

---

## **Core Value 3—Software Super Bowl Predictions... Don't Bet Against Winners Just Because They're Winners.**

---

Many fans treat the process of Super Bowl predictions as a science. Metamor reviews the team one position at a time, balancing our team's strong points against our opponent's weaknesses. We pour over last year's statistics, yards gained, yards rushed, quarterback ratings, and team by team match ups. In the end, these numbers rarely tell the whole story. In 1985 (the year Chicago won the Super Bowl), the Bears offense was rated 20<sup>th</sup> in the NFL. Yet, the only number that counted was the score at the end of the Super Bowl. After the game many of us, as Monday morning quarterbacks, analyzed the Bears overall 1985 performance. Many of our opponents in the NFL looked stronger on paper but failed to win the Super Bowl. The difference for the Bears in 1985 was their desire to win, commitment, and execution.

Let us apply this same measurement to our industry. IBM, Digital, HP, Oracle, Sybase, Informix, Microsoft and others, all have strong software developers. They look equally good on paper. All of these companies are in the Client-Server play-offs vying for our system dollars. Within this framework, the expansion teams, the "PC companies", are viewed as inappropriate choices for our mission critical server applications. When selecting a database for a large-scale system, many industry pundits are betting against Microsoft, claiming that NT and SQL Server are not yet ready for prime time. I will take that bet, and let me tell you in the following paragraphs why.

Let us begin by reviewing a bit of Microsoft history. In 1980 when IBM introduced the PC, Digital Research's CP/M operating system dominated the desktop marketplace. Digital Research produced a growing line of software products that positioned them for success. Microsoft, who at the time was best known for its Basic language interpreters, put together a deal to purchase MS-DOS, repackage it, and sell it with Basic to IBM. At the initial introduction of the PC, CPM was a better operating system. Over the years, as MS-DOS has come to dominate the desktop operating system market, we forget that Microsoft had to beat out a worthy competitor to win its current market share.

In 1985 Microsoft introduced MS-Windows. It needed to compete with Digital Research GEM (used by early versions of the Ventura Publishing tools), IBM's TopView, Quarterdeck's DesqView and VisiCorp's VisiOn. Version 1 of MS-Windows was not usable, version 2 almost ran, and version 3 was usable on one of the new 286 (AT) systems but it had limited application support. With version 3.1, Microsoft captured over 90% of the desktops. This same story continues today.

Word Perfect overpowered Microsoft's early entry into the word processing marketplace, with Microsoft's market share measured in single digits. Microsoft did not give up, releasing version after version of the product in a features war with Word Perfect. With the delay in the introduction of the Word Perfect for Windows, Microsoft had an opening, and used it to become a leading vendor of word processing software. The same pattern repeated in the spreadsheet and end-user database market segments.

What is Microsoft's secret? Persistence. Microsoft may be the most persistent company in business today. In sporting terms they have a greater and more consistent desire to win and are willing to work at it longer and harder than their competitors.

Let us see if this historical view applies to the bet your business client-server marketplace. In the 70s, mission critical systems were built on mini-computer and mainframe systems from IBM, Digital, HP, Prime, Data General and others running proprietary operating systems such as VAX/VMS and MVS. Specialized run-time environments such as IMS, CICS and ACMS were used to support high volume transaction processing. During the early 80s as the relational database industry matured, vendors such as Oracle, Ingres, and Sybase captured significant market share on the mini-computers (mostly VAX systems). Soon, with the year of UNIX beginning in the mid-eighties, and continuing for every year since then, these database applications began to migrate to UNIX based servers.

As more robust development tools such as Visual Basic, PowerBuilder, Uniface, and SQLWindows made GUI clients more practical for commercial applications, UNIX systems running relational databases became the standard servers. In the meantime, Sybase and Microsoft had a SQL Server product running on OS/2, but due to hardware and architectural limitations, it did not scale well beyond departmental solutions. With the release of Microsoft NT 3.5 and Microsoft SQL Server 4.21 this equation changed. It is true that NT version 3.1 was not appropriate for large-scale systems. However, Microsoft kept working on the problem and produced a 3.5 release that was solid, followed by the 3.51 release that had additional support for large systems. It is true that SQL Server on OS/2 was prone to performance problems so Microsoft released SQL Server on NT with a rewritten kernel. The rewritten kernel takes advantage of the NT multiple thread architecture to scale extremely well on SMP machines. In fact, the internal architecture of the SQL Server kernel on NT may be the best in the industry. With the features in SQL Server 6.0, Microsoft has an industry-leading product.

At this point you may be asking yourself, "What did Microsoft pay Shapiro to say these things?" Not a dime! Over the past six months I have participated in a significant benchmark of the NT3.5 and SQL Server 4.21 environments.

On behalf of a client for which we are building a 500+ user OLTP application we investigated the possibility of using an NT environment to replace their existing Unisys Mainframe. The question posed was, "Can an NT/SQL Server system handle the current order processing load and provide room for expansion?" The motivation for investigating this option was the significant hardware and software savings of an NT configuration when compared with a UNIX system running a traditional minicomputer database.

This proof of concept was not intended to benchmark the hardware platforms alone. It was designed to prove that the combination of the software and hardware could meet the high transaction rates required. To prove this theory, the Metamor team created a sample application using Uniface and built a test lab with 65 PCs and three different servers using a Digital Alpha 2100 with dual 195 MHz processors, a Data General dual 60 MHz Pentium, and a Dell single 90 MHz Pentium. One

of the goals was to compare the performance of the three server configurations.

Next, 1.2 million customer records and 20,000 inventory items were extracted from the mainframe and loaded into the SQL Server database. The test database consisted of over 25 tables loaded with real-world data volumes. We needed to demonstrate that the configuration of hardware and software could support the processing of 12,000 orders per hour on a fully populated database. SQA Robot was selected to drive the test because it supported the application written in Uniface, and it provided a comprehensive scripting capability. To ensure that the transaction profiles were valid, the team analyzed data from the existing system, identifying peak volumes and the distribution of one line item orders, five line-item orders, etc.

While the results varied based on server configuration, all of the configurations out-performed the existing customer mainframe environment, providing response times of under 2 seconds while processing more orders per hour than the legacy systems.

Microsoft's persistence has paid off again. I do not believe all new applications should be deployed on NT/SQL Server. At this time, SQL Server's cross operating system support is poor (SQL Server only runs on NT) and its database replication support is limited to SQL Server databases. I do believe that NT/SQL Server is the environment to beat for mid-sized, up to 1,000 users, single platform client-server applications. In conclusion, do not bet against Microsoft.