

**Informe
No. 24**



Enfrentando el año 2000. Guía Práctica

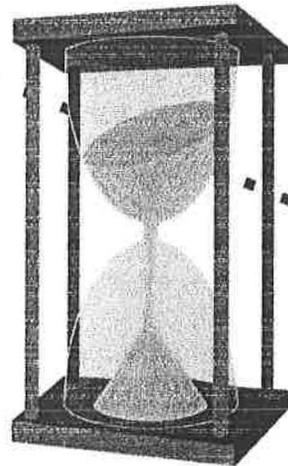
Club de Investigación Tecnológica

Enfrentando el año 2000
Guía práctica

Preparado por: Ing. Carlos Gallegos, Dr. Roberto Sasso e Ing. Ignacio Trejos

Mayo 1998

31/12/99



01/01/00

**Editado y publicado por Rho-Sigma, S.A.
a nombre del Club de Investigación Tecnológica.
Todos los derechos reservados.
Prohibida la reproducción total o parcial.
San José, Costa Rica
Mayo 1998**

Resumen Ejecutivo

El “problema del año 2000”, también conocido como “problema del cambio de siglo” y “problema del cambio de milenio”, se presenta como uno de los retos empresariales e informáticos más importantes de los últimos tiempos.

El problema se origina en una época en que las computadoras y los dispositivos de almacenamiento eran muy costosos, por lo que las empresas debían maximizar el aprovechamiento de esos recursos. Informáticamente, esto se tradujo en la técnica de representar los años mediante dos dígitos.

Hoy esto trae como consecuencia que, ante el paso del año 1999 al año 2000, la mayoría de los sistemas interpretarán el año “00” como “1900” y no como “2000”.

La corrección de un programa o un archivo individual no es particularmente difícil. La dificultad radica en que hay que revisar *todos* los sistemas para determinar su sensibilidad al manejo de fechas y, además, deberán localizarse todos los lugares que dependen del manejo de fechas (ocurrencias o instancias del problema) en los programas y bases de datos.

Este es un problema en que:

- el *tamaño* es particularmente importante
- deben revisarse todos los sistemas y repararse la mayoría de ellos
- múltiples sistemas deberán integrarse a la producción de manera sincronizada
- el intercambio de información con entes externos puede depender del manejo de fechas
- hay dispositivos y sistemas cuya dependencia del manejo de fechas no es obvia
- *el plazo es fijo, inamovible.*

Este informe presenta un conjunto de consideraciones prácticas para enfrentar el cambio de siglo. Primero se dan los antecedentes del problema y luego se enumeran acciones inmediatas que pueden llevarse a cabo. Por ser un proyecto de naturaleza fuera de lo común, recomendamos hacer un sub-proyecto piloto para afinar el enfoque que adoptará la organización para enfrentar el problema. Después aparecen secciones sobre el diagnóstico del problema en la organización, la formulación de un plan de acción, la ejecución del proyecto y las pruebas requeridas para validar las soluciones. El informe concluye con un resumen de las principales recomendaciones. Información adicional, que consideramos útil, ha sido puesta en los apéndices.

De los autores

Carlos Gallegos, Ingeniero Eléctrico con una concentración en sistemas y computadoras de la Universidad de Carleton, Ottawa, Canadá, es socio de Consultoría de KPMG Peat Marwick en Centroamérica. Tiene amplia experiencia en el desarrollo de proyectos, nacionales e internacionales, en las áreas de estrategia, tecnología de información, manufactura, productividad y auditoría de sistemas. Es miembro activo del IEEE y ha ocupado cargos directivos en la Sección Costa Rica.

Roberto Sasso, consultor gerencial en tecnología de información, es Doctor en Ingeniería de Software de la Universidad de Oxford, Inglaterra, presidente del Club de Investigación Tecnológica, Director de Computer World América Central y Vice Presidente de la compañía de consultoría Rho-Sigma S.A..

Ignacio Trejos, Ingeniero en Computación del Instituto Tecnológico de Costa Rica, M.Sc. y candidato doctoral de la Universidad de Oxford, Inglaterra, es profesor del Instituto Tecnológico de Costa Rica, Director de Investigación del Club de Investigación Tecnológica, miembro del Consejo Editorial de Computer World América Central y consultor en Ingeniería de Software. Es miembro de la Red Iberoamericana de Tecnología de Software del CYTED.

Contenido	Página
Antecedentes	1
Dónde aparece el problema	2
¿Qué hace al problema del 2000 tan peculiar?.....	3
Costo de la reparación.....	3
Enfoque del problema	4
Acciones inmediatas	5
Organización del proyecto.....	6
Perfil de recursos y Motivación	6
Proveedores y Suplidores	7
Herramientas.....	7
Equipo e instalaciones.....	8
Usuarios finales.....	8
Seguimiento y avance.....	9
Repositorio	9
Aseguramiento de Calidad y Administración de mejoras	10
Piloto	10
Diagnóstico	11
Inventario	11
Detección y evaluación.....	12
Impacto	14
Riesgos	14
Oportunidad de falla.....	14
Plan de acción	15
Prioridades.....	15
Estrategias	15
Definición de estrategia	17
Recursos	18
Tablero de control (“pizarra marcador”).....	18
Ejecución	19
Pruebas	19
Ambiente	20
Alcance.....	20
Aceptación.....	21
Conclusiones	21
Apéndice A: Algunas fechas críticas	23
Apéndice B: Patrones para localización de fechas	24
Apéndice C: Ejemplos de código	25
Apéndice D: Revisión del plan	26
Apéndice E: Tablero de control (pizarra marcador)	27
Áreas del tablero	27
Apéndice F: Recursos	29
Herramientas.....	29
Información	30
Bibliografía	31

Antecedentes

Se ha escrito muchísimo sobre el llamado *problema informático del año 2000*. En tiempos recientes, prácticamente en cualquier tipo de publicación – desde diarios hasta revistas científicas – ha aparecido alguna mención del problema. La voz de alarma que se dio oportunamente en nuestro medio no encontró oídos dispuestos a escuchar. Las sociedades modernas enfrentan el primer problema amplio de carácter informático.

El problema aparece en cualquier programa o sistema que haga un manejo de fechas diferido (proyección a sucesos futuros) o que compare fechas, cuando estas han sido representadas de alguna manera “compacta”. Representaciones de la forma AAMMDD (año-mes-día) o sus variaciones (DDMMAA, MMDDAA) surgieron en la década de 1950 por apego a representaciones textuales en tarjeta perforada (80 columnas) y por economía de espacio; en aquella época la memoria principal y el almacenamiento externo eran demasiado caros. Popularmente esto ha significado representar los años en dos dígitos (AA) *como caracteres* (ASCII o EBCDIC), en lugar de cuatro. Es muy raro el sistema que tiene representaciones alternas: enteros grandes sin signo, fechas julianas, estándar ISO 8601:1988, etc.

Este problema tiene una manifestación informática pero es un problema *empresarial*. No nos equivoquemos: la cultura de economizar espacio de almacenamiento fue alimentada por la presión financiera de maximizar la inversión en tecnología informática hecha por las empresas.

Al llegar el primero de enero del 2000 esta fecha será interpretada por sinnúmero de sistemas como el primero de enero de 1900. Otro problema relacionado, y obviado por algunos, es que al pasar el 1 de enero del 2000, este año no sólo será interpretado como 1900 (el pasado), sino que 1900 *no* fue un año bisiesto, mientras que el 2000 *sí* lo es.

Hay otros problemas de manejo de fechas latentes en el mercado: la manera en que se cuenta el tiempo en sistemas operativos Unix (se supone que el problema aparecerá cerca del 2037 o 2038), Excel maneja 1900 como bisiesto (no lo es) y no maneja correctamente las fechas después del 2078, hasta hace pocos meses el popular programa financiero Quicken no manejaba correctamente el cambio de siglo. Desconocemos si alguna particularidad de NT trae implícito algún problema de manejo de fechas o tiempo.

El problema del año 2000 involucra uno o más de los siguientes aspectos [Martin 1997]:

- Representar el año como un número de dos dígitos. Esto provoca errores en comparaciones, cálculos aritméticos, clasificación, acceso indizado a bases de datos y archivos, etc.
- Utilizar un algoritmo incorrecto para reconocer años bisiestos divisibles por 400.
- Codificar “19” en rutinas de software. Utilizar años de dos dígitos con “99” y “00” como valores reservados, con significados semejantes a “nunca borrar esto” o “esta es una cuenta demostrativa”, respectivamente.
- Tipos de datos de fecha del sistema que pueden desbordarse y fallar por haber llenado la capacidad de representación de registros de almacenamiento.

Dónde aparece el problema

El problema es común a aplicaciones que manejen plazos diferidos y ya está comenzando a manifestarse en varias aplicaciones. Algunos ejemplos en Costa Rica:

- Manejo de obligaciones: bonos, títulos valores, certificados de inversión, otras inversiones a plazo. El problema se manifiesta desde las perspectivas del que ofrece y del que invierte.
- Fondos de pensiones.
- Planillas: salarios, prestaciones, vacaciones, deducciones periódicas, etc.
- Operaciones crediticias (perspectivas del prestamista y del prestatario).
- Seguros.
- Fondos de ahorro y préstamo (mutuales, INVU, cooperativas, solidaristas, etc.).
- Cuentas de ahorro y corrientes con interés.
- Vencimientos de instrumentos de pago (tarjetas de crédito, tarjetas de débito, otros).
- Cuentas por cobrar y cuentas por pagar (antigüedad de saldos).
- Facturación de servicios públicos periódicos, particularmente los telefónicos (duración de las llamadas).
- Seguimiento de órdenes. Compras periódicas.
- Suscripciones.
- Clubes de viajes.
- Vencimiento de productos farmacéuticos y alimenticios.
- Manejo de productos perecederos, tóxicos o cualquier otro con fecha de expiración.
- Planificación de la producción.
- Análisis sobre información histórica.
- Pronósticos.

