FROM THE QSM DATABASE – PRODUCTIVITY STATISTICS BUCK 15 YEAR TREND

By Doug Putnam

The QSM database is one of the most comprehensive repositories of modern day software projects collected worldwide. It contains trends from over 5,400 completed software projects from the U.S./North America, Europe, and the Far East, representing over 200 million lines of code, 100+ development languages, and 55,000 person years of effort. During the course of the last 20+ years, QSM has maintained this database and licensed it to companies to serve as their own repository for their software and IT metrics.

The largest segment of data currently in the repository represents IT projects. Over the years, QSM has continuously monitored application-development productivity with respect to cost reduction, speed, and quality improvement. Generally, since the early 1980's, all of these dimensions have steadily improved.

However a recent productivity study revealed that this trend has undergone a reversal during the most recent three-year time period. IT applications completed between 1982 and 2000 were extracted from the database. The data sample was then sorted into six three-year time periods spanning 18 years. In each time period, trends were plotted for project size, average Productivity Index (an indexed measure of overall project efficiency), schedule, effort, staff, mean time to defect (MTTD) and reuse.

In the context of significant industry dynamics - Y2K, enterprise resource planning (ERP) solutions, the dot-com explosion/implosion, outsourcing, object oriented (OO)/client server development, etc. - the charts reveal significant findings in the context of long-term productivity trends. These are described below.

Developed Software Size Behavior Over Time

Figure 1 shows the average project size based on new-plus-modified functionality for IT projects, beginning with a three-year time period starting in 1982 and continuing through 2000.

The overall trend through the 1980's and early 1990's was a steady reduction in project size. On average, the size of software projects was cut in half during the 15-year period from 1982 to 1997. This was generally the result of more powerful development languages as technology progressed, along with deliberate strategies by IT organizations to manage projects to 12- to 18-month schedules. Generally speaking, there was also implementation of reuse architectures such as object libraries and classes as "buy and modify" IT strategies, versus building applications from scratch.

In the 1997-2000 time frame, a radical change occurred, in which the average size of IT projects virtually doubled. This was the case in size measured by both function points and lines of code (LOC), and it reversed the 15-year trend in a dramatic way.

One potential cause of this reversal is the explosion of Internet, e-commerce, and Web development architectures during the 1997-2000 time frame. This coincides with many first-generation Web products (both sites and tools) where there was no previous existence of reusable code. Many of these applications had to be built from scratch.

Although no one can say for certain, we speculate that software size may reduce in the near future in a gradual fashion, as the architectures built during the last three years are leveraged in future generations of IT projects.

