor the Euro were included in long-range budget forecasts for most U.S. enterprises until about 1994 or 1995. Both the year 2000 and the Euro are extremely labor intensive, and both have deadlines which cannot be slipped without grave peril (although both will probably be missed by many companies).

As of 1998 it is obvious that both the year 2000 and the Euro are running significantly behind schedule, due to the fact that many companies delayed and did not begin work until late 1997 or even early 1998. Indeed, some companies have still not started on either year 2000 repairs or on Euro updates.

As a result, it is no longer possible to envision completing either the year 2000 repairs or the Euro repairs as a "business as usual" activity. It is unlikely that either of these massive updates can now be completed in time.

A rough analysis of the additional software personnel that might be required can indicate how improbable it now is to assume 100% completion of either the year 2000 or the

Euro. Table 2 illustrates the approximate numbers of additional personnel needed to try and solve the year 2000 and Euro problems in the time remaining, plus normal backlogs.

Table 2: Projected Software Shortages in the United States in 1998

Software Occupation Groups	Number Employed	Year 2000 Shortage	Euro curr. Shortage	Backlog Shortage	TOTAL SHORTAGE
Programmer/analyst	400,000	40,000	32,000	40,000	112,000
Programmer (maintenance)	350,000	42,000	35,000	17,500	94,500
Programmer (development)	275,000	27,500	33,000	19,250	79,750
Project manager (1st level)	225,000	27,000	15,750	22,500	65,250
Software engineer (systems)	200,000	40,000	0	12,000	52,000
Testing specialist	125,000	43,750	12,500	12,500	68,750
Systems analyst	100,000	3,000	12,000	10,000	25,000
Software engineer (realtime)	75,000	18,750	0	7,500	26,250
Software technical writer	75,000	3,000	7,500	3,750	14,250
Software engineer (embedded)	70,000	24,500	0	8,400	32,900
Data administration specialist	50,000	6,000	6,000	5,000	17,000
Project manager (2nd level)	35,000	3,500	3,500	1,750	8,750
Software Quality Assurance	25,000	5,000	3,750	1,250	10,000
Configuration control specialist	15,000	7,500	5,250	1,050	13,800
Performance specialists	7,500	1,125	7,350	375	8,850
Project manager (3rd level)	5,000	400	400	250	1,050
Software Architect	1,500	75	120	120	315
SUBTOTAL	2,034,000	293,100	174,120	163,195	630,415

Support Occupations	Number Employed	Year 2000 Shortage	Euro Shortage	Backlog Shortage	TOTAL SHORTAGE
Software sales specialist	105,000	12,600	12,600	5,250	30,450
Customer support specialist	80,000	10,400	9,600	4,000	24,000
Systems administration	50,000	10,000	2,500	2,500	15,000
Software mgt. consultant	45,000	9,000	4,500	2,250	15,750
Software education specialist	30,000	0	0	1,800	1,800
Software librarian	15,000	1,800	1,500	750	4,050
Process auditors/assessors	7,500	4,500	1,500	1,000	7,000
Process improvement specialist	5,000	0	0	750	750
Measurement specialist	3,500	0	0	1,225	1,225
Software marketing specialist	3,000	0	0	300	300
Cost estimating specialist	2,000	600	500	400	1,500
Human factors specialist	1,000	0	0	120	120
Certified function point counter	500	50	50	100	200
SUBTOTAL	322,000	48,950	32,750	20,445	102,145
TOTAL	2,356,000	342,050	206,870	183,640	732,560

Both the year 2000 and the Euro problem are special situations that demand very high levels of change management, high levels of quality control, and extensive testing. These special needs mean that the shortage of available software personnel is much more concentrated for some occupations than for others.

Note that Table 2 assumes a mix of date field expansion, windowing, and bridging as the repair strategy for both the year 2000 and the Euro. For the year 2000 problem it is now too late to use date field expansion alone, so windowing and bridging will dominate.

Occupation Groups With The Greatest Shortages of Software Personnel

Updating software to make year 2000 repairs and add Euro-currency support is a highly technical kind of work which needs very elaborate change control, extensive testing, and careful quality control.

Some occupations such as technical writing and education will not be severely impacted by either the year 2000 or the Euro currency work. The occupations with the greatest shortages for the year 2000 and the Euro include:

• Performance specialists	118% increased demand
• Process assessors/auditors	93% increased demand
• Configuration control specialists	92% increased demand
• Cost estimating specialists	75% increased demand
• Testing specialists	55% increased demand
• Embedded software engineers	47% increased demand
• Quality assurance specialists	40% increased demand
• Function point counters	40% increased demand
• Realtime software engineers	35% increased demand
• Management consultants	35% increased demand
• Data administration specialists	34% increased demand
• Customer support specialists	30% increased demand
• Systems administration specialists	30% increased demand

It is fairly obvious that neither the United States nor any other country is turning out software specialists in sufficient quantities to meet the demands of the high-growth occupations. Indeed, it is not always possible even to meet the demands of the low-growth rate occupations.

In software as in all other technical professions, a great deal of specialization is starting to occur. In the context of the year 2000 and the Euro, the specialists that deal with data administration, with realtime and embedded software, with change management, and with testing and quality are obviously going to be in sharp demand. Table 3 shows the occupations in descending order of anticipated needs:

Table 3: Software Occupation Shortages Ranked In Order of Need

Software Occupation Groups	TOTAL SHORTAGE	CURRENT STAFF	SHORTAGE PERCENT
Performance specialists	8,850	7,500	118.00%
Configuration control specialist	13,800	15,000	92.00%
Testing specialist	68,750	125,000	55.00%
Software engineer (embedded)	32,900	70,000	47.00%
Software Quality Assurance	10,000	25,000	40.00%
Software engineer (realtime)	26,250	75,000	35.00%
Data administration specialist	17,000	50,000	34.00%
Programmer (development)	79,750	275,000	29.00%
Project manager (1st level)	65,250	225,000	29.00%
Progammer/analyst	112,000	400,000	28.00%
Programmer (maintenance)	94,500	350,000	27.00%
Software engineer (systems)	52,000	200,000	26.00%
Systems analyst	25,000	100,000	25.00%
Project manager (2nd level)	8,750	35,000	25.00%
Project manager (3rd level)	1,050	5,000	21.00%
Software Architect	315	1,500	21.00%
Software technical writer	14,250	75,000	19.00%
SUBTOTAL	630,415	2,034,000	30.99%
Support Occupations	**TOTAL SHORTAGE**	**CURRENT STAFF**	**SHORTAGE PERCENT**
Process auditors/assessors	7,000	7,500	93.33%
Cost estimating specialist	1,500	2,000	75.00%
Certified function point counter	200	500	40.00%
Measurement specialist	1,225	3,500	35.00%
Software mgt. consultant	15,750	45,000	35.00%
Customer support specialist	24,000	80,000	30.00%
Systems administration	15,000	50,000	30.00%
Software sales specialist	30,450	105,000	29.00%
Software librarian	4,050	15,000	27.00%
Process improvement specialist	750	5,000	15.00%
Human factors specialist	120	1,000	12.00%
Software marketing specialist	300	3,000	10.00%
Software education specialist	1,800	30,000	6.00%
SUBTOTAL	102,145	347,500	29.39%
TOTAL	732,560	2,381,500	30.76%

The anticipated personnel growth for reducing the normal backlog, for solving the year 2000 problem, and for dealing with the Euro each individually seems to exceed the available capacity of the software labor pool. The need to deal with all of these factors

concurrently is probably not going to be possible given current rates of personnel entering software or already available.

Trying to Balance Software Work Demands With Limited Personnel Availability

Under the combined impact of the year 2000, the Euro, and normal backlogs there will be more software work than there are available personnel until well into the next century. Given the fact that neither the Euro nor the year 2000 repairs will be finished in time, we can expect substantial effort in the first five years of the next century to be devoted to emergency repairs and disaster recovery tasks.

Software executives and managers will be faced with continued shortages of personnel for an indefinite time. Therefore it is appropriate to consider some of the possible strategies for dealing with the situation. These strategies are listed in alphabetical order and not in order of preference or recommendation.

Acquiring software companies: A strategy that is available for the larger and more profitable companies is to acquire small software consulting companies or even commercial software groups primarily to gain access to their software personnel. This strategy is fraught with risk, since there is no guarantee that the new software personnel will stay unless compensation and work are both satisfactory.

Automating the development cycle: There are thousands of software tools available and many of these are quite good. Unfortunately, many of these are not very good also. Tools for year 2000 search and repair activities, configuration control, defect tracking, and other labor-intensive tasks can be helpful. However, tools alone are not usually sufficient. Also, tools usually have a learning curve that needs to be climbed to gain full proficiency.

Bonuses and salary premiums: Given the shortage of software personnel in critical specialty areas, many companies are now paying bonuses and premium salaries for software employees. This strategy is only available to companies with adequate cash flow and profit margins. It is not a suitable strategy for the marginally funded or minimally endowed. It is also not a strategy available to government organizations. What is starting to occur is that companies with substantial financial reserves are able to absorb the better software personnel from companies with marginal financial resources, some of whom may not be able to maintain staffing levels enough to continue to build software of their own.

Canceling Non-Critical Projects: The ruthless suppression of all projects that are hard to maintain, technically obsolete, or troublesome in any way is a strategy whose time is fast approaching. Of course some of these applications perform useful functions, and it may be necessary to eliminate some functions that have value if the software is shut down. As major software applications are shut down, expect corporate downsizing and layoffs to follow suit. The 1999 – 2001 time period may well see a significant number of bankruptcies, layoffs, and downsizings due to the Euro and year 2000 events.

Contracting and/or outsourcing: When full-time personnel are overworked, the most common solution is to bring in contractors or outsource selected work. This strategy has a long history and is reasonably successful. About 70% of the time both parties seem satisfied. However, lawsuits and litigation do occur in the software contract and outsource domains, so it is not a perfect solution even though it is a fairly good solution. The costs of outsourcing for year 2000 and Euro work are often about twice those of in-house repairs, so this strategy may add costs to these two already expensive software problems.

Commercial Off-the-Shelf Software (COTS): Given the on-rushing problems with the Euro and the Year 2000, custom applications may be approaching luxury status. Many companies are trying to eliminate custom applications in favor of commercial packages. These packages may not be as flexible and complete as custom software, but if they are year 2000 compliant and easy to install and use, this may be a good solution.

Cooperating with competitors: Due to the seriousness and urgency of the year 2000 and Euro problems, the author has recommended that the U.S. government suspend anti-trust regulations for year 2000 and Euro work for a five-year period. Not only should anti-trust regulations be suspended for these serious problems, but key industries should be encouraged to pool their available software resources. Larger companies that are nearing year 2000 and Euro completion should assist smaller companies that are lagging and far behind on the ground that failure of a critical number of enterprises will damage the overall economy. Key industries include airlines, automotive, banks, defense, health care, insurance, public utilities, and telecommunications as examples.

Enterprise Software Packages: Some companies are considering adopting enterprise wide software packages such as SAP or Oracle. These packages are both year 2000 compliant and should be able to handle the Euro, so on the face of it this is a possibly useful option. The down side of the enterprise package solution is the long time span to deploy these tools and calibrate them for local use. It is almost too late to get this class of tool installed before the year 2000 unless most of the defaults are accepted without customization. The normal time period for deployment of full enterprise packages is in the 18 to 24 month zone, so this solution is nearing the danger point for both the year 2000 and Euro problems.