El problema no se circunscribe a los programas cuyo código se tiene a disposición en la organización. Hay sutiles dependencias tecnológicas prácticamente en cualquier dispositivo electrónico programable o ambiente de software:

- Centrales telefónicas privadas y públicas.
- Dispositivos de seguridad (controladores de bóvedas, sistemas de alarmas, etc.).
- Cajeros automáticos.
- Sistemas de navegación aérea y marina.
- Cajas registradoras.
- Dispositivos y sistemas con control temporizado: relojes de tiempo real, equipo electromédico, provisión energética, regulación ambiental (aire acondicionado, calentadores), anti-incendios, etc.
- Sistemas inter-bancarios y de intercambio electrónico de datos.
- Servicios de seguridad y autenticación.
- Sistema operativo (particularmente la capa del BIOS).
- Generadores de números aleatorios.
- Sistemas administradores de bases de datos. Sistemas de procesamiento de transacciones.
- Lenguajes de programación interpretados o compilados.
- Lenguajes de control de trabajos (JCL, OCL, etc.).
- Hojas de cálculo.

Por ejemplo [Martin 1997], en enero de 1997 el National Institute of Health de los Estados Unidos recibió un embarque de PCs nuevas con tres tipos distintos de BIOS, dos de las cuales fallaron durante las pruebas de cambio de siglo.

Casi todas las organizaciones modernas intercambian información. Muchos intercambios de información se dan por transferencia de datos, acceso remoto o transferencia de archivos (el estándar genérico del Intercambio Electrónico de Datos (EDI), cuyo proyecto piloto está en marcha e involucra a más de 10 empresas costarricenses, permite fechas donde los años son representados con sus últimos dos dígitos) [Sanabria 1997]. Además, muchos de los formularios pre-impresos aún tienen diseños con dos dígitos para el campo de año en las fechas.

¿Qué hace al problema del 2000 tan peculiar?

El problema del año 2000 tiene peculiaridades que lo distinguen del mantenimiento de sistemas convencional:

1. Tiene un plazo *impostergable* que es común a toda organización. Todos competirán por recursos escasos: programadores COBOL, personal para pruebas, etc.
2. Difiere del mantenimiento convencional. No tenemos el usual “este es el problema, arréglo”, sino “vaya busque dónde está el problema y determine las correcciones apropiadas”.
3. Afecta a cualquier sistema que tenga interfaces externas: sean estas automáticas, semi-automáticas o manuales. Cambios y adaptaciones, probablemente incompatibles, podrían ser planteados entre consumidores y suministradores de información; todos deben ser resueltos simultáneamente y de manera consistente.

Costo de la reparación

No se cuenta con un estudio que haya cuantificado el costo de reparar el problema del año 2000 en Centroamérica. El nivel promedio de computarización y automatización existente es muy variable, así como las dimensiones de los sistemas. Aún así, si el comportamiento es semejante al diagnosticado en los países desarrollados, podemos concluir que la inversión será alta; aunque es esperable que sea inferior, dada la menor informatización local, también deberá tenerse presente que se ha postergado la atención del problema en nuestra región.

En 1995, Mitre (www.mitre.org) desarrolló una investigación, sobre dos sistemas específicos, para el Departamento de Defensa de los Estados Unidos de América. Se analizaron 5.4 millones de líneas de código fuente, donde se buscó principalmente nueve tipos de instrucciones relacionadas con la manipulación de fechas. El estudio demostró que 1.16% de los programas (cerca de 62,600 líneas de código) manipulaban fechas y eran utilizados por el 10.03% de los módulos de los sistemas. El estudio de Mitre determinó que el costo estimado del mantenimiento correctivo de estos sistemas oscilaría entre \$0.75 y \$1.70 por línea de código en aplicaciones de sistemas de información y de \$1.00 a \$8.52 por línea de código en sistemas de control y comando. Esta información general puede ser utilizada para estimar el costo *grueso* en mantenimiento que los sistemas requieren, al suponer que entre el 10% y el 15% de los sistemas deben ser modificados.

Capers Jones [Jones 1996], de Software Productivity Research, ha estimado que, para Estados Unidos, los costos de mano de obra para reparar programas ascienden a 68,699,020,000 US\$ para un total de 8,465,060 meses/persona. Los costos de reparar bases de datos ascenderían a 57,749,380,000 US\$. Las estimaciones del Gartner Group son mayores: 300,000,000,000 US\$ (300 mil millones de dólares US) para Estados Unidos y otro tanto para el resto del mundo. Si se incluyen también los costos de reemplazo de circuitos, litigios y perjuicios, la estimación mundial asciende a 1,600,000,000,000 US\$ (1.6 billones de dólares US) en asuntos relacionados con el problema [Jones 1997b].

La reparación de cada error, una vez encontrado, puede ser trivial. La dificultad principal es *encontrar* las ocurrencias del problema en grandes volúmenes de código fuente (cuando éste existe), asegurarse que se ha realizado adecuadamente la reparación y que *no* se han introducido nuevos errores. En el mantenimiento típico de aplicaciones se vuelven a inyectar errores a una tasa del 7%, pero la estimación para el problema del año 2000 es de un 10% (se reporta un rango entre el 7% y el 12% en compañías que están haciendo reparaciones) [Jones 1997b].

Enfoque del problema

Para quien aún no ha iniciado acciones para reparar sus aplicaciones informáticas, el panorama no es halagüeño. Según [Jones 1996], éste será el comportamiento esperado en cuanto a la cobertura de aplicaciones:

Año en que se comienza a reparar	Porcentaje de aplicaciones corregidas en 1999
1994	100%
1995	100%
1996	99%
1997	80%
1998	60%
1999	30%

Un esfuerzo para resolver el problema del año 2000 típicamente involucra seis fases:

1. Concientización (sensibilización), en que se hace saber al resto de la organización el riesgo al que ésta está expuesta.
2. Inventario, donde se identifican los sistemas y – en general – la tecnología sensibles al problema del año 2000.
3. Evaluación y planificación, en que se determina la magnitud del problema, las estrategias aplicables a la situación encontrada y que son aún factibles, así como la conformación de un plan hacia la solución. También en esta fase se negocia con quienes se intercambia información (proveedores, clientes, gobierno y otros consumidores) acerca de la capacidad para manejar adecuadamente el cambio del milenio.
4. Reparación, en que se aplica la estrategia elegida para cada sistema: reemplazo, reescritura o renovación.

5. Validación, en que se prueban los sistemas cambiados para comprobar el manejo adecuado de fechas y la eliminación de cualquier defecto residual introducido durante la reparación.
6. Instalación, las aplicaciones reparadas y probadas se integran al ambiente de producción junto con las bases de datos convertidas.

La distribución del esfuerzo en los proyectos tiende a comportarse así [Jones 1996]:

Tarea	Rango
Encontrar instancias del problema del año 2000	10% - 50%
Reparar instancias del problema del año 2000	15% - 30%
Pruebas: validar las reparaciones	10% - 30%
Pruebas: de regresión a la cartera de sistemas	20% - 50%

Acciones inmediatas

Empezar un proyecto del año 2000 puede ser como entrar a un túnel por primera vez. Se sabe dónde se puso oscuro, pero no hay certeza de cuándo se saldrá de nuevo a la luz, ni los obstáculos que se puedan encontrar en el túnel. Se va a experimentar esa misma sensación de inseguridad y desasosiego. Pero se debe seguir adelante y al igual que en el túnel hay que tomar acciones inmediatas (poner las luces, ajustar el radio y conducir en estado de alerta).

En los proyectos del cambio de siglo¹ hay acciones que pueden y deben tomarse en los primeros días tanto en el área administrativa como en la financiera y la contable.

Administrativas. ¡Declare la guerra! Esto es, anuncie hacia lo interno de la organización que se va a pelear y a pelear de verdad. Nombre un Director de Proyecto, alguien con suficiente poder y credibilidad para sacar la faena (no escatime en esta decisión, nombre a su mejor director de proyectos). Conforme un Comité o Grupo de trabajo con representantes de las principales unidades de negocio, esto es, las áreas donde se generan los ingresos y las utilidades de su organización. Participe al área de tecnología de información, pero no los abrume con la responsabilidad total del proyecto. El problema del año 2000 es un riesgo y un problema de negocios.

Anúnciele a la organización que se está trabajando en serio en el asunto. Asigne un presupuesto y sea realista, se van a requerir muchos colones/dólares para ganar la batalla. Defina cuanto antes qué significa para su organización “ser certificado”, “cumplir”, o “estar preparado”. Sea razonable, no sea muy ambicioso en esta definición, el tiempo se agota irremediabilmente. Pare el sangrado inmediatamente y dé instrucciones expresas y claras a su departamento de compras o adquisiciones de no comprar ningún bien o servicio que no sea “certificado” año 2000, o que en condiciones especiales se tenga una alternativa y un plan de contingencia listo y presupuestado.

¹ Se le llamará también, indistintamente, cambio de milenio. Para efectos de este informe, el cambio se da en la transición entre los años 1999 y 2000.