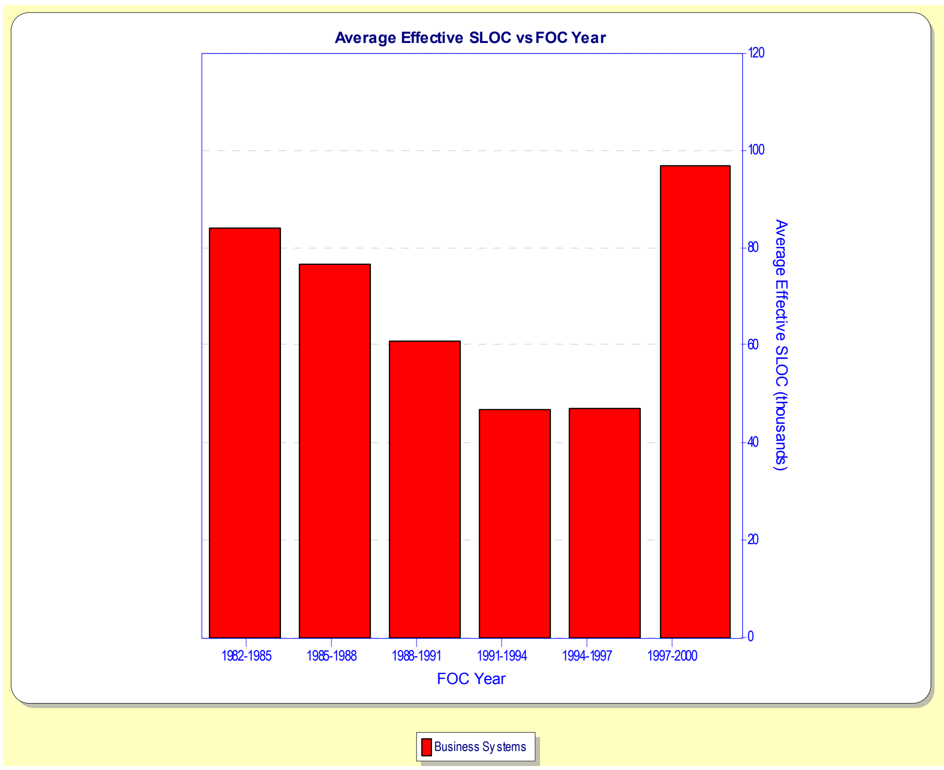

Figure 1. Average Effective Size vs. three-year time periods over an 18-year period.

Average Productivity Index Performance Over Time

The QSM Productivity Index is an aggregate measure of process productivity, calculated from metrics for size, time, and effort of completed software projects. These metrics represent three of the four core measures expressed by the Carnegie Mellon Software Engineering Institute Minimum Data Set - an established industry standard.

The Productivity Index (PI) is different from traditional measures of applications productivity that emphasize only two dimensions of metrics, such as output size (i.e. function points or lines of code), per unit effort (person-months). It incorporates shortening or lengthening of the development schedule by including development time in its calculation. Therefore, each index rise corresponds to a reduction in effort (about 25 %) and/or a shortening in time (about 10 %) from the previous value.[1]

The calculated PI increased over the 15-year period from an initial base value of 13.8 to 17.3 by the year 1997. However, during the 1997-2000 time frame, the three-year average dropped to 16.6.

We believe that several factors may have been at play to drive productivity downward. These include:

- The adverse impact of resources diverted to Y2000 projects.
- Labor churn from rotating staff to e-commerce and web initiatives.
- Significant learning curves associated with customizing and implementing large-scale applications such as ERP.
- Dramatic shift in project complexity from traditional IT applications to those that incorporate more complex elements such as wireless telecommunications, system software, fiber optic, and even real-time elements.