Hiring retired software workers: The software industry is still fairly young, but due to the wave of downsizings and layoffs of the late 1980's and early 1990's more than 100,000 software employees in the U.S. took early retirement packages or have retired at more usual ages such as employees leaving after age 60. These retired employees represent an interesting pool of talent that is starting to be tapped for year 2000 repair work. This solution is particularly useful in urban areas that have been experiencing layoffs and downsizing waves before the economy kicked into high gear.

Importing software workers of "insourcing": Since the number of software personnel in the United States is not sufficient to handle the year 2000 and Euro problems plus

normal backlogs, one possible solution is the importation of software workers from other countries for temporary periods. This method has been used for some years with small numbers of personnel. But with so little time remaining before the year 2000 and Euro problems explode on the world it is unlikely for this method to provide a solution. Also, the numbers of software workers needed is large enough do denude the total software populations of several countries, leaving them with insufficient resources of software personnel for their own needs.

Improving software processes: The topic of software process improvement is now a major theme of software conferences and journals. Unfortunately, software process improvement is usually a multi-year undertaking with substantial costs involved. There is probably not enough time remaining for software process improvement to have much impact before the year 2000 and the Euro problems pass their respective deadlines.

Insuring against software business problems: The role of insurance against various forms of business risk is an important aspect of the modern world. Insurance against the year 2000 problem is a complex issue. On one hand, some insurance companies have excluded year 2000 damages from their director and officer liability policies, and from other forms of risk insurance too. However, a number of insurance companies would be willing to offer year 2000 risk insurance to qualified organizations that can successfully pass a pre-insurance audit which demonstrates that their year 2000 work was at state of the art levels. Insurance against damages associated with Euro-currency software failures is more problematic.

Laws declaring the year 2000 problem an "act of God": Several states have enacted statutes declaring that the year 2000 problem is an "act of God." As this is written, such statutes have already been passed in Nevada and Georgia, and are being considered by the legislatures of Florida, California, and many other states as well. The purpose of the statutes is to minimize the risks that state governments will themselves be sued for their probable failure to fix the year 2000 problem. As so often happens with hasty legislation, the results will probably cause more harm than good. For one thing, how can state governments sue their own software vendors for failing to achieve year 2000 compliance when such a statute is on the books? For another, the presence of the statutes is tending to relax the already lax state responses to the year 2000 problem. Basically the statutes are simply raising the risks that medicaid, welfare, and other social benefits can be disrupted with impunity.

Offshore contracts and outsourcing: There are large pools of software personnel in many developing countries such as China, India, Russia, Thailand, and the Ukraine. Often these software personnel are well trained and very capable. Also, Eastern Europe is not likely to have as many year 2000 problems as Western Europe, nor as many Euro problems. But offshore outsourcing still involves large distances, uncertain security and legal protection of software assets, and some risk of the engagement going sour with no easy way to recover. Even so, the rate at which off-shore outsourcing is increasing is now moving to significant levels. India is a frequent choice for U.S. companies since

knowledge of spoken English is good and software quality control can sometimes be superior to U.S. norms.

Overtime and multiple shifts: The traditional way of dealing with schedule crises is resort to massive quantities of overtime, much of it unpaid. This solution is no longer effective in a world where software jobs outnumber available software personnel by perhaps 30%. Too much overtime pressure on software employees today will lead to the resignation of the software staff, leaving the company in very serious trouble.

Masking year 2000 and Euro problems: For the year 2000 problem the basic repair strategy is to expand date fields from two digits to four digits. Unfortunately for a large company with both software applications and data bases to repair, this strategy can take more than 36 calendar months. Therefore in the time remaining alternative strategies such as windowing and bridging are now being used since these can often be accomplished in less than 22 months. Of course windowing is only a temporary solution. For Euro currency updates similar variants are also occurring. Eventually these temporary masking approaches will need permanent solutions.

Mergers and acquisitions: Surprisingly some companies are considering mergers and acquisitions mainly because one of the pair is already year 2000 compliant, can support the Euro, or both. If the two companies are in similar businesses or can share software applications, this may be an interesting option.

Object-code date interception: It is theoretically possible to analyze software while it is executing and intercept some or perhaps all of the dates on the fly. This is the basis of a patent filed by software researcher Bob Bemer. Currently this method operates only on IBM mainframe software. Its efficiency in finding indirect or calculated dates is uncertain, but trials with explicit dates are encouraging. This technology is of both technical and practical interest if it works and can be expanded to other platforms.

Retraining other technical professionals: While software engineering and associated specialties are entering a labor shortage, mechanical and electrical engineering are not at full capacity. It is theoretically possible to retrain many technical workers into software engineers. The problems with this approach are not clearly visible, but need to be considered. First, retraining requires a fairly large curriculum and can take from six months to more than 18 months. After retraining, productivity and quality levels will not be optimal for another year or so. Therefore the effective performance of retrained personnel may not immediately match the levels of experienced software personnel. However, retrained personnel often have quite a few years of seniority and are making fairly large salaries. Also, many technical workers chose their current professions and don't want to become software engineers.

Reuse of Software Artifacts: Since both year 2000 repairs and Euro updates are basically similar problems, it should be possible to encapsulate the solutions for these two problems into a fairly small number of reusable artifacts. Only about a dozen variants of dates and currency routines have been noted. Of course with 500 programming

languages and dialects to deal with, the number of reusable artifacts can reach 12,000 (12 Euro and 12 year 2000 methods multiplied by 500 languages). In an ideal world, these reusable artifacts would be made available for free or for no more than actual cost by the U.S. government and by major software vendors. The year 2000 and Euro problems would be an excellent opportunity to elevate software reuse from a theoretical topic to a major business tool. However, to date the power of software reuse has not yet appeared in the Euro and year 2000 domains.

Twenty-four hour a day year 2000 and Euro repair centers: Since very little time remains for either year 2000 repairs or Euro updates, one possible solution would be to move from a one-shift 8 hour a day repair strategy to a three-shift 24 hour a day repair strategy. It is unlikely that three-shift operations could occur in the United States because software personnel willing to work the second and third shifts would be unavailable. However for large multi-national companies it is possible to have three software repair centers located eight time zones apart, such as facility in the United States, one in Eastern Europe, and one in the Pacific Rim. This way repairs could take place during the first shift in each location, and transfer to the East at the close of every shift. Although this strategy is immediately aimed at the year 2000 and Euro problems, it would give any large software outsource company a competitive edge for normal software work too.

None of these solutions are perfect, but given the serious consequences associated with both the Euro and the year 2000 all solutions need to be evaluated and considered. The combination of several solutions is also possible.

Summary and Conclusions

The year 2000 and Euro currency problems are the two largest business problems in human history. The fact that they are occurring so close together is extremely unfortunate. As a result of the juxtaposition of these two massive problems, neither is likely to be completed on schedule.

The year 2000 and the Euro are more than just software issues. Both can affect business and daily life in significant ways. The combination of the Euro and year 2000 problems will change modern business operations in ways that are hard to predict.

Software and computers are the key issues for dealing with both the Euro and the year 2000. Unfortunately, in 1998 software remains a craft that requires a very large amount of manual labor. The software labor shortage associated with the combination of the year 2000 and the Euro will dominate the software industry from 1999 through at least 2003, and will slow down or eliminate many other kinds of software work.

References and Readings on the Year 2000 and the Euro

Jones, Capers; Patterns of Software System Failure and Success; International Thomson Computer Press, Boston, MA; December 1995; 250 pages; ISBN 1-850-32804-8; 292 pages.

Jones, Capers; Table of Programming Languages and Levels (8 Versions from 1985 through July 1996); 67 pages for Version 8; Software Productivity Research, Inc., Burlington, MA; 1996.

Jones, Capers; The Year 2000 Software Problem - Quantifying the Costs and Assessing the Consequences; Addison Wesley, Reading, MA; 1998; ISBN 0-201-30964-5; 303 pages.

Jones, Capers; Software Quality – Analysis and Guidelines for Success; International Thomson Computer Press, Boston, MA; ISBN 1-85032-876-6; 1997; 492 pages.

Jones, Capers; "The Economics of Object-Oriented Software"; SPR Technical Report; Software Productivity Research, Burlington, MA; April 1997; 22 pages.

Jones, Capers; "Becoming Best in Class"; SPR Technical Report; Software Productivity Research, Burlington, MA; January 1998; 40 pages.

Jones, Capers; "The Costs, Schedule, and Value of Software Process Improvement"; SPR Technical Report; Software Productivity Research, Inc.; Burlington, MA, January 1998; 27 pages.

Jones, Capers; "Estimating Rules of Thumb for the Year 2000 and Euro-Currency Projects"; SPR Technical Report; Software Productivity Research, Inc.; Burlington, MA; January 1998.

Jones, Capers and Huffman, Jan; Resource Conflicts Between the Euro and the Year 2000; SPR Technical Report; Software Productivity Research, Inc.; Burlington, MA January 1998.

Jones, Keith; Year 2000 Software Crisis Solutions; International Thomson Computer Press, 1997.

Kappelman, Leon (editor); Solving the Year 2000 Problem; International Thomson Computer Press, 1997.

Kurtzman, Dr. Clifford; "Frequently Asked Questions about the Year 2000 and Similar Problems;" Tenagra Corporation, Houston, TX; December 1997; http://www.tenagra.com or http://www.year2000.com.

Lefkon, Dr. Dick (editor); Year 2000 Best Practices for Y2K Millennium Computing: Panic in Year Zero; Mainframe Special Interest Group (SIG) of the Association of Information Technologies (AITP); New York, NY.

Murray, Jerome T. and Murray, Marilyn M.; The Year 2000 Computing Crisis - A Millennium Date Conversion Plan; McGraw Hill, New York, NY; 1996.

Ragland, Bryce; The Year 2000 Problem Solver; McGraw Hill, New York, NY; 1997.

Robbins, Brian and Rubin, Dr. Howard; The Year 2000 Planning Guide; Rubin Systems, Inc.; Pound Ridge, NY; 1997.

Rubin, Dr. Howard; Survey of Year 2000 Preparations in Fortune 500 Companies; Meta Group, Stamford, CT; January 1998.

Rubin, Dr. Howard et al; "Software Productivity and Quality Issues and the United States IT Workforce Shortage; Rubin Systems, Inc.; Pound Ridge, NY; 1997.

Strassmann, Paul; Bender Richard, et al; Standard for Year 2000 Test Certification; Software Testing Assurance Corporation; Stamford, CT; 1998.

Ulrich, William and Ian S. Hayes; The Year 2000 Software Crisis - Challenge of the Century; Prentice Hall, Yourdon Press; 1997.

Yourdon, Ed and Yourdon, Jennifer; Time Bomb 2000: What the Year 2000 Computer Crisis Means to You; Prentice Hall PTR, Upper Saddle River, New Jersey; 1998.

RESOURCE CONFLICTS BETWEEN THE YEAR 2000 AND EURO-CURRENCY SOFTWARE PROBLEMS

Version 2

December 13, 1997

Abstract

At a national level, the year 2000 problem affects far more industries and is far more serious than the Euro-currency conversion problem. The year 2000 problem will affect more than 50% of the current software applications throughout the world. By contrast, the Euro-currency problem will only affect the 15% or so of software applications that deal with cash and currency exchange rates. But for a significant number of companies and industries, both the year 2000 problem and the Euro-currency problem are occurring simultaneously. Banks, financial institutions, retail and wholesale enterprises, and taxation agencies are faced with the impossible demands of attempting to modify software for Euro-currency conversion by January 1, 1999 and fix the year 2000 problem by January 1, 2000.

