Computer History Museum

Evolutionary Events in Core Business Information Systems

By Bruce Peterson

Written: February, 2007

CHM Reference number: R0368.2017

Author Sketch: State Univ. Long Beach, CA

1963 – 1998: USAF, Occidental Life, Hughes Aircraft, CSC

1998 – 2003: Independent Consultant

2003: Retired

Abstract: Bruce Peterson describes his long career in data processing starting as an operator of punched card equipment through becoming a programmer and then a data processing manager. He describes how he faced and solved some major challenges as a manager at Hughes Aircraft as the volume of data being processed began to overwhelm the capacity of their computer system.

Introduction

The year is 1963 and my name is Bruce Peterson. These are the “Wild West days of Data Processing”…no rules, no playbook, and no formal education required to participate! And yes, I said *Data Processing,* not Information Technology or some other esoteric term representing the evolution of business computing into some sort of science. That’s because in 1963, what we did was churn data.

I might have had a stellar career as a grease monkey except for my mother who said that she read that data processing was the future. I had just graduated from Sawyer Business School where I learned the fundamentals of data processing, i.e., how to wire a punched card accounting machine plugboard and how to run a card sorter. I was a newly minted “Operator” at Occidental Life in downtown Los Angeles. A side benefit to working in the data processing area was going to the only air-conditioned room in the building which held an IBM 650. Another benefit of the heat generated by the 650 was to warm your lunch on top of the computer hardware.

I worked for a few months and was then called by my Army Reserve unit to boot camp at Fort Polk, LA. During those 6 months, I had a career changing experience. One of my drill instructors told us that he had only completed the 4th grade but look how smart he was. I vowed that he would not become my role model.

Back home again, there was a position available for a “Computer Operator” on the graveyard shift. I sprung at the chance and before long was promoted to supervisor.

At the same time I began to search out computing and programming classes as I aspired to a greater career goal than computer operator. At that time, the only classes in my locale were taught evenings by daytime practitioners at 2 -ear community colleges. During my formal education, which took 16 years at night, I earned an AA, BS and MBA. The AA degree was the 1st granted by Santa Monica City College, as at that time, they had no formal data processing curriculum.

First Programming Job

My big break came when one of my instructors told a friend at Hughes Aircraft Company that he had a student with some promise. Soon I was a fledgling programmer at Hughes hired by a man named Woody Stark who helped many a young person into computing with more than a few achieving significant positions in the Information Technology Industry.

My first assignment was working on the Labor System. Information would be collected from employee payroll processing and passed to the Labor System, which would slice and dice the data to meet various requests from “people managers” and “project managers.”

There were reports of all kinds sorted by numbers such as payroll, project, department, month, day, week, etc. Numbers carried the reputation as being more precise than alphanumeric characters and more easily processed by the computing hardware of the day.

We worked with coding sheets that were then delivered to keypunch operators to be turned into 80-column punched cards and subsequently fed into the early model IBM System/360 computers. And then, there were the keypunch girls…..but that is another story.

But from these carefree beginnings the seeds of pending disaster were sown in Hughes Aircraft’s Business Data Processing shop and other similar installations throughout the country. Unlike the Scientific computing community whose applications only lived for the life of their current project, the business applications would stay around much longer than any of their creators envisioned and eventually take on a life of their own.

I can recall decades later being approached by young IT professionals saying, “Gee, I worked on one of your programs the other day.” I’d just smile and cringe inside for I knew the seeds of disaster were germinating…...from the 2-digit year code that would plague the late 1990s with the likely outcome of reverting all dates to 1900 on January 1, 2000; the "*spaghetti code*" with no structure or organization that was only understood by the "system gurus;" to the never-ending strings of programs to turn out just one more series of reports. These problems, unknowingly created by early business programmers, would become the most insidious issues facing future generations of Chief Information Officers.

Challenges in the mid-1970s

So now, let's fast-forward to the mid-1970s. Systems are bigger and take longer to run. But IBM and other hardware vendors increased the computing power by a factor of 2 every year….so not to worry! But start to worry we did!

