the Availability Digest

www.availabilitydigest.com
@availabilitydig

The End of Custom Software?

November 2017

Back in the 1970s and 1980s, my company The Sombers Group specialized in writing custom software packages for our customers. Today, software-implemented packages are available to meet almost any computing need. But back then, there was a paucity of such products. The need for software solutions to corporate challenges kept busy about a dozen of our staff. We served a half-dozen customers at any one time.

We programmed primarily using the basic assembly languages of the machines on which we worked. The only alternatives then were COBOL and FORTRAN – Java had not yet reared its wonderful head. Most of our work was performed on Digital PDP-8s and PDP-11s and on Tandem systems (now HPE NonStop systems). Below are just a few of our projects. It is my belief and that of others that the days of start-to-finish custom software are substantially behind us.

What Is Custom Software?

Custom software is an application designed for a particular user or group of users within an organization. The software is designed to meet their needs precisely as opposed to off-the-shelf software. Such software is typically created just for that specific entity by a third party or by an in-house group of developers. It is not packaged for reselling.

The Benefits of Custom Software

The benefit of custom software is that it provides exactly the features that a company needs in an application. These features simply may not be available in a pre-packaged software application. Implementing an application to specifically meet the needs of an organization can increase the productivity of the organization's staff. The cost of producing the custom software package is offset by the increased efficiency it brings.

If an organization has a unique need that is important enough to warrant the expense of a custom software application, then implementing that application rather than settling for an off-the-shelf application can prove to be a smart course of action.

Concerns with Custom Software

Whereas pre-packaged software applications can be purchased relatively inexpensively, customized software entails a much greater investment and carries with it significantly more risk. Off-the-shelf software can support a low price because the cost is being distributed among many users. Custom software is created for one user, and that user must bear all the costs.

The designers of the custom software must have an in-depth understanding of the user's need and how he wants the end product to address that need. Identifying new needs during the development process is not uncommon. However, these new needs add costs and result in 'scope creep.' Scope creep can result in an end product that is insufficient to meet the original needs.

Custom Software Packages That We Implemented

Following are some of the custom software packages that we wrote decades ago.

Custom Payroll Software

To meet the needs of one of our customers, we implemented a payroll package running on a PDP-8. There were no off-the-shelf payroll packages available back then. The package that we wrote was so successful that I decided to build a new business around it to provide payroll services to small companies. Thus, MiniData Services was born. We put up billboards advertising "You Pay Us 9, We'll Pay 15." This was an indication that we would do a fifteen-person payroll for just $9.

MiniData purchased two PDP-8 computers to run our payrolls. We only needed one from a capacity viewpoint, but we realized we needed two in case a computer failed. Even back then, redundancy mattered. However, in the years that we ran the dual PDP-8s, we never had a system failure. We later upgraded to PDP-11s to gain additional processing capacity.

Our first payroll customer signed up with us within one week - a small local bank. We proudly delivered our first weekly payroll to the bank, only to be greeted with a telephone call filled with expletives. It seemed like we were paying their employees several thousand dollars an hour! It turned out that our payroll package was set up to accept time-worked in weeks, whereas the bank had given us time-worked in hours. So we were paying each employee a week's salary for each hour worked. We learned our first lesson – check the payroll sheets before we deliver them.

We printed payroll summary sheets and paychecks on a high-speed printer. One problem we faced was that it took several minutes to tear down the printer and change forms. We were doing hundreds of payrolls for companies with employees numbering in the five to twenty range. If we had to change the printer for each summary sheet and for the specific checks for each company, we would never be able to process enough payrolls to build a business. We solved the problem by using a standard check for all customers.

We would print the payroll summary sheets for about a hundred customers and then put the standard blank checks in the printer. We printed each check with the company name, the employee name, and the amount. We burst the checks and placed the checks for each company with its payroll summary sheet. We then passed the checks through a magnetic ink encoder to print the companies' bank account number, the check number, and the amount in magnetic ink at the bottom of the check.

We were told that this was illegal, but it was quite legal. We never had a problem with it. The technique let us print the payrolls of a hundred or so companies in a single pass, thus letting us process the payrolls for hundreds – and eventually thousands – of companies each week.

MiniData's servicing of small payrolls became so successful that after a few years, it was copied by payroll giant ADP. MiniData Services grew to the point that it was ultimately acquired by the payroll division of Control Data Corporation.

Middleware

Another Sombers customer needed a software package that would allow it to communicate with remote computers. The need was to send messages between Digital PDP-11s. To accommodate this need, we developed a custom software package to transfer messages directly between the computers.

The package worked so well that we decided to package it up into a product to support message transfer between a variety of computers. Thus our product NetWeave was born.

NetWeave competed with IBM's MQ Series. However, whereas MQ Series placed messages to be exchanged in a queue, NetWeave immediately transferred the message to the recipient machine with no queuing, thus earning its nickname "the lean, mean messaging machine." NetWeave was expanded to support several other processors such as PDP-8s and Tandem systems.