Contables. Según un pronunciamiento de autoridades competentes, todas las inversiones y costos relacionados con el problema del año 2000 deben registrarse como un gasto en el período en que se incurrieron. Mida el impacto en su hoja de resultados y prepare una estrategia financiera al respecto.

Legales. Consulte a sus asesores legales la mejor forma para pedir y responder a las consultas, solicitudes y otras preguntas de proveedores y clientes sobre el año 2000. Recuerde que en varios países las Juntas Directivas tienen la obligación de revelar la preparación de sus organizaciones en cuanto al manejo del cambio de siglo a accionistas, empleados, gobierno, clientes y proveedores.

Haga todo lo que debe hacer y empiece hoy mismo, mañana puede ser muy tarde.

Organización del proyecto

Los proyectos del año 2000 son proyectos complejos y delicados, se requiere una alta dosis de administración de proyectos y de planeación y control. Si bien en este informe nos estamos enfocando más en empresas medianas, la organización del proyecto es vital.

Vamos a definir dos términos muy importantes. El primero es la Oficina de Administración del Proyecto (OAP), como la responsable general del proyecto del año 2000, pero no necesariamente será la responsable de ejecutar las pruebas para lograr certificaciones. La OAP puede ser responsable de coordinar las certificaciones de proveedores, terceros, equipo e instalaciones, usuarios finales, coordinación de pruebas y preparación de informes a la Alta Dirección.

El segundo es la Oficina del Proyecto, OP, quien es responsable por la renovación del código incluyendo las pruebas.

La estructura puede variar de una organización a otra y algunas funciones se pueden mezclar. En organizaciones pequeñas y medianas la OAP y la OP son la misma, mientras que en organizaciones grandes y descentralizadas normalmente la OAP está en la Corporación y las OP en las divisiones.

Perfil de recursos y Motivación

Tanto en la OAP como en la OP se van a requerir recursos humanos, técnicos y financieros. Vamos a tratar rápidamente el perfil y la motivación de los primeros. Este es un proyecto que requerirá de la mejor gente (sus mejores soldados). El Gerente/Director del Proyecto debe ser alguien muy competente, organizado, con ideas claras del negocio y la industria en que opera. Debe ser un capitán con autoridad y responsabilidad. Su misión es llevar el barco a buen puerto, su motivación nace del reto por delante y del posicionamiento para tomar proyectos de igual envergadura en el futuro. Prepárese para recompensarlo adecuadamente al cumplir la tarea.

El resto del personal debe ser automotivado y jugadores de equipo. Deben conocer su trabajo y algunos deberán ser excelentes técnicos. No incluya a la gente que le “sobre el tiempo” o que no tienen nada importante que hacer en la organización, sólo van a hacer más difícil el trayecto. Los técnicos deben ser prácticos, no es hora de experimentar con lo último en la tecnología, es hora de actuar y avanzar.

La motivación para los miembros del equipo es sumamente importante. Algunos de ellos pueden sentir cierta reticencia a dejar proyectos a los que ya hayan sido asignados. Hágales saber que el Proyecto del Cambio de Siglo es muy importante para su organización y que el esfuerzo y los resultados que ellos obtengan serán tomados muy en cuenta a futuro. Explíqueles que son parte de un equipo con una misión especial y que el conocimiento y experiencia que acumulen serán factores de éxito en su desarrollo profesional.

La OAP debe asegurarse de cubrir varias áreas y tomar acciones en al menos los siguientes aspectos:

Proveedores y Suplidores

La OAP debe crear, implementar y gerenciar los procesos y los procedimientos que aseguren que los bienes y servicios de los proveedores, suplidores de bienes y servicios, y los socios comerciales cumplen con la definición de certificación establecida por la compañía. También la OAP debe crear, implementar y gerenciar los procesos y procedimientos para evaluar la preparación y status de los proveedores, suplidores y socios comerciales con respecto del año 2000.

El objetivo es reducir o mitigar el riesgo de que productos o servicios dados por suplidores o socios comerciales vayan a interrumpir o impactar negativamente la operaciones del negocio por problemas relacionados con el año 2000.

Igualmente el objetivo es reducir el riesgo de que estos suplidores o socios comerciales puedan impactar la organización por problemas propios de la suya, sufridos como consecuencia del cambio de fecha.

Herramientas

La OAP debe escoger las herramientas adecuadas. Hay diferentes categorías de herramientas, así unas son para inventariar, o analizar, otras para administrar configuraciones. Otras son para conversión de código, o pueden ser para monitoreo de errores o conversión de datos. Una categoría aparte de herramientas es la de pruebas donde se encuentran diferentes subdivisiones (añejamiento, comparación de datos, depuración, cobertura, re-ejecución de sesiones, simulación) y finalmente hay herramientas de implementación. Algunas referencias a herramientas se incluyen en el Apéndice F.

No hay mucho tiempo para evaluar y seleccionar herramientas pero las que se escojan deben serlo con base en criterios de exactitud, flexibilidad, volúmenes, costo, facilidad de uso, minuciosidad e impacto.

La estrategia de renovación dictará las pautas por seguir en la selección de herramientas. La mayor parte de las organizaciones probablemente tenga ya un conjunto robusto de herramientas. Considere lo siguiente: a) las herramientas ya existían antes de la crisis del año 2000, b) instaladas no significa que sean usadas en la actualidad y c) algunos proveedores están re-empacando herramientas comunes para aprovechar la demanda.

Recuerde, escoja una buena herramienta. Aquí aplica aquello de que lo perfecto es enemigo de lo bueno.

Equipo e instalaciones

El alcance de equipo e instalaciones en relación con el año 2000 es el siguiente: los sistemas, componentes, activos y artefactos que dependen de una fecha en un microchip, pero que no son considerados normalmente como equipo de cómputo o software por la organización.

Se deben considerar tanto las instalaciones locales como las regionales o internacionales. Asimismo, pueden ser de carácter administrativo, financiero, de manufactura o de despacho y transporte.

Entre los tipos de equipo por considerar tenemos: elevadores, sistemas de control de energía, sistemas de protección ante incendios, ventilación, calefacción, aires acondicionados, dispositivos de mano, controles numéricos computarizados, vehículos y aviones, controladores de lógica programables (PLC), sistemas de seguridad con sus cámaras de vídeo y lectores ópticos, UPS y generadores, bóvedas, control de inventarios y dispositivos de comunicación de planta.

El objetivo es establecer los procesos y procedimientos para asegurarse que las diferentes instalaciones y equipo hayan sido certificadas o probadas para el año 2000.

Usuarios finales

El objetivo de esta área es asistir y monitorear el progreso de los usuarios finales en relación con el año 2000 y la certificación del hardware, software y aplicaciones en sus manos. Por usuarios finales entendemos aquellas personas o entidades fuera de la tradicional área o departamento de informática. También en organizaciones grandes puede incluir a usuarios de aplicaciones *no en producción* del área de sistemas.

El objetivo es soportar a los usuarios finales de la organización a ser certificados en el año 2000 y evitar así un impacto negativo en la organización.

No debe subestimarse el esfuerzo que tratar con los aspectos de los usuarios finales pueda tener en un proyecto de esta naturaleza. Dependerá en mucho del grado de descentralización que tenga su organización. Si los usuarios finales han estado acostumbrados a adquirir directamente sus aplicaciones y los equipos relacionados, y si no se han seguido estándares para toda la empresa, puede ser que tenga una enorme tarea por delante. Evalúe qué tan críticas son esas aplicaciones para su organización; ¿son, por ejemplo, simples hojas de cálculo que presentan en un formato particular y personalizado información corporativa y centralizada?, o ¿son aplicaciones que realmente agrupan y manipulan datos e información esenciales para la actividad empresarial o de la organización? En función de lo anterior deberá diseñar su estrategia para esta área.

Seguimiento y avance

En proyectos de esta naturaleza es indispensable contar con un mecanismo para **determinar y reportar la salud del proyecto**.

Los objetivos específicos de esta área son: mantener información resumida y actualizada sobre el plan de trabajo, mantener planes individuales de cada OP, analizar los planes presentados, reportar sobre el status del proyecto, informar sobre los principales logros, administrar y resolver conflictos. Para lograr lo anterior hay al menos tres funciones básicas que deben estar presentes y que son: actividades de reporte, resolución de conflictos y administración del riesgo. Más adelante se dan consideraciones respecto de una sencilla herramienta que facilita el seguimiento: el "tablero de control".

Este es un proyecto que debe tener la participación directa y constante de la Alta Dirección, quien es en última instancia la responsable de asegurar la supervivencia y éxito de la organización. No debe cometerse el error de creer que es un problema y un asunto técnico. Las más altas autoridades deben asegurarse de que gracias a su participación y monitoreo constantes se le dé la importancia y dimensión adecuada al proyecto.

Repositorio

El propósito de esta área es soportar el análisis e informar la certificación de objeto y su estado en función de medidas de desempeño, estrategias (priorización), mejores prácticas y seguimiento detallado de actividades. Un *objeto de certificación* es cualquier entidad que pueda ser monitoreada sobre el impacto del año 2000. El dueño responsable del objeto de certificación es el gerente usuario que tiene la responsabilidad del desarrollo, mantenimiento o uso del objeto de certificación. El objetivo final es mantener una base de datos que soporte a la OAP en sus requerimientos de reporte de información. Para ello es indispensable que permita poderosas consultas ad-hoc y que soporte una visión integral de la organización. Es muy importante que no se confunda este repositorio con el objetivo final del proyecto. No debe confundirse el árbol con el bosque. El repositorio es una ayuda para la administración del Proyecto del año 2000, no lo convierta en un proyecto en sí mismo.