The year 2000 problem cannot be avoided and the date cannot be pushed back. The Euro-currency problem, on the other hand, it purely arbitrary and the date can be pushed back at least five years with no serious consequences. Since no company nor country has enough resources to do both upgrades, it would be prudent for the European community to move the date for Euro-currency conversion from 2002 to 2005 in order to eliminate conflicts with the much more serious year 2000 problem.

Capers Jones, Chairman
Jan Huffman
Software Productivity Research, Inc.
1 New England Executive Park
Burlington, MA 01803-5005

Phone	781 273-0140 X-102
FAX	781 273-5176
Email	Capers@spr.com
CompuServe	75430,231

RESOURCE CONFLICTS BETWEEN THE YEAR 2000 AND EURO-CURRENCY CONVERSION PROBLEMS

INTRODUCTION

As much of the world knows, the end of the century is causing a major software crisis because of the need to update millions of software applications and change the date fields from two-digit to four-digit form. Dates stored in two-digit form will cause software applications to stop or produce incorrect results at the end of the century, since the year 99 followed by the year 00 will through off many calculations.

Date problems are already starting to occur in software, and almost every major corporation and government agency is deploying thousands of personnel and millions of dollars to attempt to achieve year 2000 compliance before the end of the century.

Another problem of similar magnitude will occur between January 1, 1999 and January 1, 2002. In what must be regarded as one of the most unwise and hazardous public policy decisions in all of human history, the European Community has scheduled the completion of conversion to a unified currency for the calendar year 2002, without any apparent regard or concern for the fact that the year 2000 problem and the Euro-Currency conversion will be in direct competition for scarce software resources.

Of course it would be an exciting thing to start the new century with a new currency, so you can see why the European Union wanted to begin the currency change before the 21st century. Unfortunately, the glamour of the standard currency blinded the political leaders to the painful fact that Europe does not have enough software personnel to do both the currency conversion and year 2000 repairs at the same time.

The net result of the unwise conversion schedule set by the European Union is that the Western European countries have been devoting most of their available surplus software resources to the currency conversion problem, and are not yet expending even minimal resources on the year 2000 problem.

A much wiser policy would have been to defer the European currency conversion until after the year 2000 problem, and not plan for full conversion until 2003 at the earliest or perhaps even 2005, which should be after the bulk of the year 2000 problem is behind us.

By treaty, any delay in the date of January 1, 1999 for freezing the exchange rates among participating European currencies must be announced before the end of 1997. With powerful monetary policy issues at stake, it is doubtful that this date will change. However, the date of January 1, 2002 for the issuance of bank notes and coinage in Euro denominations and its attendant need for commerce to be transacted in Euro monetary units can be more practically delayed.

Meanwhile, public policy can mitigate the impact of Euro currency on the information technology community in both the public and private sectors. Governments, for example,

can continue to accept statistical reports and tax returns in local currency until the Euro-denominated bank notes are issued. They can likewise delay requiring posting of prices in Euro units until shortly before this time.

When the implications of the Euro currency problem are viewed in the light of stock markets, bond markets, commodities, and derivatives the world is facing what looks right now like total chaos and major disruptions of all forms of investment trading, due in large part to the exceptionally bad timing of the Euro-currency start up. European financial markets should continue to report in the same currencies as they do now in order to give their partners time to support Euro currency after they have dealt with year 2000 repairs.

The date for year 2000 repairs is fixed and cannot be changed. The date for Euro-currency conversion is purely political and arbitrary and could be pushed back at the stroke of a pen, as indeed it should be.

To illustrate why the year 2000 problem is a far more serious issue, table 1 shows some of the kinds of software applications that are likely to be affected by the year 2000 problem, the Euro-currency problem, or both:

Table 1: Key Applications Impacted by Year 2000 and Euro-Currency Problems

Year 2000 Problem	**Euro-Currency Problems**
Retail sales	Retail sales
Banking applications	Banking applications
Taxation and finance	Taxation and finance
Accounting and billing	Accounting and billing
Telephone switching	No impact
Electricity generation	No impact
Air traffic control	No impact
Military applications	No impact

Note that a significant number of software applications are affected by both the year 2000 problem and the Euro-currency problem simultaneously. This is a very hazardous situation, since any reasonable estimate of the work required to do either update indicates that available software resources will be severely strained. It is clearly impossible to achieve both year 2000 compliance and Euro-currency compliance at the same time.

In Europe itself, the European currency conversion impact on software applications is as large as the impact of the year 2000 problem in many industries, and in two industries, banking and retailing, will probably be even more expensive.

Some businesses, particularly those in the financial sector, are preparing now to handle Euro currency to take advantage of market opportunities. Others, such as stock traders,

have to be ready to use Euro currency by 1999, as reporting in Euro units is scheduled to begin then.

The Euro-currency situation only affects software dealing with finance, and is not a major factor in weapons systems, telephone switching, aircraft flight control, and some of the other areas where the year 2000 problem may cause safety hazards.

Further, the impact of failing to achieve the European currency conversion target date will only result in financial and political problems, even if they might be severe ones. In fact, the major result of not achieving the target date for Euro-currency conversion is to reveal how little politicians know about technical issues, and how dangerous their ignorance can be.

Failing to achieve year 2000 conversion, on the other hand, can result in much more serious problems involving possible disruption of air traffic, shut down of power and telecommunication facilities, and potential injuries or death due to failure of medical instruments, airline navigation equipment, or railroad control devices.

European Damages From Simultaneous Year 2000 and Euro-Currency Updates

The most probable result of the unwise policy of having the schedule for the European currency conversion overlap the schedule for the year 2000 problem will be the following:

Through 1997 and early 1998 the Western European countries will continue to devote most of their software efforts to the European currency problem, and pay little or no attention to the year 2000 problem. In early 1998, the Western European countries will realize the far more serious hazards associated with the year 2000 problem and begin emergency repairs. Since software resources are limited in Europe as they are in the United States, some of the resources for year 2000 repairs will probably be diverted from the currency conversion projects.

Since there are not enough software resources available in Western Europe to complete both the year 2000 repairs and the currency conversion work on parallel schedules, the dilution of effort between these two enormous problems will probably result in these three disturbing phenomena:

- Western Europe will not achieve year 2000 compliance for even 65% of the applications that need year 2000 repairs.

- The simultaneous attempts to perform year 2000 repairs and currency conversion work on nearly identical schedules will cause both efforts to slip well beyond their deadlines. Western Europe will probably not be able to complete the currency conversion work until about 2005, and will probably not be fully recovered from the year 2000 crisis until at least 2003.

- Since the most convenient pools of surplus software personnel reside in Eastern Europe, the Eastern European countries of Russia, the Ukraine, the Czech Republic and others will probably end up with very significant outsource contracts, and will grow to become major software powers while Western Europe may face losses and reductions in software staffing after the two crises have passed.

While the European currency conversion problem is a major topic in Western Europe in 1997, many U.S. companies are not yet even dealing with it although some U.S. companies will be affected too when the new currency becomes operable.

Essentially the U.S. software organizations have been ignoring the Euro-currency issue and concentrating on the year 2000, while their counterparts in Western Europe have been ignoring the year 2000 problem and concentrating on the Euro-currency conversion.

It would not be surprising to find that the unwise scheduling of the European currency conversion to begin in 1999, only one year before the year 2000 and leap year problems hit, will end up by damaging the Western European economies severely, while perhaps strengthening at least the software balance of trade for Eastern Europe and the Pacific Rim, and perhaps South America as well.

The following table shows a rough comparison of the two major software undertakings for the 15 countries in the European Union. The Euro-currency effort is merely set at 90% of the year 2000 effort, on the grounds that fewer applications (15%) will be affected, but that the cost of modifying them will be greater. These assumptions are both arbitrary and questionable, and reflect only a preliminary judgment. Table 2 is taken from my book The Year 2000 Software Problem - Quantifying the Costs and Assessing the Consequences (Addison Wesley Longman, 1998).

Table 2: Comparison of European Currency Conversion and Year 2000 Effort

	Software Staffs	Portfolio Size in Funct. Pts.	Year 2000 Effort (Months)	Currency Effort (Months)	Total Effort: European Union
Germany	550,000	440,000,000	2,612,500	2,351,250	4,963,750
United Kingdom	390,000	312,000,000	1,852,500	1,667,250	3,519,750
France	385,000	308,000,000	1,828,750	1,645,875	3,474,625
Italy	375,000	290,625,000	1,725,586	1,553,027	3,278,613
Spain	235,000	170,375,000	1,011,602	910,442	1,922,044
Netherlands	100,000	77,500,000	460,158	414,142	874,300
Belgium	65,000	50,375,000	299,102	269,192	568,294
Portugal	65,000	45,500,000	270,156	243,140	513,296
Greece	60,000	45,000,000	249,600	224,640	474,240
Sweden	60,000	45,000,000	267,188	240,469	507,657
Austria	51,000	39,525,000	237,150	213,435	450,585
Denmark	34,000	27,200,000	159,800	143,820	303,620
Finland	32,000	24,800,000	147,200	132,480	279,680
Ireland	24,000	19,200,000	111,600	100,440	212,040
Luxembourg	3,000	2,325,000	13,710	12,339	26,049

TOTAL	2,429,000	1,897,425,000	11,246,602	10,121,942	21,368,544
COSTS			$94,471,456,800	$85,024,311,120	$179,495,767,920

If this table is close to reality, then roughly eight months of programming effort on the part of every software professional in the European Union may be devoted to year 2000 or Euro-currency work over the next two and a half years. The inescapable conclusion must be that the European Union will fall far behind other geographic regions in terms of the ability to build new applications and become a major software vendor in world markets.

Although the table does not show the information, the costs for damages, litigation, and recovery of unrepaired software applications in the European Union should be much larger than almost any other geographic region, since Western Europe has lagged in year 2000 repairs:

- Expect about $4.00 in year 2000 damages for every $1.00 on year 2000 repairs, on the grounds that the European Union will probably not be much more than 65% compliant at the end of the century.

- For most organizations, losses due to failure to completely support Euro currency will be limited to those due to miscalculations, manual calculation efforts, lost business opportunities, and government sanctions. Euro currency support failures should not endanger people or property, and litigation should be minimal, although organizations giving financial advice could find themselves at significant risk.

The most likely beneficiary of the European Union software problems will probably be Eastern Europe, since a surplus of trained software personnel are available which will not be preempted by either the year 2000 or Euro-conversion problems.

It is quite unfortunate that the European Union is moving into a very hazardous first decade of the next century. The timing of the European currency conversion could hardly have been worse, and illustrates the basic human problem of "tunnel vision" or failing to understand how one topic connects to other topics.

The Euro-Currency Impact on U.S. Corporations and Government Agencies

The immediate impact of the Euro-currency problem in the United States is significant, but far less severe than the impact in Western Europe. In Europe, all cash and credit card transactions in restaurants and retail establishments, as well railroads, airlines, bus companies, etc. need to handle transactions in both local currency and the new Euro-currency.

For United States corporations, the European branches of multi-nationals will be affected, and organizations whose software handles currencies, cash, and exchange rates will need

modification. Also, governmental software which deals that imports and exports and international taxation will require modification. Table 3 compares the probable costs.