By this time I am a first level manager. I am in charge of the Cost Information System, which collects and reports on work performed by Hughes on various customer contracts. The system is relatively new and was created by a team made up of a Program Manager (Jack Stevens, a user with experience managing the activities for which the system was being created) and a data processing manager (George Johnson) with a strong formal business education. This combination (probably the first at Hughes) resulted in a well thought-out and well developed system to serve the program management community throughout the company (probably another first). While the Cost Information System would continue to grow and serve evolving needs over the years, the core was a solid set of data to be used in reporting contract activities.

Here enters the first of two problems I will expand on in this narrative:

The wall clock has 24 hours, with 63 hours available between 5pm Friday and 8am Monday. The Cost Information System (CIS) takes its data from all the systems that start running on Friday night. Since the company is growing (Hughes is the largest private employer in California with 23 plant sites in Southern California and Arizona) it follows that the systems keep growing. Sooooo, there is more work than can be processed in the 63 hour weekend window. Furthermore, if there are any interruptions, everything is even further behind. CIS is the last step in a long processing stream and for Hughes customers, probably the most important.

At that time, we had graduated from the first IBM System/360 computers that processed on a single thread (one application at a time) to later generations of equipment that could process multiple application streams simultaneously. Problem was, most of the applications were written and processed with a "single thread" processing approach where one program followed the other until all the system processing steps had been completed.

CIS, which under the best of conditions began processing on Sunday morning, now processed well into Monday graveyard shift and sometimes into the next business workday. Not only did this mean that to print and distribute these critical reports took well into the workweek, but also the conflict created by dayshift computing combined with the still running CIS caused chaos across the company. So what to do?

After many months of increasing difficulty getting the CIS process completed before the Monday morning dayshift began, I hit upon the idea of “Pitchfork Processing.” This concept took data from updated CIS master files and spawned multiple strings of processing jobs to simultaneously act on those data. Previously, programs were run one at a time until all the weekend’s processing steps were completed.

The end result of Pitchfork Processing (think about the multiple tines of a pitchfork radiating out from the single handle) was an improvement of 12 wall clock hours in the completion of the weekend processing. And of course, it thrust me into the spotlight as someone who thought out of the box (never to be just one of the guys again).

Let’s fast-forward another 5 or so years to what can be labeled the Year End Accounting Close Crisis. By that time, I am a mid-level manager responsible for all the Finance and Accounting Systems supporting the company.

Every year some poor member of the Operations or Programming staff was anointed as the Year End Leader; this was a task in which purgatory would be a welcome step up. And every year, with the company growing and the systems becoming more complex, there was a disaster.

These disasters often came in the form of problems early in the multi-week year-end close processing cycle that were not uncovered until the processing had gone on for days or weeks. They gave the data processing organization a very bad name within the user communities.

To the surprise of many, rather than try to hide from this onerous task, I volunteered for it! I figured that I could make it work the first time without a mishap…..and I did.

How did I do it? I treated the year-end as a major project to be planned and managed rather than to ride out the storm hoping for the best.

First I developed a plan and formed a team made up of Operations, Programming and Accounting personnel.

We declared a moratorium on programming changes, ensuring that every program running during the year-end process had been run in a production cycle at least once. Controls were formalized and rigorously checked following each process, there was a year-end manager in charge for every shift, status meetings and handoffs followed each shift as well as a single master log of every activity, problem and decision made.

The result was a year-end close on time and correct for the first time in memory.

How? In short we overwhelmed the problem. But we didn't stop there. Many of the brute force steps taken to achieve this success were integrated into the production job set. This resulted in not only a better year-end process but more reliable processing throughout the entire year.

Again, the spotlight shone and lit my path for the remainder of my time at Hughes.

While I held loftier positions and met greater challenges, the personal satisfaction from my hands-on success with these two computing challenges was never duplicated.

To my mother's credit, her readings provided me an enjoyable and lucrative career. However, now that I have stopped working for other people, I have returned to my true love….being a classic car restorer.