Aseguramiento de Calidad y Administración de mejoras

Es imprescindible definir los requerimientos de calidad y las guías que se van a emplear en el desarrollo del proyecto. Es importante tratar de maximizar cualquier esquema de control de calidad que ya exista en la organización o en su defecto incorporar uno adecuado. La OAP debe definir claramente los requerimientos y controles de calidad y debe asegurarse que se documenten apropiadamente. Los factores críticos de éxito consideran la responsabilidad de la organización por la certificación, contemplar requerimientos legales, clara definición de los requerimientos de certificación, adecuada política de comunicación, gente preparada y entrenada.

Piloto

Para poder planificar con cierto grado de detalle el trabajo a realizar es necesario llevar a cabo un piloto en el que se “certifique” (sea reemplazando o renovando) una aplicación. Este piloto es necesario debido a que no se cuenta con experiencia anterior en la cual basar cualquier estimado. El Piloto debería además contemplar un ejercicio de “certificación” con uno de los proveedores de tecnología (base de datos, central telefónica, etc.) y una aplicación de usuario final.

El objetivo del Piloto, por lo tanto, no es tan solo lograr la “certificación”, sino sobre todo el **aprender** acerca del proceso de “certificación” de todas las aplicaciones así como de software especializado, equipos electrónicos y aplicaciones de usuario final.

La aplicación que se escoja para el Piloto es de suma importancia. Debe ser suficientemente compleja para que el esfuerzo sea representativo, y suficientemente sencilla para poder obtener resultados a corto plazo.

El proceso de certificación en algunos casos puede no requerir ni de reemplazo ni de renovación de equipos y/o programas, pero en todos los casos requerirá de **pruebas**. Si bien es cierto que a base de pruebas nunca podremos demostrar la ausencia de errores, si las pruebas son bien diseñadas y son ejecutadas con suficiente rigurosidad, podremos obtener un alto grado de confianza en la capacidad que tengan las aplicaciones de sobrevivir el cambio de siglo.

En el caso en que sea necesario modificar las aplicaciones para poder pasar las pruebas, debe estar muy clara la necesidad de probar no solo las fechas claves sino también la funcionalidad de la aplicación (con frecuencia al arreglarle algo a una aplicación, se rompe por otro lado).

Una parte muy importante del aprendizaje es la instalación de un ambiente de pruebas en el cual se pueden realizar las modificaciones y pruebas, sin interrumpir la operación normal del negocio. Idealmente debería ser un ambiente de hardware, software y datos idéntico al ambiente real en el que están operando los sistemas.

Al escoger la aplicación piloto deberían también tomarse en cuenta las interfaces entre los diversos sistemas. El orden en que se “certifiquen” las aplicaciones deberá ser tal que permita la implementación inmediata de cada aplicación certificada. A toda costa se deberá desestimular el mantenimiento (funcional) en aplicaciones que se encuentran en proceso de “certificación”.

El resultado principal del Piloto, es por lo tanto, un conocimiento mucho más realista de la magnitud y el detalle de las labores por realizar. Este conocimiento no es solo para ilustrar y concientizar al personal técnico, este conocimiento deberá compartirse con todos los involucrados en el proyecto.

Si bien es imposible realizar una planificación certera para un proyecto (de cualquier índole) nunca antes realizado, el Piloto debería producir suficiente información y conocimiento para reducir considerablemente la incertidumbre.

Diagnóstico

Después del piloto o, preferiblemente, en paralelo con él, debe procederse a diagnosticar el estado en que se encuentra la organización en relación con su capacidad de manejar el cambio de siglo. Asimismo, deberá cuantificarse el impacto que tendría en la organización el no manejar adecuadamente el problema, así como la probabilidad de falla que esto tendrá. Este análisis de riesgo permitirá determinar la estrategia adecuada para cada sistema, así como la asignación de prioridades entre estos.

Comenzar definiendo un plan para el diagnóstico y formar uno o más equipos para realizar el inventario. Paralelamente, diseminar en los niveles directivos y gerenciales una definición común de los tipos de problemas que se desea enfrentar, así como las metas del esfuerzo de cambio de siglo. Asegurarse que los niveles directivos y gerenciales permanecen informados y comprometidos con el esfuerzo.

Inventario

1. Formar una cartera de sistemas de información:
 - Identificar los sistemas de cuya correcta operación depende la funcionalidad de la empresa y los servicios al cliente.
 - Añadir los sistemas de cuya correcta operación depende la adecuada toma de decisiones estratégicas y tácticas, así como el control de la empresa.
 - Añadir cualquier otro sistema considerado importante.
 - Obtener un inventario de los proyectos en ejecución (internos o contratados), sus fechas esperadas de entrega y la probabilidad de su cumplimiento.
2. Para cada uno de los anteriores, determinar:
 - si ya existe un proyecto en curso para sustituirlo: fecha esperada de entrega, fecha esperada de sustitución, capacidad del sistema sustituto para manejar el cambio de siglo.
 - si el sistema fue desarrollado en casa o si fue adquirido externamente. En el primer caso, determinar quiénes fueron los autores (analistas, programadores) que lo desarrollaron. En el segundo caso, determinar la empresa proveedora.
 - disponibilidad de código fuente, "sincronía" del código fuente con el código objeto que está en producción. Disponibilidad de documentación adicional del sistema.

- tecnología: lenguaje y dialecto (versión, bibliotecas de rutinas, etc.), plataforma de ejecución (sistema operativo, sistema administrador de bases de datos / administrador de archivos, arquitectura de comunicaciones, monitor de transacciones).
3. Con analistas y usuarios revisar la completitud de la cartera de sistemas. El objetivo es confirmar que no quedan sistemas o aplicaciones importantes por fuera.
 4. Obtener un inventario de la tecnología de información de la empresa:
 - software de operación (sistema operativo, sistema administrador de bases de datos, monitor de transacciones, sistema de red, monitores de comunicaciones, etc.).
 - software de oficina.
 - equipo de cómputo.
 - equipo de comunicaciones.
 - dispositivos controlados automáticamente.

En cada caso, obtener la versión, fecha de adquisición y proveedor.

5. Obtener un inventario de las organizaciones con las cuales se intercambia información de manera automática o semi-automática. Considerar tanto entradas (consumo) como salidas (provisión).

Detección y evaluación

Determinar la probabilidad de falla de cada elemento inventariado. Una primera aproximación se puede obtener mediante el juicio de un conjunto de usuarios y analistas, tomando en cuenta la capacidad que tenga el elemento de cumplir con el cambio de siglo.

1. Para los sistemas propios:
 - Con los analistas y principales usuarios determinar cuáles de los sistemas dependen del manejo de fechas, cálculo de intereses o cualquier otro elemento de información que se obtenga en función del tiempo. Determinar cuáles sistemas dependen de información calculada con base en manejo de fechas/tiempo directamente (vía bases de datos, comunicaciones) o indirectamente (vía redigitación de datos).
 - Someter los sistemas a pruebas simulando fechas críticas. Ver Apéndice A.
 - Hacer búsquedas en el código de los programas para determinar grado de cumplimiento. (Ver Apéndice B para sugerencias de patrones de búsqueda).
 - Cuidado con la conversión de archivos y bases de datos.
2. Para los sistemas adquiridos:
 - Solicitar al proveedor constancia de su grado de cumplimiento con manejo del cambio de siglo.
 - Solicitar al proveedor “parches”, soluciones temporales o definitivas al manejo de fechas y cambio de siglo.
 - Solicitar al proveedor instrucciones para realizar los cambios, o bien asistencia técnica.
 - Solicitar y diseñar especificaciones de prueba para los sistemas, revisar su completitud y ejecutarlas.

- Re-evaluar si existen sistemas propios (“in-house”) que dependan de esos sistemas adquiridos.
 - Mucho cuidado con los plazos que indiquen los proveedores para entregar parches o versiones nuevas. Revisar el historial de cumplimiento de plazos del proveedor.
 - Cuidado con la conversión de archivos y bases de datos.
3. Determinar si el ambiente de operación está preparado (y certificado) ya para manejar correctamente el cambio de siglo. Determinar si el proveedor del ambiente ofrece actualizaciones del software, hardware o firmware para efectuar correctamente el manejo de fechas. Determinar el costo de la migración en cuanto a: el software de operación propiamente, los sistemas y programas propios o adquiridos. Determinar los plazos de entrega para parches, sustitutos, etc. Solicitar las instrucciones para realizar la migración.
 4. Determinar cuán preparados están los sistemas de nuestros *suministradores* de información para brindarnos datos que cumplan con *nuestro* requerimiento de cambio de siglo. Negocie plazos para resolver el cumplimiento, incluya este ítem dentro del plan y dele seguimiento.
 5. Determinar cuán preparados están nuestros sistemas para ofrecer datos que cumplan con los requerimientos de cambio de siglo de nuestros *consumidores* de información. Negocie plazos para resolver el cumplimiento, incluya este ítem dentro del plan y dele seguimiento.
 6. Determinar cuán preparados están los sistemas y dispositivos periféricos para lidiar con el problema del cambio de siglo. Particularmente: cajeros automáticos, lectores de tarjetas de banda magnética en puntos de venta, controladores de dispositivos de seguridad (bóvedas bancarias, control de humedad, temperatura, anti-incendios, etc.).
 - Solicitar al proveedor constancia de su grado de cumplimiento con manejo del cambio de siglo.
 - Si es software: solicitar al proveedor “parches”, soluciones temporales o definitivas al manejo de fechas y cambio de siglo.
 - Si es hardware: solicitar los chips sustitutos (microprocesadores, microcontroladores, ROMs, tableros).
 - Solicitar al proveedor instrucciones para realizar los cambios, o bien asistencia técnica.
 - Es probable que no sea necesario cambiar ni hardware ni software, quizás baste con reiniciar el sistema/dispositivo o cambiar algunos parámetros que lo gobiernan. Estas instrucciones y/o procedimientos deben ser facilitados por el proveedor.
 - Solicitar especificaciones de prueba para los sistemas, revisar su completitud y ejecutarlas.
 - Re-evaluar los sistemas propios que dependan de estos sistemas y dispositivos.
 - Revisar el historial de cumplimiento de plazos del proveedor.
 7. Asignar un puntaje a la probabilidad de falla de cada elemento. Puede usarse una escala numérica del 0 al 10, que comprenda significados como:
 - probabilidad nula de falla (certeza absoluta de cumplimiento o absoluta independencia del manejo de fechas)
 - baja probabilidad de falla
 - mediana probabilidad de falla
 - alta probabilidad de falla
 - certeza absoluta de falla.