Table 3: Estimate of United States Year 2000 and Euro-Currency Costs

Expense Elements	Year 2000 Software Repair Costs	Euro-Currency Support Costs
Initial software repairs	$70,000,000,000	$8,750,000,000
Secondary bad fixes	$7,000,000,000	$875,000,000
Test library repairs	$10,000,000,000	$300,000,000
Database repairs	$60,000,000,000	$600,000,000
Hardware chip replacement	$10,000,000,000	$0
Hardware performance upgrades	$20,000,000,000	$500,000,000
Subtotal	$177,000,000,000	$11,025,000,000
Litigation and legal damages	$100,000,000,000	$2,500,000,000
Post-event damages	$116,075,625,000	$11,607,562,500
Post-event recovery	$281,168,750,000	$33,740,250,000
Subtotal	$497,244,375,000	$47,847,812,500
TOTAL	$674,244,375,000	$58,872,812,500

Although the probable Euro-currency conversion costs for the United States are only a small fraction of U.S. costs for the far more serious year 2000 repairs, the Euro-currency work can be troublesome for multi-national corporations. For example, the Euro-currency conversion will be troublesome for multi-national retail groups such as McDonald's, Colonel Sanders, and other retail establishments that have a significant presence in Western Europe.

Also severely impacted will be stock brokerages and the software which support stock analysis over long periods. When European stocks switch over from national currencies to the Euro, long range analysis will require an abrupt need for currency conversion logic, which is yet another expensive kinds of software upgrade.

Since the January 1, 2002 deadline for Euro-currency conversion will probably not be met even in Europe, and since it is a purely arbitrary date which can be pushed out if necessary, U.S. companies should request that President Clinton and the U.S. government ask the European Union to delay the Euro-currency deadline until 2005, so that it does not interfere with the far more serious year 2000 repair efforts.

Summary and Conclusions

The world's software community is facing two major problems that are on parallel schedules: the massive year 2000 problem and the smaller but still enormous Euro-currency conversion problem.

The schedule for the year 2000 repairs is absolutely fixed and cannot be pushed back. The schedule for the Euro-currency conversion is totally artificial and arbitrary, and can be pushed back very easily.

Because the year 2000 problem is potentially serious enough to cause a recession or depression, it should receive top priority on a global basis. Unfortunately, Western Europe has been devoting far too much effort to the Euro-currency conversion work and far too little to year 2000 repairs.

The European Union should wake up to the fact that Western Europe cannot do both sets of changes at the same time. Since the year 2000 problem cannot be slipped back, but the currency conversion date can be reset to any convenient date, the most prudent course for Western Europe would be to move the Euro-currency date from January 1, 2002 to January 1, 2005.

As the situation now stands, it is highly unlikely that the European Union will be able to achieve either year 2000 compliance or currency conversion compliance with their available software resources. It is folly to assume that both of these massive efforts can take place at the same time.

Software Productivity Research

Contingency Planning:

Damage Control for Euro and Year 2000 Software Problems

Capers Jones
Software Productivity Research

One New England Executive Park
Burlington, Massachusetts 01803
781.273.0140 Fax: 781.273.5176
www.spr.com

Why Contingency Planning?

Software defect repairs and updates are never 100% efficient.

- **Best case for the United States:**
  - **15% of year 2000 problems not fixed in time**
  - **20% of Euro updates not ready in time**

- **Best case for the European Monetary Union:**
  - **25% of year 2000 problems not fixed in time**
  - **10% of Euro updates not ready in time**

- **Best case everywhere else:**
  - **35% of both year 2000 and Euro problems not fixed in time**

Other Dates Needing Contingency Plans

January 1, 1999	Beginning of the Euro problems
August 21, 1999	GPS calendar rollover problem
September 9, 1999	"9's" end of file problem
January 1, 2000	Year 2000 problem
February 29, 2000	Year 2000 leap year problem
September 8, 2001	"9's" end of file problem for UNIX
Circa 2010	Dow Jones 5-digit problem
Circa 2015	Telephone number problem
January 19, 2038	UNIX and C rollover problem
Circa 2050	Social security number problem

Basic Components of Contingency Planning

- Your own Universe
  - Software
  - Hardware
  - Data Bases
  - Test libraries
  - Infrastructure
  - Litigation
  - Reinfection
- Your clients
  - Incoming information from them to you
  - Outgoing information from your to them
  - Your ability to ship goods
- Your suppliers, vendors, and government agencies
  - Their ability to stay in business
  - Their ability to service your needs
  - Their own year 2000 compliance

Cautions About Contingency Plan Liabilities

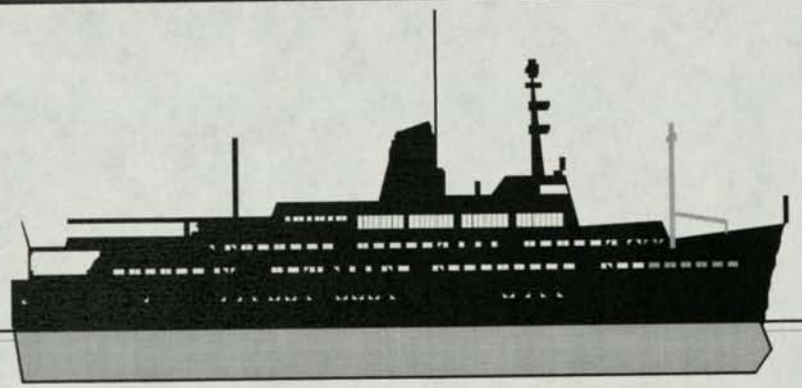
- CONTINGENCY PLANS MAY HAVE LEGAL IMPLICATIONS
- CHECK WITH YOUR ATTORNEY ABOUT LIABILITY ISSUES
- CONTINGENCY PLANS ASSUME FAILURES AND PROBLEMS
- PUBLISHING INFORMATION ABOUT FAILURES AND PROBLEMS MAY IMPLY INADEQUATE PERFORMANCE, EVEN THOUGH SOFTWARE FAILURES AND PROBLEMS HAVE OCCURRED CONTINUOUSLY FOR MORE THAN 50 YEARS
- CONTINGENCY PLANS MAY BE USED AGAINST YOU IN A COURT OF LAW.

Year 2000 Versus Euro Contingency Plans

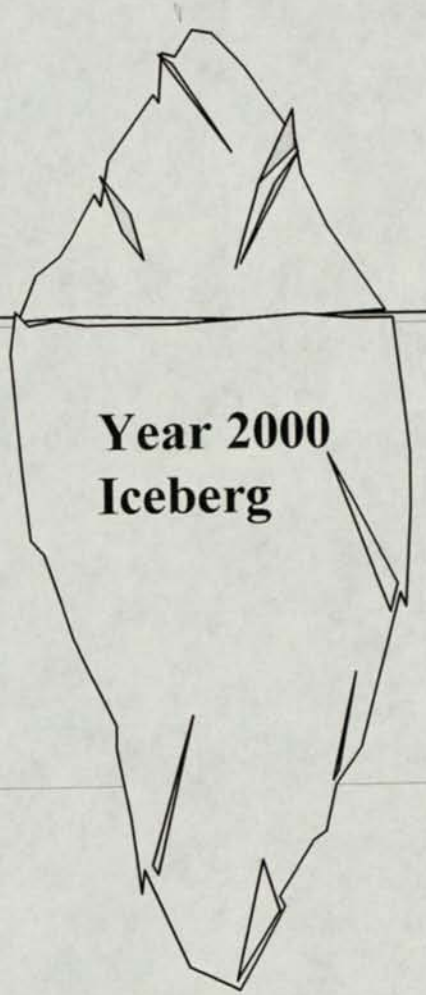
EURO CONTINGENCY PLANNING LAGS YEAR 2000 PLANNING

- Politicians can't admit to Euro problems without admitting that the problems were created by politicians. Therefore the damages the Euro might cause have not been estimated.
- Because governments are not dealing with the Euro dangers, few studies of potential problems have been carried out.
- Software failures due to the Euro can be as numerous as failures due to the year 2000.
- Liability and litigation from Euro software problems can be serious and expensive.
- The costs and timing of Euro damages and recovery work are basically unknown circa 1998.

THE EURO STEAMS TOWARD THE YEAR 2000



The Euro Titanic

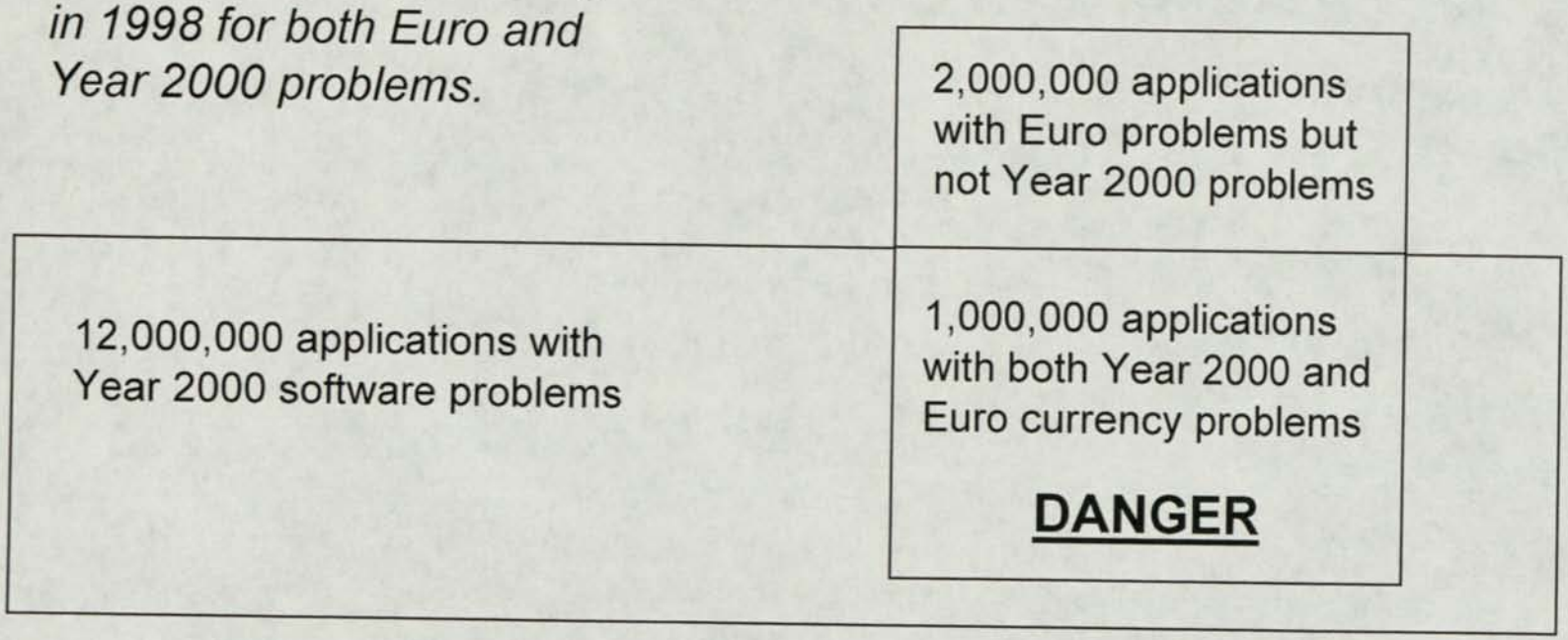


Year 2000
Iceberg

INTERSECTION OF YEAR 2000 AND EURO PROBLEMS

U.S. overview of Euro and Year 2000 Problems

Start contingency planning in 1998 for both Euro and Year 2000 problems.