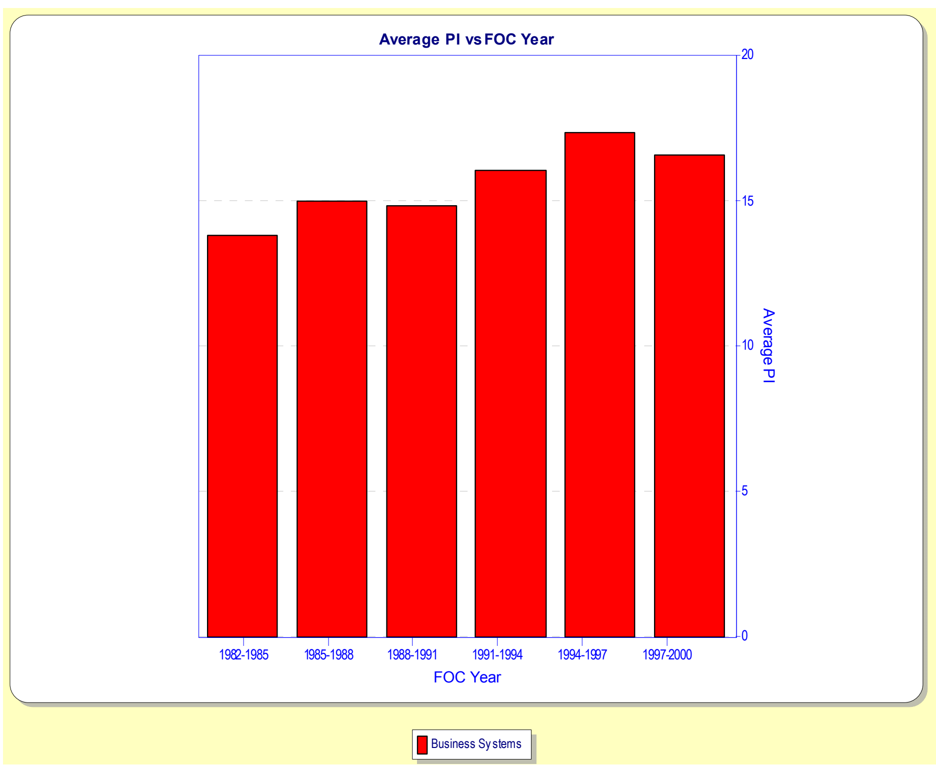

Figure 2. Average Productivity Index vs. 3-year time periods over 18 years

[1] Traditional metrics for productivity do not include time. Therefore, if effort improves but schedules lengthen, their "improvement" can potentially be misleading.

Staffing Performance Over Time

Figure 3 shows an average staffing profile on a typical project over time. The trend had been reasonably constant in the range of 6 - 7 people per project during the 1990's. In 1997 to 2000, the average project team increased to 9 people. That's about a 50 % increase in average team size.

There appear to be two factors contributing to this trend: project size growth and an acceleration of project deadlines to complete at Internet Speed. In essence, companies are striving to build even more functionality in less time, and react to these pressures by adding more people to projects.

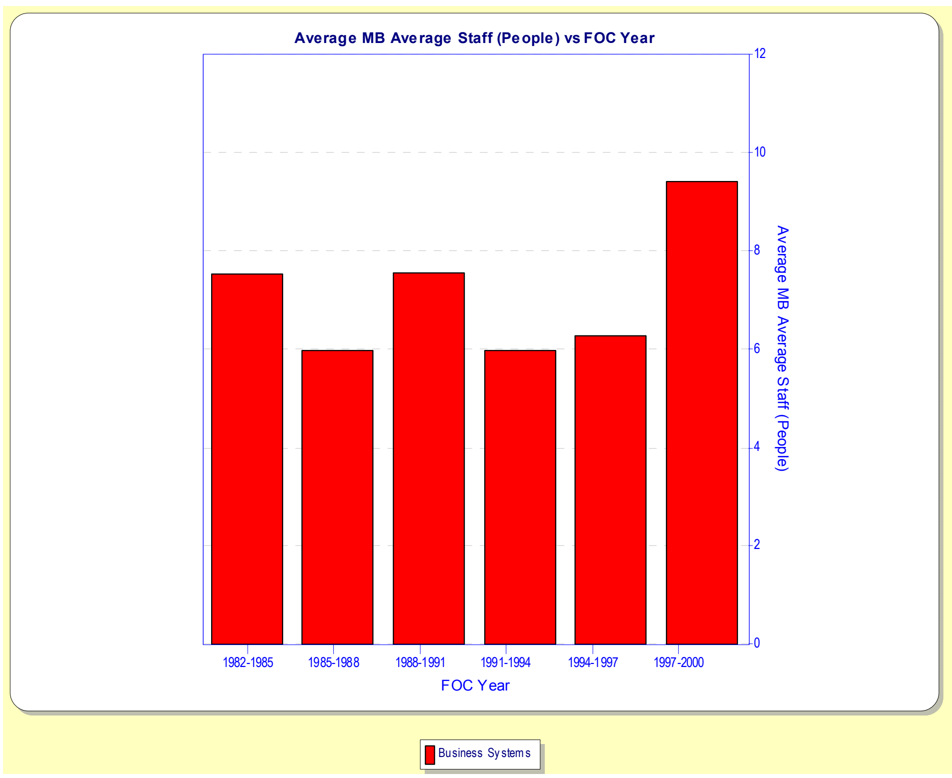

Figure 3. Average Staffing vs. 3-year time periods over the 18 years.