Impacto

El impacto o criticidad es un indicador, desde la perspectiva de los usuarios, de la importancia del sistema en cuestión para toda la organización.

Los usuarios pueden reconocer que el sistema maneja fechas y que la funcionalidad de éste depende del correcto manejo de las fechas (o de los datos que se calculan a partir de ellas: intereses, antigüedad de saldos, etc.). No se espera que los usuarios sepan cómo maneja las fechas cada sistema. Se debe, entonces, estimar el impacto operativo de los errores en los datos dependientes de las fechas, sean estos datos generados, recibidos, utilizados o compartidos por el sistema.

Los siguientes son niveles indicativos de impacto:

- Alto. Incapacidad de operar, terminar operaciones o dar respuestas correctas: provocaría detención inmediata de las operaciones directas, permitiría violaciones de la seguridad, causaría daños a vidas humanas, causaría pérdidas financieras, causaría perjuicios materiales, causaría daños ambientales, causaría perjuicios importantes a clientes o proveedores.
- Mediano. El sistema sólo genera información de fechas que es utilizada por personas quienes pueden detectar, interceptar y corregir los errores sin afectar las operaciones de manera importante. Los retrasos del sistema no provocarían fallas o interrupciones de las operaciones.
- Bajo. El sistema no recibe, utiliza o transmite fechas. O bien, existe un sistema sustituto probado y preparado para entrar en operación.

Puede asignarse una cuantificación al grado de impacto. Una escala numérica entre 0 y 10 es suficiente.

Riesgos

Determinar el riesgo de cada sistema o aplicación. Esto se obtiene determinando el impacto (o criticidad) de cada elemento y cuantificando la probabilidad de falla del elemento. Si se usaron escalas numéricas para impacto y probabilidad de falla, el riesgo puede obtenerse mediante la multiplicación de esas dos magnitudes. Si las escalas fueron cualitativas, el riesgo puede obtenerse "al ojo".

Oportunidad de falla

Para cada elemento en el inventario, debe determinarse el momento en que fallaría por su incapacidad (parcial o total) de manejar el cambio de siglo. Es posible que el elemento ya haya fallado. De otra manera, sométale a prueba con fechas como las sugeridas en el Apéndice A.

Plan de acción

En esta sección complementamos lo dicho en la sección *Acciones inmediatas*, resaltando algunas de las particularidades que se presentan en los proyectos de cambio de siglo. Estas consideraciones a tener en cuenta al formular un plan de acción, el cual tendrá los elementos usuales: riesgos, estrategia, responsabilidades, esfuerzo estimado, recursos requeridos, organización de equipos, plazos, desglose de actividades, mecanismos de control, herramientas por utilizar, plan de pruebas, plan de instalación y entrega. En el Apéndice D aparece una lista de cotejo para revisar el plan.

Prioridades

Han de asignarse prioridades a los elementos contenidos en el inventario. Las prioridades serán determinadas por el riesgo de cada elemento y por su momento esperado de falla. Estas prioridades deben ser confirmadas con los directores de las áreas usuarias (consumidoras o suministradoras de información para los sistemas, dependientes de los dispositivos) representados en el Comité de trabajo.

Insistimos: *será necesario establecer prioridades*. Los recursos y el tiempo son escasos. En caso de duda, pregúntese: ¿será capaz mi empresa de seguir operando si el sistema X no maneja adecuadamente las fechas al venir el año 2000?

Estrategias

Para cada aplicación, sistema o dispositivo existe – por lo general – una de cuatro estrategias aplicables: reemplazo, renovación, (re-)desarrollo y retiro. Hay una quinta estrategia, que debe evaluarse *con* conocimiento: no hacer nada, porque *sabemos* que el elemento no lo requiere.

Reemplazo

La estrategia de reemplazar aplicaciones con problemas del año 2000 es únicamente viable en empresas pequeñas, o para aplicaciones con un grado de integración relativamente bajo.

Esta estrategia era mucho más atractiva hace uno o dos años, pero conforme se acerca la fecha no negociable se torna menos atractiva. Esta estrategia es en realidad un camino ya recorrido por mucha organizaciones: el proceso de comprar e instalar paquetes de aplicación. Desafortunadamente, la experiencia en la mayoría de las organizaciones es que siempre se requiere más tiempo del estimado, aún cuando se *sabe* que se requerirá más tiempo del estimado. Si a esto le agregamos el tiempo requerido y la complejidad de escoger entre múltiples paquetes, vemos por qué muy pronto dejará de ser una alternativa viable.

El gran atractivo de esta estrategia es que, al final del proceso, la organización tiene mejores sistemas, en lugar de tener los mismos sistemas antiguos, con todas sus limitaciones (incluida la del manejo de fechas para el cambio de siglo). Sin embargo, al acercarse la fecha es cada vez menos viable pretender matar varios pájaros con una sola pedrada.

En algunas ocasiones, esta alternativa es la única posible (por ejemplo, cuando no se dispone del código fuente, o no se dispone de recursos humanos que conozcan el lenguaje, como PL/I o Algol).

Renovación

Esta es la estrategia del remiendo. Es también una estrategia de potencialmente alto riesgo, pero en muchas ocasiones es la estrategia más viable (por ejemplo en el caso de aplicaciones muy específicas para las cuales no existen paquetes y sí se cuenta con documentación aceptable, el remiendo es mucho más fácil que desarrollar la aplicación de nuevo).

Debe tenerse sumo cuidado de no escoger esta estrategia con base en la disponibilidad del analista o programador que desarrolló la aplicación originalmente. Es muy probable que conforme se acerque la fecha, la competencia por el recurso humano calificado se torne desagradable.

En nuestro medio, dejar las cosas para última hora es casi la norma. La experiencia en otros países nos demuestra que los que empezaron antes son los que tienen mayor preocupación por terminar a tiempo, mientras que los que no han empezado o están empezando, por algún motivo, creen que sí van a terminar a tiempo. Es de esperar que en nuestro medio, al igual que en el resto del mundo, el próximo año, los honorarios del personal calificado para resolver el problema aumenten exorbitantemente. En un ambiente de altos honorarios y pánico generalizado, es sumamente difícil mantener la lealtad del personal.

Las buenas noticias son que en la actualidad existe un gran número de herramientas que permiten buscar y modificar dentro de los programas fuente las ocurrencias del manejo de fechas. Dependiendo del lenguaje y el ambiente en que se encuentren las aplicaciones, la disponibilidad de herramientas puede ser realmente abrumadora (por ejemplo COBOL CICS).

Las malas noticias son que ya no hay tiempo para realizar una selección concienzuda de la herramienta. O se escoge una herramienta casi al azar, o se hace el remiendo a mano (con el editor preferido).

El riesgo asociado con esta alternativa es la incertidumbre de si estará lista a tiempo. No existe experiencia en modificaciones tan extensas. Ya sea que se utilicen las técnicas de ampliar todos los campos que almacenan fechas así como los algoritmos que calculan años bisiestos, o se utilice la técnica de crear una ventana en el tiempo (windowing), la cantidad de programas que hay que inspeccionar y modificar es sumamente grande. Además, la cantidad y rigurosidad de las pruebas por hacer en muchos casos nunca antes se han realizado en la organización. Con esta alternativa es casi seguro que los arreglos estarán listos cuando terminen, y no antes. Esto confirma la necesidad de establecer prioridades entre los sistemas.

(Re-)Desarrollo

La alternativa de volver a desarrollar una aplicación para corregir el almacenamiento y manejo de fechas puede ser sumamente desalentadora. El historial de cumplimiento de plazos en los proyectos de desarrollo e implantación de sistemas es malo, en el mejor de los casos.

Hay, sin embargo, muchos ejemplos de aplicaciones que deberán ser desarrolladas de nuevo para cumplir con los requisitos del cambio de siglo. Aplicaciones que no están disponibles en el mercado, aplicaciones para las cuales no existe el código fuente, aplicaciones que están fallando y requieren de cirugía mayor con carácter de urgencia, etc.

A estas alturas del partido, si se decidiera volver a desarrollar una o más aplicaciones, esto se debería hacer con las herramientas de desarrollo que mejor conoce la organización, pues este es el peor momento para introducir nuevos lenguajes, herramientas y ambientes de software. Este proyecto no puede darse el lujo de costear la curva de aprendizaje de nuevas tecnologías.