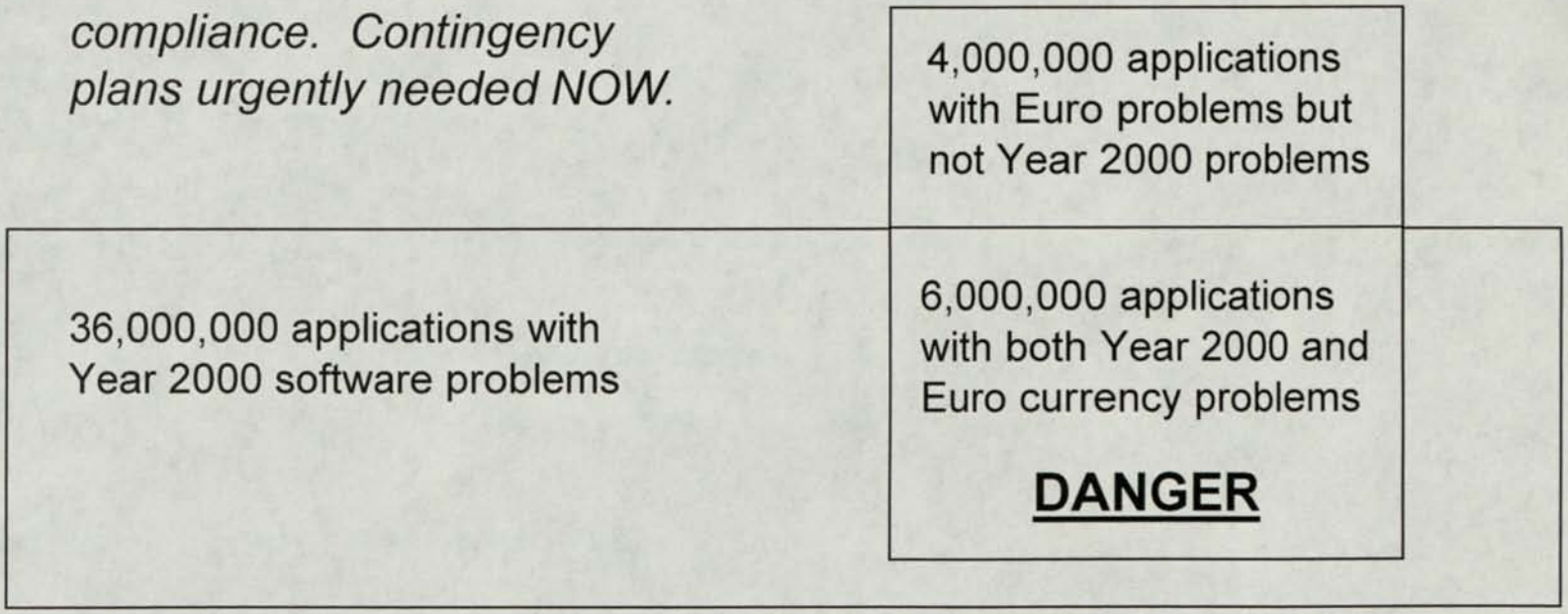


2,000,000 U.S. Year 2000 applications won't be fixed in time
500,000 U.S. Euro currency updates won't be ready in time
300,000 U.S. Euro/Year 2000 applications will fail

INTERSECTION OF YEAR 2000 AND EURO PROBLEMS

World-wide overview of Euro and Year 2000 Problems

It is now too late for full compliance. Contingency plans urgently needed NOW.



8,000,000 Year 2000 applications won't be fixed in time
1,000,000 Euro currency updates won't be ready in time
2,000,000 Euro/Year 2000 applications will fail

YEAR 2000 CONTINGENCY PLANS

- The Year 2000 problem is surprising in its impact

- Year 2000 contingency plans must be wide-ranging:
  - Failure of electric power and public utilities
  - Failure of financial transactions
  - Failure of government agencies
  - Failure of physical devices
  - Failure of transportation by air, rail, and sea
  - Failure of telephone systems
  - Possible structural damage (I.e. frozen pipes)
  - Partial availability of staff (lack of transportation)
  - Significant litigation risks

EUROPEAN CURRENCY CONTINGENCY PLANS

- The Euro introduction is planned from 1999 through 2002

- The Euro introduction is very complex and has high risks for economic and financial damages

- Contingency plans are needed for software failures:
  - Financial transactions
  - Stock transactions
  - Litigation

The Euro: What Can Go Wrong?

CLASS OF PROBLEM	PROBABILITY
• Exchange rate errors	80%
• Failures of software applications	50%
– ATM controls	
– Payrolls	
– Tax calculations	
• Failures of data base operations	50%
– Data mining	
– On-line analytical processing (OLAP)	
• Disruption of international business	35%

UNITED STATES YEAR 2000 STATUS IN MID 1998

- Fortune 500 companies
  - 50 will be > 99% compliant
  - 100 will be > 90% compliant
  - 150 will be > 75% compliant
  - 100 will be > 50% compliant
  - 100 will be < 50% compliant
- 2,500 large corporations
  - 250 will be > 99% compliant
  - 500 will be > 90% compliant
  - 1,250 will be < 90% compliant
- 25,000 small-medium corporations
  - 500 will be > 99% compliant
  - 2,500 will be > 90% compliant
  - 22,000 will be < 90% compliant

UNITED STATES YEAR 2000 STATUS IN MID 1998

- **500 Federal Agencies and Military Bases**
  - 25 will be > 99% compliant
  - 75 will be > 90% compliant
  - 150 will be > 75% compliant
  - 100 will be > 50% compliant
  - 150 will be < 50% compliant
- **50 State Governments**
  - 5 will be > 99% compliant
  - 15 will be > 90% compliant
  - 30 will be < 90% compliant
- **5,000 Urban Governments**
  - 250 will be > 99% compliant
  - 750 will be > 90% compliant
  - 4,000 will be < 90% compliant

UNITED STATES YEAR 2000 STATUS IN MID 1998

COMPLIANCE	ENTERPRISES	RISKS
> 99%	1,250	Minimal
> 90%	5,000	Moderate
> 75%	7,500	Significant
> 50%	10,000	Major
< 50%	12,500	Severe
TOTAL	36,250	Significant to Severe

Bottom line: More than three fourths of U.S. enterprises face significant to severe year 2000 problems.

YEAR 2000 DAMAGE POSSIBILITIES IN TEN INDUSTRIES

INDUSTRY	MAJOR YEAR 2000 HAZARD
Airlines	**Safety hazards**
Air traffic control chaos	**(15% probability)**
Flight schedule confusion	
Navigation equipment failures	
Maintenance schedules thrown off	
Possible Airline insurance cancellation	
Possible grounding of commercial Airlines	

CONTINGENCY PLANS

Must be prepared by FAA and airlines. Business contingency plans must assume flight cancellations/delays for month of January 2000 AD

YEAR 2000 DAMAGE POSSIBILITIES IN TEN INDUSTRIES (cont.)

INDUSTRY	MAJOR YEAR 2000 HAZARD
Defense	Security hazards
Base security compromised	(20% probability)
Computer security compromised	
Strategic weapons malfunction	
Command, communication network problems	
Aircraft maintenance records thrown off	
Logistics and supply systems thrown off	
Satellites may malfunction	

CONTINGENCY PLANS

Must be performed by military services. Civilian contingency plans unlikely.

YEAR 2000 DAMAGE POSSIBILITIES IN TEN INDUSTRIES (cont.)

INDUSTRY	MAJOR YEAR 2000 HAZARD
Finance	Financial transaction hazards
Interest calculations in error	(50% probability)
Account balances thrown off	
Credit card charges in error	
Funds transfer thrown off	
Mortgage/loan interest payments in error	
Possible "escheatment" of accounts	
Stock and bond trading may stop temporarily	
Stock and bond values may decline	

CONTINGENCY PLANS

Keep watch on financial institutions. Prepare reserve funds for January 2000.

YEAR 2000 DAMAGE POSSIBILITIES IN TEN INDUSTRIES (cont.)

INDUSTRY	MAJOR YEAR 2000 HAZARD
Health Care	Safety hazards
Patient monitoring devices malfunction	(40% probability)
Failure of complex devices (MRI, Cat scans)	
Medical instruments malfunction	
Possible shut-down of pharmaceutical labs	
Prescription refill problems	
Drug lot expiration date errors	
Billing errors due to year 2000	

CONTINGENCY PLANS

Must be performed by health providers. Check prescription refill dates.

Significant risk of injuries and litigation. Possible malpractice lawsuits.

YEAR 2000 DAMAGE POSSIBILITIES IN TEN INDUSTRIES (cont.)

INDUSTRY	MAJOR YEAR 2000 HAZARD
Insurance	Liability, benefit hazards
Policy due dates in error	(50% probability)
Benefits and interest calculation errors	
Annuities miscalculated	
Payment records in error	
Insurers may refuse to pay year 2000 damage claims	
Insurers may refuse to cover directors and officers	

CONTINGENCY PLANS

May need to seek special year 2000 coverage. May face business interruptions without adequate insurance coverage. Also need to consider physical damages such as frozen pipes.

YEAR 2000 DAMAGE POSSIBILITIES IN TEN INDUSTRIES (cont.)

INDUSTRY	MAJOR YEAR 2000 HAZARD
Local Government	Local economic hazards
Taxes miscalculated	(85% probability)
Divorce, marriage records misdated	
Birth, death records misdated	
Bond ratings may be downgraded	
May need to raise taxes	
May experience tax revenue reductions	

CONTINGENCY PLANS

Should be published by government agencies in July of 1999. Businesses should include government agencies in their own contingency plans.

YEAR 2000 DAMAGE POSSIBILITIES IN TEN INDUSTRIES (cont.)

INDUSTRY	MAJOR YEAR 2000 HAZARD
Manufacturing	Operational hazards
Subcontract parts fail to arrive	(60% probability)
Just-in-time arrivals thrown off	
Assembly lines shut down	
Process control systems stop	
Aging of accounts receivable and cash flow	
Aging of accounts payable and cash flow	
Pension payments miscalculated	
Distribution of goods may stop temporarily	

CONTINGENCY PLANS

Should be published by all manufacturers by early 1999.

YEAR 2000 DAMAGE POSSIBILITIES IN TEN INDUSTRIES (cont.)

INDUSTRY	MAJOR YEAR 2000 HAZARD
National Governments	Citizen record hazards
IRS tax records in error	(70% probability)
Annuities and entitlements miscalculated (or lost)	
Pensions miscalculated (or lost)	
Disbursements miscalculated (or lost)	
Retirement benefits miscalculated (or lost)	
Possible Medicare errors	
Possible social security errors	

CONTINGENCY PLANS

Government agencies should publish contingency plans by mid 1999. Many will not. Businesses should include government failures in their own plans.

YEAR 2000 DAMAGE POSSIBILITIES IN TEN INDUSTRIES (cont.)

INDUSTRY	MAJOR YEAR 2000 HAZARD
Public Utilities	Safety hazards
Electric meters malfunction	(60% probability)
Gas meters malfunction	
Distribution of electric power thrown off	
Temporary loss of water pressure	
Possible loss of electric power in many communities	

CONTINGENCY PLANS

This is the most severe kind of year 2000 problem. Public utilities should publish contingency/recovery plans by 1Q99. Businesses should also include utility failures in their contingency plans. Anticipate loss of electricity, water, and gas for several days.

YEAR 2000 DAMAGE POSSIBILITIES IN TEN INDUSTRIES (cont.)