NetWeave became quite popular and was licensed by many companies for interprocessor communication. It is still in service today, some forty years later, based on the payment of license fees. However, since NetWeave is a middleware product and operates just above the operating system, most companies that still use it are not at all aware of where it is being used.

Racetrack Totalizator

Perhaps the largest special software system we built was for the New York Racing Association (NYRA). NYRA operated three race tracks in New York state – Aqueduct, Belmont, and Saratoga. Each race track depended upon a totalizator system, known as a 'tote' system. The tote system accepted wagers from ticket issuing machines (TIMs) around the track; calculated the potential payoffs for each horse if it should win, place, or show; and posted these payoffs on a large infield display board and display terminals situated around the stands. It also calculated the payoffs for more esoteric wagers such as the daily double (picking the winning horse in two different races) and the trifecta (similar but for three races).

When The Sombers Group was called in, the tote systems for the NYRA race tracks were being provided by Automatic Totalizators, Inc. (Autotote), located in Australia. The systems were implemented via relay logic. The mechanical relays caused problems, and Autotote considered replacing them with computers.

We implemented a tote system for Autotote using a PDP-8. This included building a TIM-scanner complex that would scan the TIMs in the race track to pick up wagers. The wagers were logged in the computer tote system, and signals were returned to the requesting TIMs to issue the tickets. The first PDP-8 system was put into service at the Saratoga race track that operated for just a month in the summer in Saratoga, New York. The tote system performed superbly, and Autotote decided to use this technology at its larger racetracks at Aqueduct and Belmont.

To provide the capacity for these larger race tracks, we ported the code to PDP-11s. Three PDP-11s were provided. One was a backup system. The other two systems actively processed TIM requests and compared results. If both computers agreed to sell a ticket, a sell signal was sent to the TIM. If both computers rejected the sale, the TIM was told to cancel the ticket.

If one computer approved the sale but the other rejected the sale, the TIM was told to cancel the sale. If there were a series of cancellations by one computer but not the other computer, the computer that was doing the canceling was taken out of service; and the third backup system replaced it.

We later reprogrammed the tote system to run on Tandem systems. This was in the early days of Tandem, and we were one of the first users of these highly reliable systems. Reliability was of paramount importance. Imagine a customer who wanted to place a $50 wager on a long shot but couldn't because the tote system had just failed. Then his horse wins and pays 20:1. He would have won $1,000. Incidents such as this could cause riots at the race track.

Magazine Publishers

Time Magazine had a need to communicate effectively with their correspondents in the field. A correspondent is a freelance journalist who contributes reports to a news organization on an ongoing basis but is paid individually for each piece of published work.

The correspondents interacted with the central editorial group in New York City via an aging communications system that was beginning to exhibit frequent failures. These failures could be disastrous as issue time approached.

Therefore, it was imperative for the magazine to develop a new system, one which would be fault tolerant, would provide 100% availability, and would interconnect the correspondent/editorial network.

Sombers developed for the magazine a news-desk message switch and editorial system. The system used two Stratus fault-tolerant computers supporting a network of text-editing PCs. One system handled all communications, and the other system was a database manager that provided rapid access to files in a relational data structure.

In addition, Sombers developed a multitasking, PC-based text-editing package that integrated functions of the Stratus computers with those of the news desk staff's IBM PCs. The result was an exceptionally fast response time for search, edit and storage requests.

The magazine's staff of editors, correspondents, and writers who work out of thirty-three bureaus worldwide were able to communicate with each other and have access to the system's full editing capabilities and editorial functions. Stories were made available to the news desk staff for review prior to distribution to section editors and writers. Edited stories were then returned to the originating correspondents for additional input.

The system routed queries, responses, and administrative messages between the magazine's New York staff and its correspondents. Its primary functions were to serve as a communications hub, a store-and-forward system, a text editing system, and a database management system.

The network was so effective that McGraw-Hill had us implement a similar network for them.

The End of Custom Software?

Sadly, the interest in custom programming is fading. Existing software packages abound for just about every need. They are often easily customizable to meet the needs of a specific organization. Sometimes this is as simple as making changes to tables.

True, thousands of programmers are still kept busy. However, they are by-and-large working on the software packages upon which everyone now depends rather than custom, one-off software projects.

Having said that, we must point out that there is still a thriving need for custom software. A Google search will show that there are many companies still dedicated to creating custom software packages to meet customer's specific needs. As long as the needs of customers exceed the capabilities of packaged software, there will always be a need for custom software.

Summary

The custom software packages that we wrote were successful because they gave companies what they needed. Our custom software for inter-computer communications was so effective that we bundled it into our very successful product – NetWeave.

The MiniData payroll package was customized for each new customer to provide specialized payroll deductions. The racetrack totalizator package was customized to reflect the configuration of ticket-issuing machines, display terminals, and wagering types at each racetrack. The magazine network software was easily reconfigured to support another magazine.

However, the need for custom software still exists. There are still many companies that specialize in custom software, though not as many as in the past because of the success of existing packages.