Una ventaja de esta alternativa puede ser que teniendo el equipo de gente que desarrolló la aplicación originalmente (o en su defecto, documentación adecuada y actualizada – ¿en nuestro medio?), las estimaciones para el desarrollo pueden ser mucho más acertadas que las estimaciones para una aplicación verdaderamente nueva.

Si se escoge esta vía para una aplicación, asegúrese de dar prioridades a los requerimientos que primero se van a implementar, evite funcionalidades innecesarias y procure hacer un buen modelo de datos que posibilite la implementación de los requerimientos conocidos (aunque para algunos requerimientos menos prioritarios se postergue su implementación).

Retiro

El diagnóstico puede revelar que una aplicación ya no tiene utilidad o que será en breve reemplazada por una que está adquiriéndose o desarrollándose. La estrategia puede contemplar el retiro planificado de la aplicación obsoleta.

Definición de estrategia

Uno puede plantearse este proceso de decisión para definir la estrategia de reparación:

```
if existe un paquete que hace la misma función
then if hay tiempo de implementar
      then Reemplazar
      else Renovar
else if hay tiempo de reescribir
      then Re-desarrollar (reescribir)
      else Renovar
```

[Martin 1997] recomienda seleccionar enfoques de solución para cada tipo y mezcla de sistemas, balanceándolos contra las restricciones. Así se pueden obtener estimaciones realistas de costos, sobre la base del diagnóstico y las soluciones propuestas. Pregúntese:

- ¿Cuántos sistemas son afectados? ¿Cuántos sistemas requieren ser reparados, probados e implantados sincronizadamente?
- ¿Cuál es el costo del cambio, en términos de esfuerzo, almacenamiento, procesador, licencias de software, migraciones de datos, pruebas y capacidades que se postergan o difieren?
- ¿Cuán rápidamente puede implantarse la solución?
- ¿Cuál es el período operativo de los sistemas? (P.ej. año calendario, año fiscal, zafras, etc.)
- ¿Cuál es el costo de no cambiar (dejar las cosas como están), en términos de responsabilidades, funcionamiento defectuoso y errores?

Recursos

La gente es el más importante de todos los recursos aplicables a la solución del problema del año 2000. Como se ha indicado arriba, es vital contar con un Director de Proyecto competente y experimentado (obviamente, esta persona quizás no tenga experiencia pertinente en *este* problema, pero sí la tendrá en *otros* problemas). También es esencial contar con analistas y programadores competentes, así como otras personas que realicen las pruebas de manera rigurosa. Durante el diagnóstico y las pruebas de aceptación es fundamental contar con el criterio de los usuarios.

No debe subestimarse la necesidad de contar con un ambiente de pruebas que sea equiparable al ambiente normal de producción. Si no es posible adquirirlo, deberán buscarse otras opciones: usar horarios alternos, interrumpir la operación normal por períodos definidos, alquilar tiempo de computadora, definir un ambiente operativo virtual (en sistemas operativos con máquinas virtuales esto es posible), etc.

Los otros recursos importantes son las herramientas de software que se han desarrollado para enfrentar este problema. Hay una gran variedad de herramientas, pero no muchas de ellas son realmente confiables. En el Apéndice F se listan algunas de estas herramientas, así como otros recursos informativos disponibles en Internet.

Tablero de control (“pizarra marcador”)

En [Martin 1997] se presenta un tablero de control semejante a una “pizarra marcador”. El tablero fue desarrollado por Mitre para facilitar el control global de los proyectos de preparación para el cambio de siglo. El tablero ayuda a los directores y administradores en los siguientes aspectos:

- dar seguimiento al riesgo durante el proceso de resolución,
- compenetrarse con el estado de los recursos a medida que procede la resolución, e
- identificar sistemas que son críticos.

El tablero debe llenarse desde que se hace el diagnóstico y se formula el plan de acción. El tablero será actualizado periódicamente para reflejar los cambios en el estado del proyecto, facilitando la atención a la evolución de los riesgos, el control y la dirección del proyecto. Ponga en cada fila un elemento al cual se le da seguimiento respecto del cumplimiento con el cambio de siglo: sistemas propios, sistemas adquiridos, sistemas de reemplazo, relaciones con suministradores y consumidores de información, dispositivos electromecánicos, etc.

En el Anexo E se dan más detalles sobre el tablero de control. Se incluye un tablero que muestra dos aplicaciones: una crítica y una no crítica. Los interesados en obtener un machote del tablero en Excel, pueden enviar un mensaje a: itrejos@mimas.cic.itcr.ac.cr.

Ejecución

Una vez elaborado el plan de trabajo, y definida la estrategia de certificación de cada aplicación (comprar una nueva, arreglar la actual, desarrollar nuevamente o retirar la actual) se deberá, de una manera ordenada proceder a certificación de cada una de las aplicaciones.

En instalaciones con aplicaciones relativamente nuevas (cinco años o menos) es bastante probable que la aplicaciones no requieran mayores ajustes (tal vez solo para el 29 de febrero 2000). Sin embargo, sería altamente irresponsable no realizar las pruebas, ya que a pesar de contar con las mejores herramientas, bibliotecas, estándares, y diseños, en última instancia se depende del estado de ánimo del programador el día en que escribió (o remendó) cada programa.

El sentimiento de “gasto improductivo” es uno de los grandes enemigos de los proyectos del año 2000. La sensación de que todo el tiempo, esfuerzo y dinero – que se debe invertir en este proyecto – no agrega valor a la empresa, que sale directamente de las utilidades y que además consume recursos que podrían haberse dedicado a proyectos que sí agregan valor, no debe nublar la razón de quienes toman decisiones.

No es del todo cierto que el proyecto del año 2000 no le vaya a agregar ningún valor a la organización. **Existe un beneficio importante**, el cual puede verse como un subproducto del proyecto, que consiste en el conocimiento detallado que adquiere la organización respecto de su situación tanto de hardware como de software. Hoy en día, en nuestro medio, la gran mayoría de las organizaciones no tienen ni la menor idea de qué software tienen o en qué estado se encuentra. Este conocimiento es vital para la planificación y el desarrollo tecnológico de cualquier organización.

Pruebas

La experiencia en países menos desorganizados (donde las cosas no se dejan para última hora) nos muestra que las pruebas de los sistemas representan más del 50% de todo el esfuerzo requerido por el proyecto del año 2000.

En nuestro medio es muy probable que el porcentaje sea mayor debido, primero, a que la mayoría de las organizaciones no cuentan con un ambiente de pruebas y, segundo, a que en la mayoría de las organizaciones no existe la cultura, de parte del usuario, de realizar pruebas exhaustivas.

Existen dos tipos de pruebas que podrían realizar las organizaciones. Primero están las pruebas que se realizan con la convicción (o la esperanza) de que todo esté bien, el ejemplo típico es la organización que tiene sistemas relativamente nuevos, con fechas completas y hasta con Diccionario de Datos. Aún en los casos en los que se tienen motivos válidos para esperar que las aplicaciones operen correctamente ante el cambio de siglo, es necesario hacer las pruebas que corroboren las sospechas. No hacer las pruebas sería irresponsable. En segundo tipo de pruebas son aquellas que se deben realizar luego de modificar las aplicaciones actuales o adquirir aplicaciones nuevas. En este segundo tipo de pruebas, se debe no solo probar el funcionamiento correcto ante el cambio de siglo sino comprobar también el funcionamiento correcto de toda la funcionalidad de las aplicaciones (esto es, son pruebas de regresión).

Ambiente

El primer paso para poder realizar las pruebas necesarias es el establecimiento de un ambiente de hardware y software que sea una fiel réplica del ambiente de producción. Este ambiente debe contar con todas las mismas versiones del hardware y el software. El ambiente debe además contener una base de datos reales (aunque no necesariamente al día), así como un conjunto de transacciones representativas de la actividad de la organización.

Alcance

El alcance de las pruebas, como se mencionó antes, depende de si estas se realizan antes o después de efectuar modificaciones. Estrictamente hablando, la responsabilidad del proyecto del año 2000 debería ser únicamente probar la funcionalidad relacionada con el cambio de siglo. En el caso de aplicaciones nuevas o aplicaciones modificadas, se deberá, obviamente, probar toda la funcionalidad de todas las aplicaciones.

La funcionalidad relacionada con el cambio de siglo variará de una organización a otra, sin embargo en todos los casos se deberá comprobar el funcionamiento correcto de las aplicaciones en el 9 de setiembre de 1999, el 31 de diciembre de 1999, el 1 de enero del 2000 y el 29 de febrero del 2000.

Adicionalmente, algunas organizaciones, dependiendo de su actividad, deberán comprobar que una transacción realizada en diciembre con fecha valor en enero funcione correctamente, o una transacción realizada en enero con fecha valor en diciembre también funcione correctamente. Dependiendo de la organización, el 31 de enero podría ser una fecha clave, o el 31 de marzo (fin del trimestre).

La lista de fechas claves de cada organización deberá ser decidida de antemano por cada organización, tomando en cuenta la naturaleza de sus actividades y las fechas críticas respectivas. Es también recomendable revisar la lista de fechas claves utilizadas por organizaciones con actividades similares en otros países. En el Apéndice A presentamos una lista de fechas claves utilizadas por varias compañías a nivel mundial.

Aceptación

En todas las organizaciones, las pruebas deben ser aceptadas por el usuario. El proceso de aceptación debe ser formalizado, es decir, se debe conocer el resultado de las pruebas antes de realizarlas. Debe, además estar muy clara la responsabilidad de aquellos que estarán aceptando las aplicaciones.