Schedule Performance Over Time

Figure 4 shows the average duration of projects for each three-year time period. In 1982, the typical IT project lasted nearly 2.5 years. By the 1994-1997 time frame, average duration had dropped to 8 months! That's a pretty impressive trend over the 15-year time period. The two most important driving factors were the reduction in project size and the improvements in productivity (see figures 1 and 2). Both behaviors result in schedule reduction.

It's interesting to note the industry studies on overruns and slippages, beginning with a study by the U.S. General Accounting Office in 1979, up to and including the recent Standish Group Chaos report in the late 1990s. The data is irrefutable – projects have completed faster and faster every year. One can only surmise that the industry "overrun and chaos" studies reflect that demands and expectations simply outstrip even this dramatic rate of improvement. Internet-speed deadlines, overly optimistic estimates, and scope growth/change may be more believable culprits with respect to projects being "late".

However, in the 1997-2000 time frame, this trend sustained a reversal, in which the average project schedule grew to 9.5 months. This is likely due to the combination of project growth and the aforementioned drop in the QSM Productivity Index.

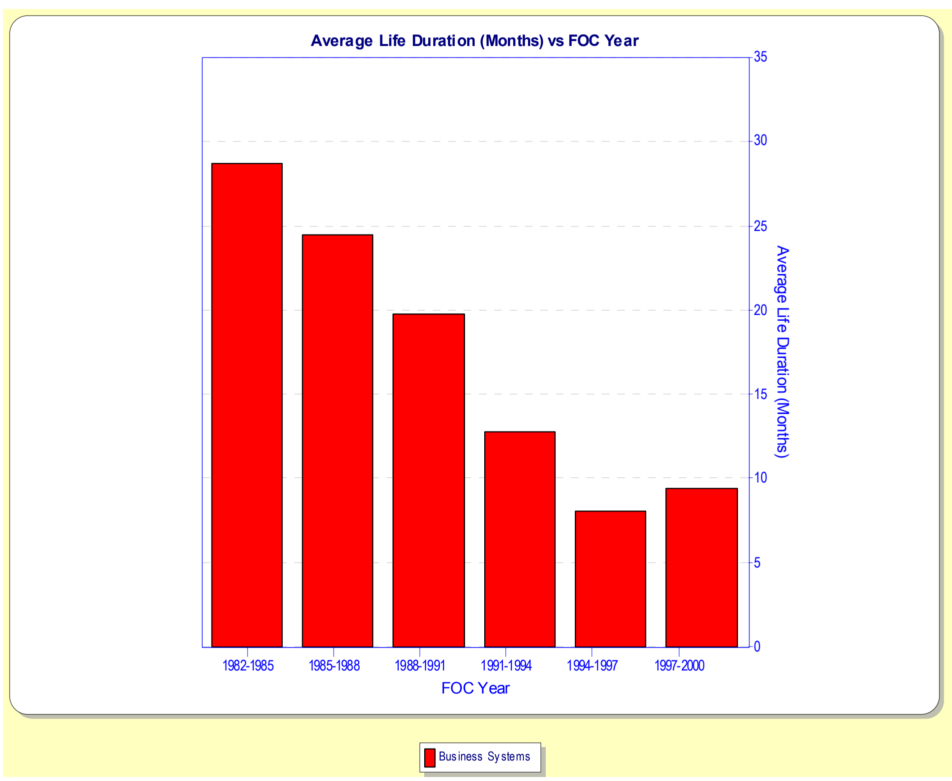

Figure 4. Average schedule vs. 3-year time periods over the 18 years.

Effort Performance Over Time

Figure 5 shows the average effort for each time interval. In the 1982 – 1985 time frame the average effort per project was over 165 person months; 15 years later, that figure dropped dramatically to less than 60 person-months.