INDUSTRY	MAJOR YEAR 2000 HAZARD
Telecommunications	Service disruption hazards
Intercontinental switching disrupted	(35% probability)
Domestic call switching disrupted	
Billing records in error	
Possible loss of telephone services	

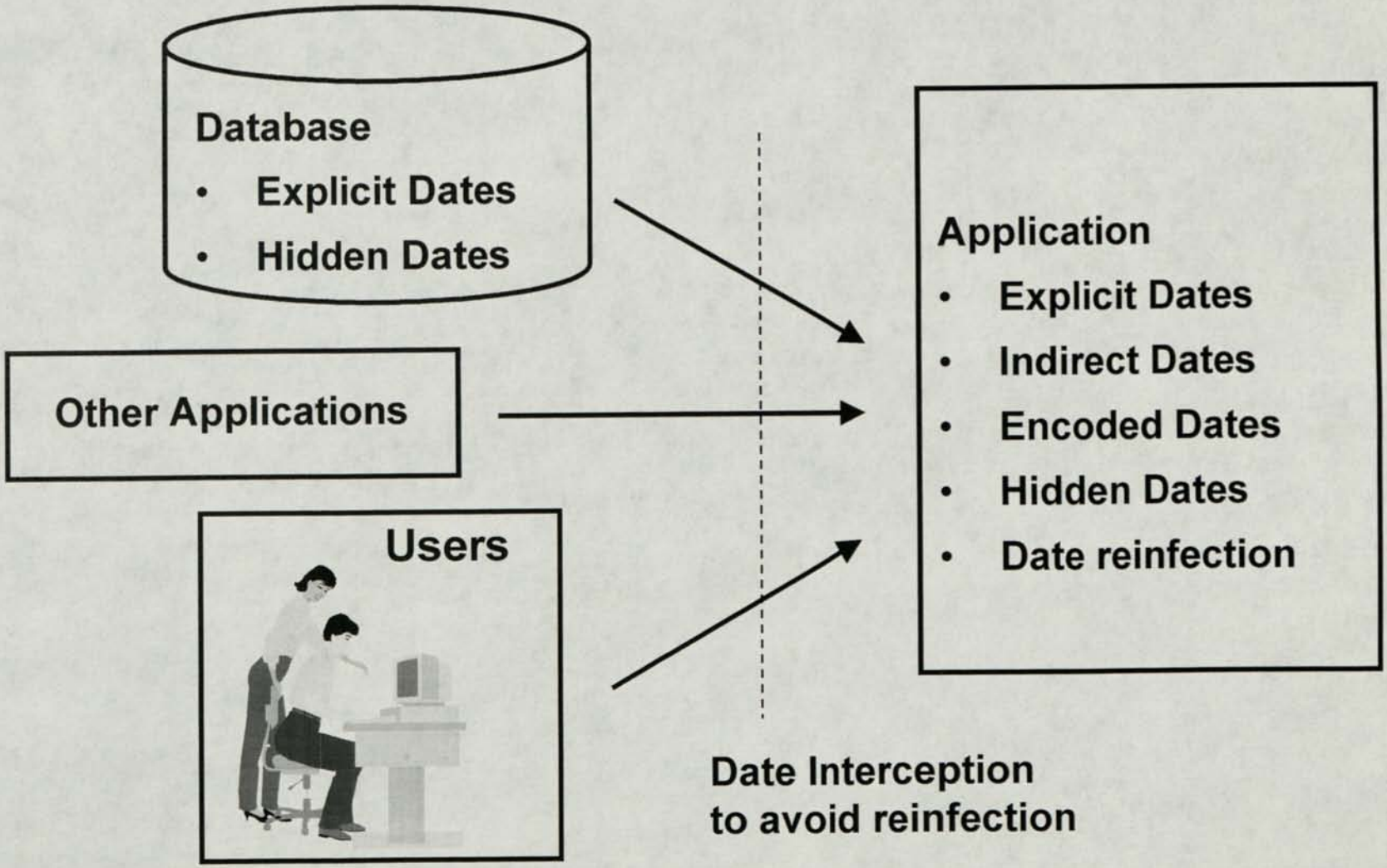
CONTINGENCY PLANS

Telcos should publish contingency plans by mid 1999. Businesses should include loss of telecom services in their own contingency plans. Anticipate at least partial loss of telephone services for several days.

Year 2000 Readiness in the United States

Industry	On-Time	Lagging	
Urban governments	10%	90%	Least prepared
Water companies	20%	80%	
Hospitals/Health care	20%	80%	
Retail stores	20%	80%	
State governments	30%	70%	
Electric power	30%	70%	
Federal government	30%	70%	
Manufacturing	35%	65%	
Telephone companies	65%	35%	
Airlines	60%	40%	
Insurance	75%	25%	
Banks	80%	20%	Best prepared
Overall U.S.	40%	60%	

SOURCES OF YEAR 2000 DATES



ENTERPRISE YEAR 2000 UNIVERSE

Databases
Repositories
Data Warehouses

External Applications

Customers
Employees
Suppliers
Governments

Active Applications		
Packages	In-house	Outsourced
Applications in Progress		
Packages	In-house	Outsourced
Applications Being Replaced		
Packages	In-house	Outsourced
Applications Without Dates		
Applications Retired/Archived		

Hardware and Embedded Applications

Test Libraries

Forms of Date Representation

FORMAT	EXAMPLE	CONTINGENCY
EXPLICIT	10 - 12 - 98 (US) 12 - 10 - 98 (Europe) 98 - 12- 10 (ISO) 1998-12-10 (ISO full) 98-280 (Julian)	99% find rate 3% bad fixes
INDIRECT	Hire Date + 12 months	80% find rate 10% bad fixes
HIDDEN	12398321	60% find rate 15% bad fixes
ENCODED	99 - 99 - 99	?
ENCRYPTED	ZABCIHCBA	?

Date Remediation Efficiency

FORM	DETECTION EFFICIENCY	FALSE DATES	BAD FIXES	PERF. LOSS
EXPLICIT	99%	2%	3%	- 5%
INDIRECT	80%	15%	10%	- 10%
HIDDEN	60%	30%	15%	- 15%
ENCODED	?%	?%	?%	?%
ENCRYPTED	?%	?%	?%	?%

Compliance Audits and Risk Assessment

The software industry has never achieved 100% defect removal.

The current average is about 85% defect removal efficiency.

Testing alone seldom tops 70% in defect removal efficiency.

Compliance audits will not guarantee 100% Year 2000 repairs.

Compliance audits will not guarantee 100% Euro repairs.

Compliance audits can validate that state of the art methods were used.

Compliance audits can validate that repair statistics are kept.

Compliance audits may be required for insurance coverage.

Year 2000 and Euro problems can occur even if audits are passed.

Reinfection can occur after a successful compliance audit.

PROBABLE NUMBER OF U.S. UNREPAIRED YEAR 2000 PROBLEMS

Software Types	Active Applications Containing Year 2000 Problems	Percent of Applications Repaired in Time	Percent of Applications Not Repaired in Time	Unrepaired Applications in January of 2000 AD
MIS	4,687,500	82.00%	18.00%	843,750
Systems	1,250,000	77.00%	23.00%	287,500
Commercial (PC)	2,500,000	90.00%	10.00%	250,000
Commercial (Mini)	1,250,000	95.00%	5.00%	62,500
Embedded	200,000	77.00%	23.00%	46,000
Commercial (Mainframe)	1,125,000	96.00%	4.00%	45,000
End-user	125,000	65.00%	35.00%	43,750
Military Logistics	210,000	94.00%	6.00%	12,600
Scientific	28,125	80.00%	20.00%	5,625
Military Weapons	30,000	96.00%	4.00%	1,200
Military CCC	15,000	97.00%	3.00%	450
Other	600,000	80.00%	20.00%	120,000
Sum/Average	12,020,625	85.75%	14.25%	1,718,375

Global Prospects of Successful Date Repairs

Software Problems	Applications with Problem	Percent Fixed in time	Unrepaired Applications	Year of Damage
Year 2000	36,000,000	80%	7,200,000	2000
Euro	10,000,000	75%	2,500,000	2002
UNIX rollover	12,000,000	90%	1,200,000	2038
End of file 9	4,000,000	90%	400,000	2001
Leap year	2,000,000	90%	200,000	2000
GPS rollover	250,000	98%	5,000	1999
TOTAL	64,250,000	87%	11,505,000	--

TRAIL OF YEAR 2000 EXPENSES

Year When 2000 Repairs Start	Software Expense Percent	Hardware Expense Percent	Litigation Expense Percent	Damages From Unrepaired Year 2000 Hits
1994	1%	0%	0%	0%
1995	3%	0%	0%	0%
1996	10%	1%	1%	0%
1997	15%	1%	2%	2%
1998	20%	5%	5%	3%
1999	27%	20%	10%	5%
2000	15%	25%	17%	50%
2001	7%	20%	24%	22%
2002	4%	15%	25%	10%
2003	0%	10%	10%	5%
2004	0%	2%	5%	2%
2005	0%	1%	2%	1%

IMPACT ANALYSIS OF MISSED YEAR 2000 REPAIRS

Percent Repaired	Percent Missed	Impact of Unrepaired Year 2000 Problems
100%	0%	No impact
90%	10%	Severe problems
85%	15%	Best case for United States
80%	20%	Possible recession
75%	25%	Best case for Western Europe
70%	30%	Possible depression
60%	40%	Economic chaos
50%	50%	Governments collapse

Impact Analysis of Missed Euro Currency Repairs

Percent Repaired	Percent Missed	Impact of Unrepaired Year 2000 Problems
100%	0%	No impact
90%	10%	Some EU Financial problems
85%	15%	Best case for Europe
80%	20%	Critical EU financial problems
75%	25%	Best case for United States
70%	30%	Possible EU depression
60%	40%	EU finance and political chaos
50%	50%	EU Governments collapse

Euro Readiness in European Monetary Union

Industry	On-Time	Lagging	
Urban governments	35%	65%	Least prepared
Telephone companies	45%	55%	
Payroll processing	45%	55%	
Restaurants/Food	55%	45%	
Provincial governments	55%	45%	
Retail stores	70%	30%	
National governments	75%	25%	
Stock exchanges	75%	25%	
Insurance	80%	20%	
Banks	90%	10%	Best prepared
Overall	60%	40%	

LITIGATION PROSPECTS OF YEAR 2000 PROBLEM

TEN FORMS OF YEAR 2000 AND EURO LITIGATION

1) Client lawsuits for financial damages
2) Shareholder lawsuits for stock losses
3) Shareholder lawsuits for violation of fiduciary duty
4) Damages for injuries or deaths caused by year 2000
5) Class-action suits by commercial software users
6) Suits against incompetent year 2000 or Euro vendors
7) Suits against hardware companies
8) Malpractice suits against corporate attorneys
9) Suits against government agencies
10) Suits against insurance companies

YEAR 2000 OCCUPATIONAL LIABILITIES

Most likely forms of litigation for failure to achieve full Year 2000 compliance

OCCUPATION GROUPS	PROBABLE LIABILITY
Chief Executive Officer (CEO)	Violation of Fiduciary Duty
Chief Operating Officer (COO)	Violation of Fiduciary Duty
Chief Financial Officer (CFO)	Violation of Fiduciary Duty
Chief Information Officer (CIO)	Violation of Fiduciary Duty
Boards of Directors	Violation of Fiduciary Duty
Corporate Attorney	Professional Malpractice
Director of Risk	Professional Malpractice
Year 2000 Consultant	Professional Malpractice
Year 2000 Tool Vendor	Misrepresentation
Year 2000 Outsourcer	Breach of Contract