Este proceso de pruebas y aceptación es lo más cercano a la “certificación” que es posible tener (dado lo avanzado de la fecha). Es importante que quede muy claro entre los participantes el efecto y el impacto que tendría en la organización el “certificar” aplicaciones que luego fallen en el cambio de siglo.

Conclusiones

Dado que el plazo es inamovible para estos proyectos, el tiempo es – literalmente – precioso. No es admisible *perder* tiempo. Tómese en cuenta lo siguiente:

- Reducir al mínimo el tiempo de concientización. A estas alturas la mayoría de las organizaciones deberían saber que hay un problema informático potencial ante el cambio de siglo. Si aún no hay consciencia en la organización, es urgente dar charlas con los directivos y conviene considerar el contratar a un conferencista externo para ayudar en esta fase.
- Toda organización debe definir qué entiende por “cumplir con el manejo del año 2000”.
- Escoger al mejor director de proyectos (no necesariamente un informático) como el director del proyecto de cambio de siglo.
- El proyecto del año 2000 *no es sexy*, principalmente para los informáticos. Sin embargo, si su equipo lo realiza con éxito habrá demostrado una excelente capacidad para realizar proyectos complejos con severas restricciones. El proyecto del año 2000 puede ser una excelente escuela para futuros directores de proyecto y representa un importante desafío ingenieril.
- El proyecto requiere de una planificación seria y de entrar en acción cuanto antes. No se debe postergar el inicio de la ejecución en aras de un plan perfecto; los planes perfectos no existen y en este caso la perfección en el plan es indeseable. Ser pragmático es primordial. El plan debe ser suficiente para proceder de manera ordenada y controlable. Es vital tener buenas estructuras de retroalimentación y la capacidad de responder rápidamente para ajustar el curso del proyecto.
- La participación de las áreas usuarias será determinante para el éxito del proyecto. El conocimiento funcional de las aplicaciones está en los usuarios.
- Habrá necesidad de establecer prioridades: entre sistemas y dentro de cada sistema.
- Habrá necesidad de diferir el desarrollo de nuevos sistemas o nueva funcionalidad: ante recursos limitados, la corrección del problema del año 2000 no se puede postergar.

- Si se escoge re-desarrollar una aplicación existente: utilice las herramientas mejor conocidas en la organización, evite implementar requerimientos innecesarios y concéntrese en implementar los requerimientos prioritarios. Aproveche la oportunidad, eso sí, para diseñar un modelo de datos que permita la implementación (futura) de todos los requerimientos.
- No hay tiempo ya para hacer un estudio exhaustivo de herramientas aplicables a la situación de la organización. Hay una plétora de herramientas en el mercado; de ellas, únicamente unas 10 herramientas son realmente útiles. La mayoría de las herramientas disponibles son para dialectos de COBOL.
- Los proyectos en ejecución (nuevos desarrollo, reemplazos) ya deberían tener entre sus requerimientos el manejar adecuadamente el cambio de siglo.
- Haga un respaldo completo de *todo*: datos, programas, *scripts*, *jobs*, software de base, software de oficina, etc.
- Enfrentar el problema del año 2000 reportará grandes beneficios a las organizaciones: conocerán mejor cuáles son sus aplicaciones, bases de datos y dispositivos; estarán en mejor capacidad de planificar la aplicación estratégica de la tecnología de información en el futuro; tendrán capacidad para enfrentar más exitosamente la evolución y mantenimiento futuros de sus sistemas.

Apéndice A: Algunas fechas críticas

A continuación se listan algunas fechas que pueden provocar problemas:

Fecha	Suceso
3000.01.01	No es un año bisiesto
2100.01.01	No es un año bisiesto
1998.01.01	Año "bandera" 98
1999.01.01	Año "bandera" 99
1999.09.09	Número mágico 9/9/99
1999.09.30	Cierre fiscal 1999
2000.01.01	Desborde de años de dos dígitos
2000.1.10	Primera fecha de 9 caracteres
2000.02.29	Año bisiesto
2000.03.31	Primer trimestre del 2000
2000.04.01	Primer trimestre del 2000
2000.09.30	Cierre fiscal 2000
2000.10.10	Primera fecha de 10 caracteres
2000.12.31	Día 366 del año
2001.01.01	Siglo 21 (según algunos)

Apéndice B: Patrones para localización de fechas

Los patrones y elementos que aparecen a continuación son muy dependientes de los lenguajes de programación, pero son indicativos del tipo de búsqueda por efectuar y pueden dar *indicios* para facilitar la localización de datos-fecha.

1. Declaraciones de tipo o dimensión:
 - PIC 99999,
 - PIC 9(6),
 - PIC 99,
 - DATE.
2. Literales en programas (en formatos de entrada o salida, etiquetas):
 - AAMMDD, MMDDAA, DDMMAA,
 - “19”, 19,
 - “###/###/###”, “###/###/19##”,
 - “30”, “60”, “90”, “120”,
 - 30, 60, 90, 120.
3. Literales para inicializar variables o constantes, también para formatos:
 - “AAMMDD”, “DDMMAA”, “MMDDAA”,
 - “AA/MM/DD”, “DD/MM/AA”, “MM/DD/AA”,
 - “AA-MM-DD”, “DD-MM-AA”, “MM-DD-AA”,
 - “19AAMMDD”, “DDMM19AA”, “MMDD19AA”,
 - “19AA/MM/DD”, “DD/MM/19AA”, “MM/DD/19AA”,
 - “19AA-MM-DD”, “DD-MM-19AA”, “MM-DD-19AA”,
 - “###/###/###”, “###/###/19##”,
 - 19, “19”,
 - “ANO”, “AÑO”, “A#O”, “DIA”, “MES”, “FECHA”, “PLAZO”, “VENCIMIENTO”,
 - 12, “12”, 29, “29”, 28, “28”.
4. Nombres de variables (ANO, ANIO, ANNO, DIA, MES, etc., como prefijos, sufijos).
5. Variables que aparecen “cerca” de una literal o constante² sospechosa, en formas o pantallas de presentación/entrada de datos. Es típico que sea algo como “Fecha de formalización (AA/MM/DD) --/--/--”, donde la variable que recibirá los datos aparece a la derecha de un rótulo, o inmediatamente debajo de este.
6. Variables que aparecen en reportes: típicamente hay varias líneas de detalle bajo un rótulo. La idea es correlacionar el encabezado de columna con la variable de la línea de detalle que aparece en la columna.
7. Invocaciones a rutinas de fecha del sistema:
 - TODAYS-DATE,
 - CALL SYSTEM-DATE

² Entendamos *literal* como un valor de un tipo de datos que puedo expresar como tal, mientras que una *constante* es un valor que cuenta con un nombre (identificador) con cierto alcance dentro del programa.

Apéndice C: Ejemplos de código

Las siguientes líneas contienen código (hipotético pero razonablemente típico) que podría verse afectado por el cambio de siglo:

- SUBTRACT fecha_contratacion FROM TODAYS-DATE GIVING tiempo_empleo.
- LET monto_cobro = tarifa * (fin_llamada_telef - inic_llamada_telef + 1)
- if (fecha_factura < fecha_orden) error("Factura Mala")
- IF anno = 99 THEN PERFORM Borrar-Registros-Movimientos.
- if backup_date > file_last_change_date then file_overwrite_okay :=TRUE
- if (dias-transcurridos (fecha_hoy - fecha_ult_servicio)) > periodo_operativo then maquina.off()
- IF edad (fecha_actual,fecha_nacimiento) > 65 THEN Autorice_pension ELSE Niegue_pension

Apéndice D: Revisión del plan

A continuación aparece una lista de cotejo que busca facilitar la revisión de un plan de acción para enfrentar el año 2000.

Para cada sistema

- ¿Se han enumerado los principales riesgos identificados? ¿Se han enunciado acciones para mitigar o evitar los riesgos identificados? ¿Se han examinado críticamente los riesgos y los mecanismos de mitigación?
- ¿Se ha considerado adecuadamente la estrategia recomendada, así como los papeles (“roles”) y responsabilidades que implica la estrategia?
- ¿Es la estrategia recomendada congruente con los riesgos identificados y evaluados?
- ¿Se ha hecho una estimación del esfuerzo requerido para reparar el sistema (recursos, tiempo, costos)?
- ¿Se han identificado las actividades requeridas para reparar el sistema?
- ¿Son razonables y realistas los tiempos estimados para cada actividad?
- ¿Se han identificado los recursos humanos, materiales, tecnológicos y financieros requeridos para reparar el sistema?

Para el plan en conjunto

- ¿Se cuenta con la aprobación y el apoyo comprometido de los directivos y las gerencias?
- ¿Están claras las prioridades entre aplicaciones? ¿Se han establecido las prioridades de manera objetiva, mediante un análisis de riesgos, oportunidades de falla y dependencias entre aplicaciones? ¿Están aceptadas las prioridades por el Comité de trabajo y las áreas usuarias?
- ¿Se valora la disponibilidad de recursos por los cuales se podría competir?
- ¿Se valora el conflicto del plan con otros proyectos de la organización?
- ¿Se incluyen actividades explícitas para: sistemas propios, sistemas adquiridos, sistemas de reemplazo, dispositivos, relaciones con suministradores o consumidores de información?
- ¿Se han calendarizado adecuadamente las actividades? ¿Es posible paralelizar actividades (en caso de disponer de recursos suficientes)?
- ¿Se han definido los responsables de ejecutar cada actividad?
- ¿Se ha negociado y programado con las áreas usuarias la dedicación de su personal a tareas/actividades del proyecto, particularmente las relacionadas con las pruebas de aceptación?
- ¿Se han incluido actividades explícitas para integrar los sistemas a la producción?
- ¿Ha sido validado el plan con otros líderes de proyectos o consultores externos? ¿Ha sido valorado el plan con los posibles miembros del equipo del proyecto?