However, in the 1997-2000 timeframe, average project effort (and associated cost) nearly doubled to over 100 person months, at a cost of about US $1.5 million. This is a dramatic reversal of the previous 15-year trend. It appears to be a combination of project growth/drop in reuse, the 50 % increase in average team size, and the modest drop in productivity.

One might expect that rising project costs might be a factor in companies attempting to reduce costs by outsourcing in the economic climate over the last several years.[2]

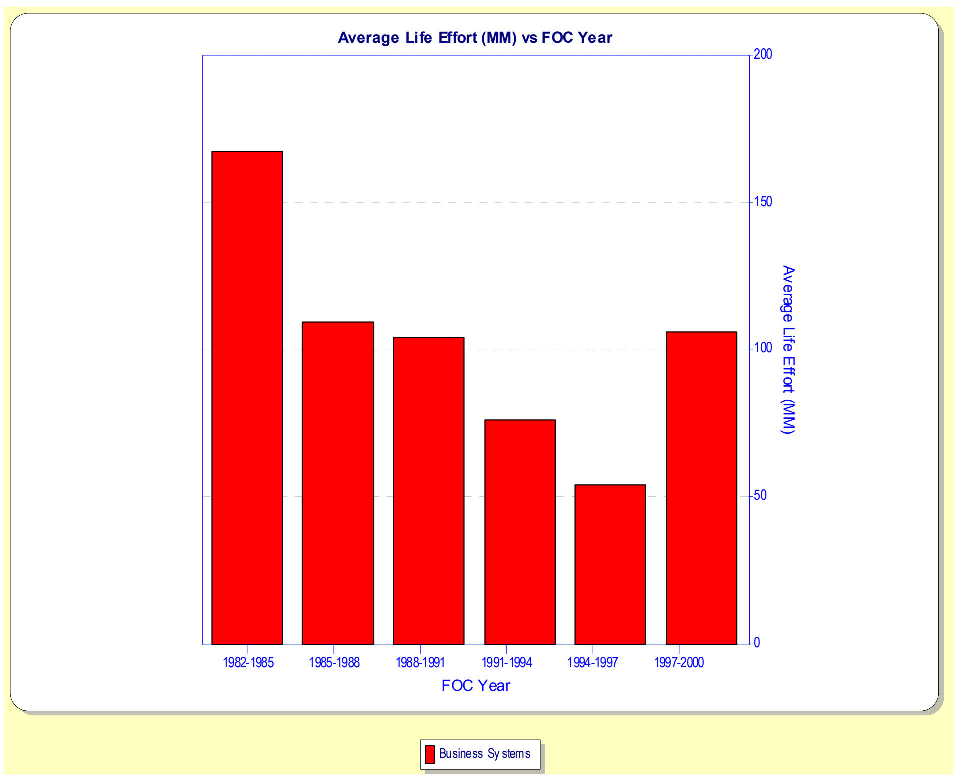

Figure 5. Average Effort vs. 3-year time period for the 18 years.

[2] It should be noted that the QSM Database is not restricted to projects built in-house. It contains statistics for both in-house and outsourced projects.

Software Reuse Performance Over Time

Figure 6 shows the trend in software reuse. The reuse is expressed as the percentage of reuse that was achieved during the three year time period. With the exception of the initial three-year data segment, the overall trend from 1985 to 1997 was an increase of software reuse. This approached 65% during the early and mid 90's.

From 1997 – 2000, the reuse trend retreated back to approximately 50%. We believe that most of the Internet and first generation object-oriented projects were comprised of new development.

Looking forward, it's likely that 60-70% reuse is a practical upper limit that can be realistically expected over a broad range of products in an organization in the normal course of business.

