



**ECONOMIC AND SOCIAL
COUNCIL**

Distr.
LIMITED
E/ESCWA/TECH/1999/WG.2/10
6 September 1999
ORIGINAL: ENGLISH

Economic and Social Commission for Western Asia

Expert Group Meeting on Project Planning
and Management in Research and
Development and Quality Assurance
Beirut, 21-23 September 1999

ECONOMIC COMMISSION
FOR WESTERN ASIA
29 SEP 1999
LIBRARY + DOCUMENT SECTION

YEAR 2000 PROJECT MANAGEMENT

Note: This document has been reproduced in the form in which it was received, without formal editing. The opinions expressed are those of the author and do not necessarily reflect the views of ESCWA.

Year 2000 Project Management

Frank R. Parth

Introduction

Year 2000 project management is probably the most unusual type of project management we have ever seen. Unlike managing the delivery of a specific product within cost and schedule constraints, Y2K project management consists of setting in place a number of processes in parallel, all oriented towards making the company compliant by an absolutely rigid date. There is no one specific set of activities, Y2K PM instead is a collection of activities oriented towards bringing the company into a compliant state. For the first time in project management, we have a project that literally affects every part of the company, from its development capabilities, to its financial systems, its facilities, advertising, legal department, production capabilities, and all up and down the supply chain. No part of the company is left untouched. This forms special challenges in successfully completing the project. Because it is so unique, there is much that can be learned from the techniques and methodologies that are being developed.

The basic problem

There are three causes for the Year 2000 problem that we are facing now. The beginnings of the problem lie in the beginnings of mainframe processing. Prior to the availability of magnetic storage media all data was stored on punch cards, 80 columns wide. With only 80 columns to work with if the programmer could save two columns by compressing the date he did. This was accepted "best practices" back then.

Programs evolved faster than the hardware did. As programs were changed the changes were written using the same date format in order to stay compatible with existing code. Eventually storage hardware began to catch up with the storage demands. But it has only been very recently in the history of computers that storage has become both large in size and affordable in price. Even as far back as the late 1980's, hard disks were relatively tiny and still expensive. Since the explosion in storage capacity during the past decade the reasons for saving a minimal two date fields has disappeared. But by this point the two-digit year field has become so standardized in millions of commercial and proprietary programs that until now there has been no incentive to change it.

The second cause of the problem is that since a full four digit year is necessary for some processing, the century date was built into the hardware chips themselves rather than being coded in software. This led us to the problem that currently exists with embedded chips and with Intel-based PC's. The overwhelming majority of PC systems made prior to 1997, and many since then, have 1900 burned into a chip called the Real Time Clock (RTC). The output of the RTC is read by the Basic Input/Output System (BIOS) and fed to the operating system and to the application programs. This is true for all architectures based on the Intel platform.

The third and final cause of the problem is our attempts to be efficient. When typing dates in spreadsheets and database applications, it is much easier to type a two digit year than a four digit year. The developers of these programs designed the programs so that it would recognize a two digit year and add 1900 to it, correctly representing a date in the 20th Century. This resulted in many programs having 1900 built into them.

These three causes created the Y2K problem as we face it now. The end result is that there is almost nothing that depends on technology that does not potentially have a Y2K problem.

Normally, doing things in parallel creates a lot of risk in the project and project managers don't like to do this. However, the Y2K problem was so widespread in an organization that we learned that one of the first things to do is to set up a high-level project office to manage the overall problem for the company, then set up parallel efforts in the areas of:

- Triage
- Inventory
- Vendor, supplier, and business relationship management
- Technology fixes: testing, fixing, retesting and integrating
- Risk identification and management
- Communications

The size of the Y2K problem caused a lot of project management problems. At the beginning of the problem it was easy to say that much of the problem lay in the computer software. However, most large American and European companies have millions of lines of computer code written in dozens of different languages. Just identifying which code might have problems and which didn't took a lot of personnel and the creation of a lot of software tools. Once we understood that the problem really affected the entire organization, the scope of the management effort increased dramatically.