Apéndice E: Tablero de control (pizarra marcador)

El tablero de control diseñado por Mitre representa los principales factores de riesgo que se usan para determinar cuáles sistemas requieren financiamiento, recursos adicionales (obviamente, el tiempo *no* es uno de ellos) u otra forma de intervención administrativa para reducir el riesgo.

Cada fila en el tablero representa el estado (status) de un sistema. Se utiliza una codificación con colores con el siguiente significado:

Rojo (R)	Requiere atención inmediata
Amarillo (A)	Tiene un riesgo moderado. Debe dársele seguimiento.
Verde (V)	El proceso de reparación procede sin contratiempos. No se anticipa que requiera intervención administrativa.
Azul (Z)	Se han resuelto los problemas satisfactoriamente o se ha completado una actividad.
Blanco	(En el área de estado de implementación) Actividad no iniciada.

Áreas del tablero

Resumen del riesgo

Esta área permite identificar los sistemas que requieren atención administrativa inmediata. Los sistemas de más alta prioridad tienen alta criticidad o alto riesgo de solución. El *riesgo de solución* es un indicador que resume el riesgo de que no se puedan resolver problemas del cambio de siglo antes de que se manifieste un error. Es el máximo de los niveles de riesgo señalados en el área de *fuentes de riesgo*.

- Se usará el rojo cuando se ha establecido que un sistema es de alta criticidad y no se ha evaluado el detalle de su estrategia de solución.
- Se usará el azul cuando es seguro que un sistema sustituto entrará en operación antes de la oportunidad de falla del sistema.
- El riesgo de solución debe ser re-evaluado cuando cambien los niveles de riesgo de las fuentes de riesgo y el estado de la implementación.

Fuentes de riesgo

Hay tres principales fuentes de riesgo: dificultad técnica, escasez o carencia de recursos, y tiempo insuficiente.

- **Riesgo técnico:** indica la dificultad intrínseca de resolver el problema del cambio de siglo para un sistema. Puede originarse en: tamaño del sistema, número de interfaces con otros sistemas, carencia de documentación adecuada, carencia de programas fuente actualizados, inadecuadas técnicas de programación empleadas en el sistema, etc. Si es alto requerirá un seguimiento cuidadoso.

- **Riesgos de recursos:** representan deficiencias en los recursos disponibles (humanos, financieros, medios, contratos externos).
 - ◆ Rojo: recursos no disponibles.
 - ◆ Amarillo: recursos disponibles pero probablemente inadecuados, o no disponibles para completar el trabajo.
 - ◆ Verde: recursos adecuados para completar el trabajo.
 - ◆ Azul: no hay necesidad de recursos adicionales.
- **Riesgo de tiempo:** indica que el proceso de resolución, incluyendo la validación de la reparación, no podría completarse antes de que ocurra un error relacionado con el manejo de fechas.
 - ◆ Rojo: no se conoce el tiempo requerido porque no se ha hecho una valoración, o se conocen solo soluciones que tomarán demasiado tiempo.
 - ◆ Amarillo: hay retraso respecto del cronograma pero puede recuperarse, o bien hay dudas de si habrá tiempo de concluir y validar los cambios.
 - ◆ Verde: hay avance sin contratiempos y es improbable que se deje de cumplir los plazos.
 - ◆ Azul: se ha concluido la resolución del problema de cambio de siglo

Estado de implementación

En esta área se muestra el estado de implementación de las actividades requeridas para completar la solución al problema del cambio de siglo. Las actividades son: inventario, evaluación, reparación, validación (pruebas) e implementación.

La clave para interpretar los colores en cada actividad es:

- Blanco: actividad no iniciada.
- Rojo: actividad malograda. La actividad no dio resultados exitosos, descubre más problemas o no se concluye de acuerdo con el cronograma.
- Amarillo: hay retraso respecto del cronograma, o es probable que revele más problemas.
- Verde: hay avance sin contratiempos, no se requiere intervención administrativa.
- Azul: se ha concluido la actividad exitosamente.

Finalmente, la *fecha de conclusión* indica el plazo más tardío en que el sistema debe estar listo para ser instalado. Esta fecha indica cuánto tiempo queda para la reparación.

Apéndice F: Recursos

Herramientas

A continuación listamos algunas herramientas aplicables al problema de cambio de siglo [Zvengitzov 1997].

- **Rescue, Rescue Studio, DB-Convert.** ArtInSoft. www.artinsoft.com. Herramientas para la reingeniería automática de software y para la migración de datos. Windows, Windows 95, Windows NT. Informix 4GL, COBOL, Business BASIC.
- **CA-Inventory/2000.** Computer Associates International. www.cai.com. Examina bibliotecas fuente y objeto a fin de categorizar y asociar elementos fuente y objeto. MVS. Varios lenguajes.
- **RPG Source Retrieval.** Advanced Systems Concepts. www.advsyscon.com. Ayuda a determinar características suficientes para identificar el programa fuente original y los datos que manipula. AS/400. RPG.
- **Analyzer 2000.** Ironsoft. www.execpc.com/~ironsoft/. Herramienta para ayudar en el inventario de aplicaciones, basada en patrones. MVS, OS/2. COBOL, RPG, JCL.
- **CA-Impact/2000.** Computer Associates International. www.cai.com. Herramienta más avanzada para inventario y parcialmente útil para análisis de impacto. MVS, VSE, Windows. COBOL, DL/I, SQL, CA-Datacom, CA-IDMS.
- **Cayenne 2000.** Cayenne Software. www.cadre.com. Encuentra campos fecha potenciales, parcialmente útil para análisis de impacto. Windows, Windows 95. COBOL.
- **Century file conversion.** Quintic Systems. www.quintic.com. Conversión de archivos e inventario de campos fecha en código fuente. MVS, Tandem, VSE. COBOL.
- **Giles.** Global Software. www.globsoft.com. Construye diccionario de datos de una aplicación, para localización y análisis de fechas. MVS. COBOL, CSP, Focus, JCL, Natural, PL/I, SQL, CA-Easytrieve.
- **Manage2000.** Intersolv. www.intersolv.com. Ambiente de trabajo que coopera con CA-Realia II Workbench y Micro Focus COBOL Workbench. Varios sistemas operativos. COBOL, CICS, DB2, IMS, JCL, MFS, VSAM.
- **McCabe Visual Reengineering Toolset.** McCabe & Associates. www.mccabe.com. Varias herramientas para análisis estático y dinámico de código, así como para pruebas. Varios ambientes. Varios lenguajes.
- **Micro Focus Challenge 2000, COBOL Workbench, Revolve.** Micro Focus. www.microfocus.com. Inventario y análisis de fechas; basado en patrones. MS-DOS, Windows, OS/2. COBOL, SQL, DB2.
- **HourGlass 2000.** Mainware. softmall.ibm.com:80/mainware/. Permite trabajar con fechas y horas simuladas. MVS. COBOL, 4GLs, CICS, IMS.
- **Comudas.** IBM. www.ibm.com. Biblioteca de fechas. MVS. COBOL, PL/I.
- **TransCentury Calendar Routines.** Platinum Technology. www.platinum.com. Biblioteca fuente para manejo de fechas; está globalizada. COBOL.

Información

- **KPMG Peat Marwick.** www.kpmg.com. En Costa Rica: www.kpmg.co.cr
- **Computer World.** www.computerworld.com
- **Info World.** www.infoworld.com
- **Lista de cotejo Y2K.** www.mitre.org/research/cots/COMPLIANCE_CHECKLIST.doc
- **Catálogo de herramientas.** www.mitre.org/research/y2k/docs/TOOLS_CAT.html
- **Catálogo de paquetes.** www.mitre.org/research/cots/COMPLIANCE_CAT.html
- **Cuestionario Y2k.** www.mitre.org/research/cots/Y2K_QUESTION.html
- **BIOS que cumplen Y2K.** www.mitre.org/research/cots/COMPLIANT_BIOS.html
- **Peter de Jager.** www.year2000.com/cgi-bin/clock.cgi
- **IBM Y2K.** www.s390.ibm.com/stories/tran2000.html
- **Information Systems Audit and Control Association.** www.isaca.org/yr2000.htm
- **Information Technology Association of America.** www.ita.org
- **Peritus Technology.** www.peritus.com
- **The Software Technology Support Center.** www.stsc.hill.af.mil/~red/index.html

Bibliografía

- [Jones 1996] Jones, Capers. The economic impact of the Year-2000 software problem in the United States. IT Metrics Strategies. Vol. II, No. 6, junio 1996.
- [Jones 1997a] Jones, Capers. The global economic impact of the Year 2000 software problem. <http://www.spr.com>
- [Jones 1997b] Jones, Capers. Slow response to Year 2000 problem (entrevista). IEEE Software. Vol. 14, No. 3, mayo/junio 1997.
- [Martin 1997] Martin, Robert. Dealing with dates: solutions for the year 2000. IEEE Computer, marzo 1997, pp. 44-51.
- [Sanabria 1997] Sanabria, Iván. Señor industrial, ¿Está usted preparado para el año 2000? Computer World América Central, noviembre 1997.
- [Zvengitzov 1997] Zvengintzov, Nicholas. A resource guide to Year 2000 tools. IEEE Computer, marzo 1997, pp. 58-63.