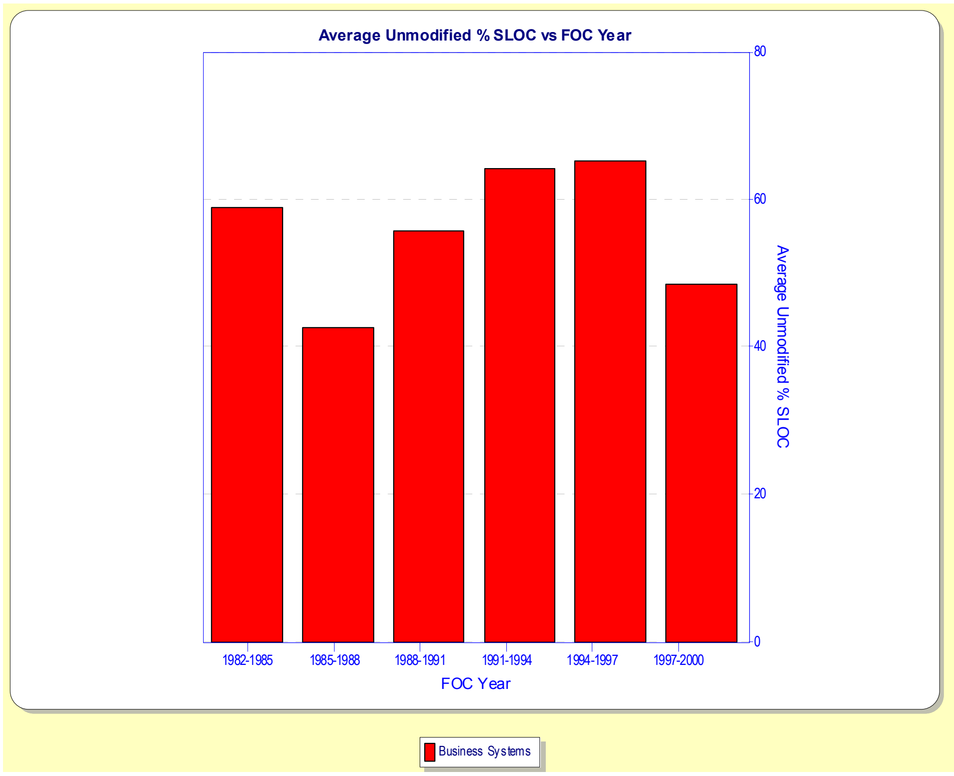

Figure 6. Average reuse vs. 3-year time periods for the 18 years.

Mean Time to Defect Over Time

Figure 7 shows the trend for reliability at delivery over time. The reliability is expressed as Mean Time to Defect (MTTD), or the average time between occurrences of runtime errors in a software application.

The data shows that MTTD remained relatively constant during the 1980's at about five days on average, and improved to just under nine days during the 1994-1997 period. Then, during the three-year segment from 1997 to 2000, it improved dramatically to an average of 12.5 days.

Contributing factors are likely to include improved process maturity and more attention to quality issues. Many modern-day software applications also require 24 hours-per-day, 7 days-per-week operation, with greater emphasis on system availability. All of these advances appear to be producing good results with respect to quality.