YEAR 2000 EXECUTIVE RISK MATRIX

PROBABILITY	Executive Responses: Active 35%	Executive Responses: Passive 65%
Serious Damages 85%	30% chance of heading off major corporate law suits due to year 2000 problems	55% chance of law suit for violation of fiduciary duty
Minor Damages 15%	5% chance of law suit for misuse of funds	10% chance of slipping by with no serious problems and no major costs

NON-STANDARD METHODS FOR YEAR 2000 ACCELERATION

1) Suspend anti-trust regulations for year 2000 work

2) Suspend Euro-currency conversion until 2005

3) Suspend all other software except emergency repairs

4) Move to 24-hour around the clock year 2000 repairs

5) Establish year 2000 factories eight hours apart

6) Terminate hard-to-repair applications

7) Switch to "masking" instead of date field expansion

8) Experiment with object-code interception

9) Begin contingency planning NOW

10) Expect no help from the government

RANKINGS OF POTENTIAL YEAR 2000 COST ELEMENTS

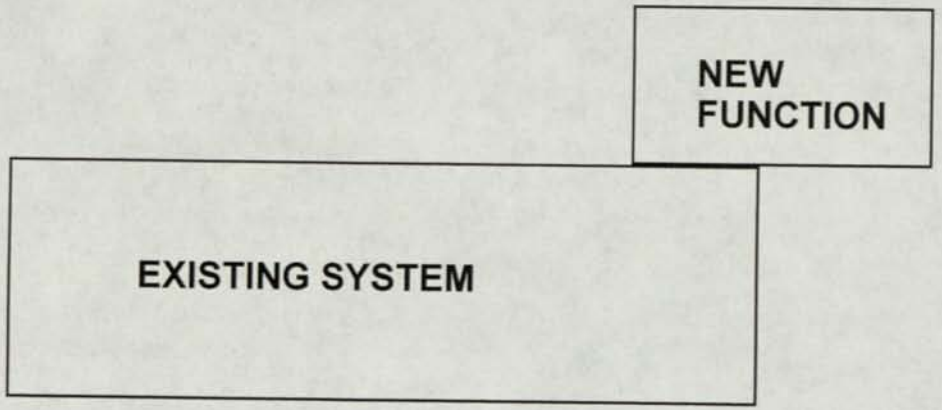
(Percentages and costs are relative to in-house date field expansions)

Expense Category	Percent	$/LOC	$/FP
Litigation: Year 2000 Deaths or Injuries	10,000%	$50.00	$5,000.00
Litigation: Year 2000 Damages	5,000%	$25.00	$2,500.00
Litigation: Shareholder	2,500%	$12.50	$1,250.00
Litigation: Professional Malpractice	750%	$3.75	$375.00
Litigation: Tax Errors	600%	$3.00	$300.00
Post-2000 Recovery Costs	325%	$1.95	$195.00
Database Date Field Expansion (outsourced)	300%	$1.35	$135.00
Post-2000 Damages (missed Year 2000 hits)	275%	$1.65	$165.00
Database Date Field Expansion (in-house)	250%	$1.13	$112.50
Software Date Field Expansion (outsourced)	250%	$1.25	$125.00
Date Field Compression	120%	$0.60	$60.00

RANKINGS OF POTENTIAL YEAR 2000 COST ELEMENTS (cont.)

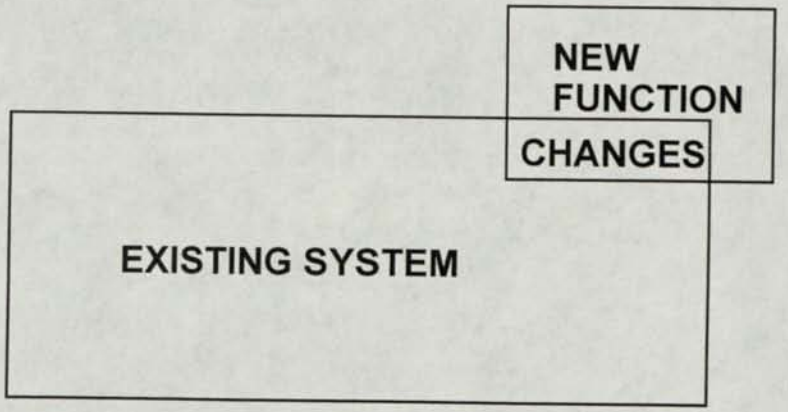
Expense Category	Percent	$/LOC	$/FP
Software Date Field Expansion (in-house)	100%	$0.50	$50.00
Software Date Field Expansion (off-shore)	90%	$0.45	$45.00
Bridging (database applications)	65%	$0.33	$32.50
Data Duplexing (batch applications)	40%	$0.20	$20.00
Encapsulation (date shifting)	35%	$0.18	$17.50
Regression Test Library Repairs (outsourced)	35%	$0.18	$17.50
Year 2000 Regression Testing	35%	$0.18	$17.50
Windowing (sliding)	30%	$0.15	$15.00
Year 2000 Performance Restoration	25%	$0.13	$12.50
Year 2000 Insurance	25%	$0.13	$12.50
Windowing (fixed)	20%	$0.10	$10.00
Regression Test Library Repairs (in-house)	20%	$0.10	$10.00
Year 2000 Date Searching (manual)	20%	$0.10	$10.00
Triage of Applications	10%	$0.05	$5.00
Year 2000 Date Searching (automated)	5%	$0.03	$2.50
Year 2000 Awareness Briefings	2%	$0.01	$1.00
SUM		$104.96	$10,600.96

TYPE 1 ENHANCEMENT: ADDING BLOCK FUNCTIONS



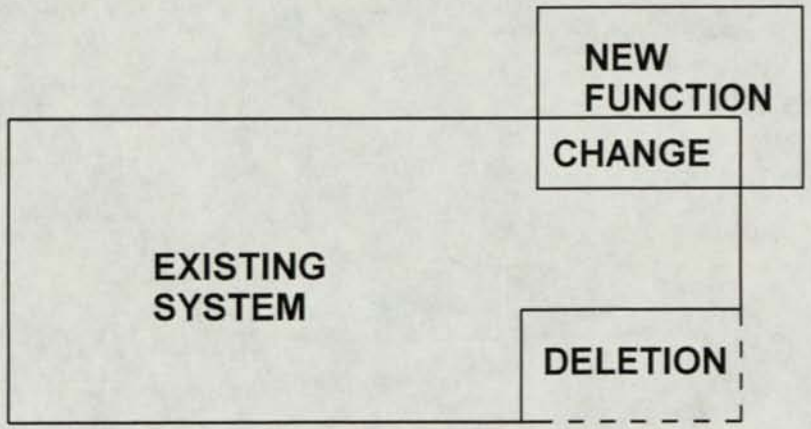
- Similar to Year 2000 windowing and encapsulation
- Possible only with well-structured systems
- Productivity close to that of new programs
- Productivity = 5 to 15 function points per staff month
- Repair efficiency = > 99%
- Bad fix injections = < 1%

TYPE 2 ENHANCEMENT: BLOCK FUNCTIONS WITH MODIFICATIONS



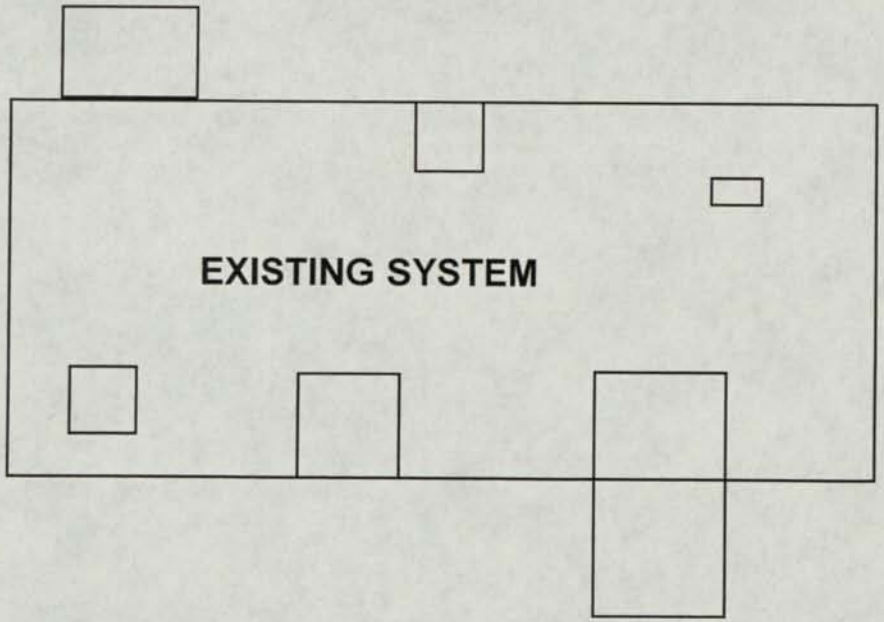
- Similar to bridging for Year 2000 repairs
- Best case for Euro currency repairs
- Probable only with well-structured systems
- Productivity lower than new programs
- Productivity = 3 to 10 function points per staff month
- Repair efficiency = > 95%
- Bad fix injections = < 3%

TYPE 3 ENHANCEMENT: BLOCK FUNCTIONS WITH MODIFICATIONS AND DELETIONS



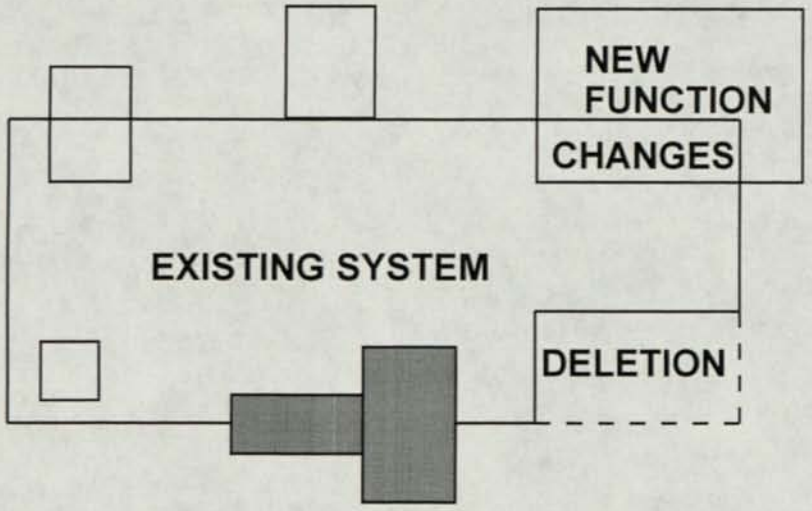
- Similar to data duplexing for Year 2000 repairs
- Probable only with well-structured systems
- Productivity lower than new programs
- Deletions are hard to measure
- Productivity = 2 to 8 function points per staff month
- Repair efficiency = > 90%
- Bad fix injections = < 7%

TYPE 4 ENHANCEMENT: SCATTER UPDATES



- Typical of legacy date fields expansion for Year 2000
- Typical of Euro conversion on aging applications
- Productivity much lower than new programs
- Productivity = 1 to 5 function points per staff month
- Repair efficiency = < 90%
- Bad fix injections = >10%

TYPE 5 ENHANCEMENT: HYBRID



- Typical of concurrent year 2000 and Euro updates
- Productivity much lower than new programs
- Schedules hard to predict: usually run late
- Very difficult to measure accurately
- Productivity = 0.5 to 3 function points per staff month
- Repair efficiency = < 65%
- Bad fix injections = > 15%