As we learned more about the potential impacts of the problem we realized that it was more logical to consider it a business problem rather than a technology problem. A modern business does not exist in isolation. It requires other companies to supply it with materials, it requires utilities (power, water, communications, gas), it requires a distribution system to move its finished products to customers and to bring it the raw materials, it requires a financial system to manage money. A business exists inside a network of supporting businesses. If any one of those other companies fail to survive the transition to the year 2000, the business itself is in trouble. Project management expanded outside the walls of the business to include assessing the viability of the company's vendors, suppliers, business relationships, and utility infrastructure. This was a radical departure from what most project managers were used to dealing with, but it was necessary for success.

Risk Management

Three of the most difficult challenges in project management are:

- Accurately identifying the duration of tasks
- Managing the scope of the project
- Correctly identifying and managing the risks.

The Year 2000 problem was especially difficult in all of these areas.

It is an unfortunate aspect of life for project managers that we make our promises of how long the project will take and how much it will cost at the very beginning, when we know the least about the project. Only once we're far into the project can we accurately estimate the final cost and duration. The most critical part of this is correctly identifying task durations. If we have a small-medium project with 100 tasks on it, and we underestimate each task by 5%, we will significantly exceed the project's deadline and cost. The Y2K problem was worse than most projects in this respect. We truly did not know how to predict what parts of the system would have a problem and how much effort it would take to fix. We learned to build milestones into the project at which point we would have enough information to estimate the next phase of the project and obtain approval for it.

Project managers are used to dealing with risk in two dimensions: technological risk (what are the odds that we can actually produce this) and programmatic risk (how often is the company successful when it tries to do projects like this). The Y2K problem taught us another dimension of risk - organizational risk.

Since the Y2K problem affected virtually everything inside the company, any part of the company could cause project risk by not taking the problem seriously enough. This was written into many risk management plans as an area to watch. Establishing an internal communications group to help people understand the problem and how important it was became critical to many projects success.

In addition to risk from inside the organization, there was significant risk to a company's successful Y2K project from outside the company. A company could become completely Y2K compliant, and yet still not survive the transition to the year 2000 if the electrical power went out for two months or if the company's critical suppliers failed to survive. The size and scope of risk management plans grew extensively during this effort and taught us that there is far more risk to projects besides just the internal issues.

The Project Management Problems

The Y2K problem was a unique project management problem in many ways. For the first time ever in project management, there was no possibility of moving the end date. Project managers are used to keeping the requirements, the resources, and the schedule in balance. With the Y2K problem we have no control over the deadline date. There is no possibility of moving January 1st, 2000, in order to complete the project. This was a challenge we had never faced before.

A second unique aspect of it is that because computers are so common around the world, there is very little in the industrial world that is not affected by it:

- Computer hardware
- Computer software/operating systems/application programs/databases
- Networks
- Communications systems
- Power generating systems
- Distribution systems
- Manufacturing plants
- Banking and financial systems
- Health care and hospitals
- Etc.

A third unique aspect is the size of the management problem. The Y2K problem was originally thought to be limited to the computer systems. However, virtually everything in a modern company is managed by its computer systems. We learned very quickly that this was not a technology problem as much as it was a business problem. This put a severe stress on our traditional project management techniques because they were not designed to manage such a wide scope of effort.

Project Management of the Year 2000 Problem

Project managers learn best from other project managers. Most of this learning takes place through conferences, education courses, books, and published articles. In the case of the Y2K problem, books and articles were too slow to spread knowledge as quickly as it needed to be spread. Even conferences take months to plan and coordinate. Fortunately, there was a solution to this communications problem available - the Internet. Many locations and discussion groups were quickly set up and became critical communications rivers for passing information out and for sharing experiences. As quickly as the problem developed the ability to share information quickly became critical.

The hard deadline date proved to be very a very challenging problem. Since there was no possibility of moving it, we learned to (a) prioritize what was truly critical for an organization to operate and what could be put off until later, and (b) to do everything in parallel.

Prioritization allowed us to focus the limited resources on those areas that were necessary for the organization to survive rather than attack all of the problems during the limited time period. In order to do this, we copied a technique used in the medical field called triage. One of the first things a project manager did was to divide the Y2K issues into four categories:

- Mission critical - necessary for the company to keep operating after the year 2000,
- Severe - the problem was critical, but we could work around it by changing procedures or finding some other way to do it,
- Sustainable - there would be a problem, but we could live with it until we could fix it later,
- No problem - this area could be ignored.