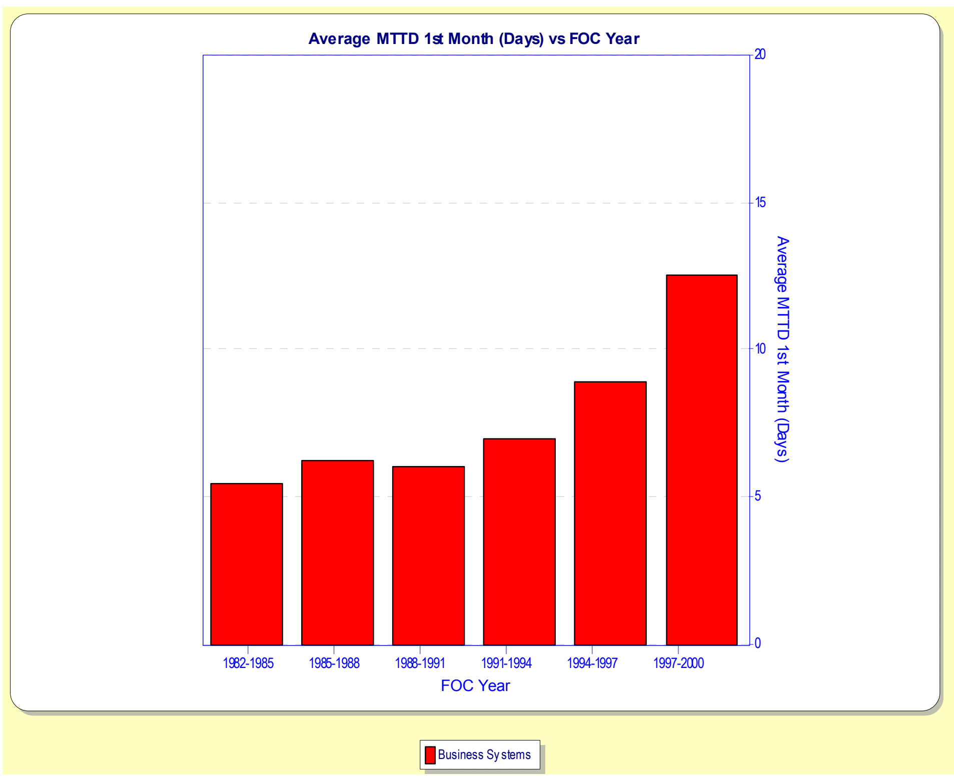

Figure 7. MTTD vs. 3-year time period over the 18 years.

Conclusions

In summary, the data reveals significant changes in applications productivity in the 1997 – 2000 time frame. With the exception of quality improvements, all other indicators reversed a long-term 15-year trend.

- Staffing was higher.
- Effort was higher
- Schedules took longer.
- Software reuse was lower.
- Project size (new + modified functionality) was much larger.
- Productivity was down.

We speculate that several underlying factors were at play during this time frame, which may have been the root causes of these results. It appears that they comprise a significant increase in IT project complexity during a time of dramatic change. In this context, temporary decreases in productivity are to be expected. These factors include:

- Implementation of off-the-shelf ERP solutions was attempted on a large scale in many organizations around the Y2000 time frame. In many cases, the complexity of these endeavors was widely underestimated during the planning stages.
- Object Oriented (OO) development was started in earnest. This meant a new infrastructure had to be built for all application-specific classes. Many organizations discovered that implementing OO was more difficult than anticipated. It took more time and more effort during its initial adoption.
- Web based development and the advent of dot-com enterprises pulled many talented engineers away from traditional development, producing turbulence as highly skilled people left for the Internet startups and lowered the overall skill level in the *Fortune* 1000 companies.

The data shows that productivity and the other associated management metrics don't always improve linearly in year over year. We believe the data reflects the turbulence that was experienced during the last three years. However, over the long term there is no question about the productivity improvements that our industry is exhibiting. The recent data may simply be reflecting a slowdown that manifested itself during the turbulence of the Y2000 transition and the Internet and e-commerce revolutions.

This analysis demonstrates the industry insights that are achievable through the use of metrics. Some companies already conduct analyses like this within their own companies, but many others don't know where they are, where they are headed, and have no road map to guide their decisions moving forward.

To avoid this fate, establish your own productivity benchmarks and set your process improvement plan in motion. If you do, it will be possible to see and explain what's going on in your take pro-active steps to improve your productivity and set yourselves ahead of the pack.

About the Author

Doug Putnam is Vice President of Professional Services at QSM, Inc. He has more than 19 years of experience in the software industry. Mr. Putnam has written and lectured extensively throughout the world and has participated in over 100 estimation and measurement engagements during his career with QSM. He can be reached at doug_putnam@qsm.com.