DEFECT SEVERITIES

CLASS	DEFINITION	COMMERCIAL	EURO	YEAR 2000
Severity 1	System stops or cannot be used	5%	15%	50%
Severity 2	System used with seriously reduced functions	20%	50%	40%
Severity 3	System used with minor restrictions	55%	30%	6%
Severity 4	System used with no functional restrictions	20%	5%	4%
	Total valid defects	100%	100%	100%

The Data Base and Data Mining Problems

Data bases contain both currency and date information

Data base updates are harder than software updates

Data base technology has no size or quality metrics

The Year 2000 and Euro data problems may have these impacts:

- Costs may by much higher than software repairs
- Data mining may stop or become unreliable
- Data errors affect salaries, pensions, and finances

The data base community is lagging software in compliance

Contingency plans for data are difficult

PROBABLE BEST-CASE COSTS FOR U.S UNREPAIRED YEAR 2000 PROBLEMS

Software Types	Unrepaired Applications in January of 2000 AD	Probable Damages per Application	Probable Recovery Costs per Application	Probable Damages and Recovery
MIS	843,750	$75,000	$150,000	$189,843,750,000
Systems	287,500	$125,000	$375,000	$143,750,000,000
Military Logistics	12,600	$500,000	$1,000,000	$18,900,000,000
Military Weapons	1,200	$2,000,000	$10,000,000	$14,400,000,000
Commercial (Mainframe)	45,000	$100,000	$200,000	$13,500,000,000
Military CCC	450	$2,500,000	$15,000,000	$7,875,000,000
Embedded	46,000	$10,000	$50,000	$2,760,000,000
Commercial (Mini)	62,5000	$10,000	$20,000	$1,875,000,000
Commercial (PC)	250,000	$2,500	$5,000	$1,875,000,000
Scientific	5,625	$20,000	$40,000	$337,500,000
End-User	43,750	$2,500	$5,000	$328,125,000
Other	120,000	$5,000	$10,000	$1,800,000,000
Sum/Average	1,718,375	$445,833	$2,440,455	$397,244,375,000

OVERALL TOTAL OF U.S. PRE-2000 AND POST-2000 REPAIRS AND RECOVERY

Function points deployed in the U.S.	1,702,125,000
LOC deployed in the U.S.	194,042,250,000

Year 2000 Topic	U.S. Year 2000 Cost Elements	Cost per Function Point	Cost per LOC
Expenses Prior to 2000 AD			
Initial Software Repairs	$70,000,000,000	$41.13	$0.36
Secondary "Bad Fix" Software Repairs	$7,000,000,000	$4.11	$0.04
Test Library Repairs	$10,000,000,000	$5.88	$0.05
Database Repairs	$60,000,000,000	$35.25	$0.31
Hardware Chip Replacements	$10,000,000,000	$5.88	$0.05
Hardware Performance Upgrades	$20,000,000,000	$11.75	$0.10
Subtotal	$177,000,000,000	$103.99	$0.91
Expenses After 2000 AD			
Litigation and Damages	$100,000,000,000	$58.75	$0.52
Post-2000 Damages	$116,075,625,000	$68.19	$0.60
Post-2000 Recovery Expenses	$281,168,750,000	$165.19	$1.45
Subtotal	$497,244,375,000	$292.13	$2.56
TOTAL	**$674,244,375,000**	**$396.12**	**$3.47**

The U.S. "National Debt" for the Euro and Year 2000

Year 2000 Cost Elements		Cost per Capita
Repairs	$177,000,000,000	$668
Damages	$487,244,375,000	$1838
	---	---
TOTAL	$674,244,375,000	$2506

Euro Currency Cost Elements		Cost per Capita
Upgrades	$60,000,000,000	$226
Damages	$175,000,000,000	$660
	---	---
	$235,000,000,000	$886

GLOBAL TOTAL OF PRE-2000 AND POST-2000 REPAIRS AND RECOVERY

Global Function Points Deployed	10,212,750,000
Global LOC Deployed	1,164,253,500,000

Year 2000 Topic	Global Year 2000 Cost Elements	Cost per Function Point	Cost per LOC
Expenses Prior to 2000 AD			
Initial Software Repairs	$530,000,000,000	$51.90	$0.46
Secondary "Bad Fix" Software Repairs	$50,000,000,000	$4.90	$0.04
Test Library Repairs	$75,000,000,000	$7.34	$0.06
Database Repairs	$454,000,000,000	$44.45	$0.39
Hardware Chip Replacements	$76,000,000 000	$7.44	$0.07
Hardware Performance Upgrades	150,000,000,000	$14.69	$0.13
Subtotal	$1,335,000,000,000	$130.72	$1.15
Expenses After 2000 AD			
Litigation and Damages	$300,000,000,000	$29.38	$0.26
Post-2000 Damages	$580,378,125,000	$56.83	$0.50
Post-2000 Recovery Expenses	$1,405,843,750,000	$137.66	$1.21
Subtotal	$2,286,221,875,000	$223.86	$1.96
TOTAL	**$3,621,221,875,000**	**$354.58**	**$3.11**

CONTINGENCY PLANNING: START RIGHT NOW

At least 20% of Year 2000 problems won't be fixed in time

At least 25% of Euro-currency updates won't be fixed in time

Start contingency planning in 1998

Perform risk analysis of your key applications

Perform risk analysis of your key suppliers

Perform risk analysis of your public utilities

Perform risk analysis of all government interfaces

Create an emergency response team

Create list of year 2000 and Euro crisis coordinators

CONTINGENCY PLANNING (Continued)

YEAR 2000 CONTINGENCY PLANNING REPRESENTATIVES

Year 2000 Executive or Project Manager

Corporate Legal Counsel + Outside Year 2000 Legal Specialist

Director of Risk

Director of Human Resources

Major Operating and Support Groups

- **Manufacturing**
- **Purchasing**
- **Finance and Accounting**
- **Sales and Marketing**

Year 2000 Contingency Consultants

Supplier, client, and even competitor Y2K specialists

Contingency Plan Topics

The impact of loss of electric power for > 1 day

The impact of loss of telephone services for > 1 day

The impact of loss of shipping and air travel for > 1 day

Manual alternatives for your automated operations:
Payroll, Invoicing, Order entry, etc.

Dealing with damages your errors cause your customers

Dealing with damages your suppliers cause you

Dealing with massive government errors:
Local, State, Federal

Dealing with data base and data mining issues

Defensive strategies to minimize litigation risks

Contingency Plan Topics (Continued)

The impact of loss of gasoline for > 10 days

The impact of loss of public transport for > 10 days

Moving key operations to other locations still able to operate

Providing temporary quarters for 24-hour operations

Year 2000 audit trails to minimize litigation risks

Euro audit trails to minimize litigation risks

Special director and officer liability insurance

Special business interruption insurance

Year 2000 Contingency Planning Calendar

July - September of 1999

- Emergency response team named
- Formal contingency plan reviewed and updated
- Preliminary contingency budget and cost estimate prepared
- Internal contact list updated
- External contact list updated
- Enterprise year 2000 status report prepared
- Client and vendor year 2000 status report prepared
- Infrastructure and government status report prepared
- Competitive status report prepared

Year 2000 Contingency Planning Calendar

October - December of 1999

- Emergency response team trained
- Formal contingency plan approved by attorneys and management
- Formal contingency budget and cost estimates approved
- Public announcements of year 2000 readiness and contingency plan
- Some assets shifted to "safe harbors"
- Infrastructure back-up equipment deployed
  - Emergency generators
  - Radio equipment
  - Fuel supplies
- Legal and liability status report prepared

Year 2000 Contingency Planning Calendar

January - March of 2000

- All applications monitored for date problems and reinfection
- All vendor and external applications monitored
- Emergency year 2000 repairs commence
- Post year 2000 cost estimates updated
- Initial filing of damage claims with insurers
- Initial filing of damage law suits against vendors
- Your own defense from external law suits commence
- Daily status reports on year 2000 repair progress
- Weekly press releases on year 2000 repair progress

Year 2000 Contingency Planning Calendar

April - June of 2000

- Litigation discovery proceeds; some depositions may start
- If problems are severe, bankruptcy filings may commence
- Revised SEC cost statement prepared with post-2000 data
- Overall economic analysis of year 2000 impact prepared
- Manual back-up procedures phased out; automation resumes
- Some year 2000 teams can resume normal work
- Some suspended development projects can resume
- Plans commence on next wave of problems

EURO CONTINGENCY PLANNING DURING 1998

At least 25% of U.S. software Euro updates won't be fixed in time

Perform risk analysis of your key financial applications

Train key personnel in triangulation method

Select and train a Euro fast-response team by 3Q98

Prepare manual back-ups for financial transactions
- Retail sales
- Currency exchanges

Create a budget for Euro damages and recovery costs

Examine Euro impact on data and decision support
- Data mining
- On-line analytical processing (OLAP)

Personal Year 2000 Contingency Planning

Problem Area	Odds	Contingency
• IRS tax errors	55%	Keep paper records in 1999
• Air travel problems	30%	Caution in January 2000
• Road travel problems	25%	Caution in January 2000
• Sea travel problems	25%	Check status in 3Q99
• Stocks and bonds	25%	Check status in 2Q99
• Heating oil	20%	Refill in December 1999
• Natural gas	20%	NONE - check supplier 3Q99
• Telephone service	20%	Try cellular as backup

Personal Year 2000 Contingency Planning

Problem Area	Odds	Contingency
• Sewage treatment	20%	Check status in 3Q99
• Bank/ATM failures	15%	Keep some cash ready
• Payroll errors	15%	Check status in 3Q99
• Credit report errors	15%	Check credit often in 2000
• Insurance errors	15%	Keep paper records of claims
• Electric power loss	15%	Generator + fuel supply
• Radio/TV outtages	15%	Battery short-wave radio
• Gasoline shortages	15%	NONE - Check in December

Personal Year 2000 Contingency Planning

Problem Area	Odds	Contingency
• Food shortages	15%	Keep two-week supply
• Drinking water	15%	Keep 5 gallons (bottled)
• Sewage treatment	15%	NONE - Check status 3Q99
• Cellular phones	10%	Use CB radio as backup
• Prescriptions refills	10%	Refill in December of 1999
OVERALL		Check newspapers, television, and other sources of year 2000 status reports during last half of 1999.

Urgent Contingency Planning Needs

- A National register of Year 2000 and Euro managers in the U.S.
- National registers of Year 2000 and Euro managers in the EMU.
- A formal statement by the IRS that they will not take punitive action if taxpayer records are wrong due to Year 2000 problems.
- A "safe harbor" law so that companies can publish contingency plans without the plans being used against them in litigation.
- A new SEC requirement that Year 2000 compliance statements include contingency plan cost estimates.
- A formal statement by the White House that U.S. military services will assist in restoring electric power, fuel supplies, water supplies, and essential services after Year 2000 problems occur.

Summary and Conclusions

- Starting in 1999 date and field expansions will dominate software.
- These problems all have one root cause: fields were too small.
- Between 1999 and 2050 field expansions will cost $5 trillion dollars.
- We need general solutions - not many separate solutions.
- The software problems of the Euro, Year 2000, and other dates will damage the software industry for a generation.
- Contingency planning must start now in 1998.
- Contingency planning must cover all risk factors.
- It is too late to fix all Euro and Year 2000 problems.
- The best we can hope for is a quick recovery.