

IC Development Capability— Achieving Best-In-Class

White Paper

Developed by
Numetrics Management Systems, Inc.
20863 Stevens Creek Blvd., Suite 510 Cupertino, CA 95014
Tel: (408) 351 5800 Fax: (408) 351 5850
Email: info@numetrics.com
Web site: www.numetrics.com

INTRODUCTION

Many semiconductor companies talk about being best-in-class, but few have reliable measures for determining whether they've truly earned that appellation. Benchmarking, using the methodology prescribed in [Benchmarking IC Development Capability—Why?](#), provides the means for creating a sound framework with which organizations can determine whether their IC development performance is in fact best-in-class. Not only does the methodology provide a best-in-class litmus test, but companies can quickly leverage it to rapidly improve their financial performance—by dramatically improving schedule predictability and generating realistic estimates of project staffing needs.

Good schedule predictability and best-in-class status go hand in hand. The latter is not likely to occur in the absence of the former, and both are also closely tied to risk management. In fact, semiconductor companies have begun placing greater emphasis on risk management. A vital part of this includes a rigorous, probability-based assessment of whether or not product development schedule targets are truly achievable—today, most development schedules are not achievable, but the projects get funded anyway. Many end up cancelled and most are completed very late. Indeed, inaccurate schedule estimates are a primary cause behind missed financial projections.

DEFINING BEST-IN-CLASS

Being best-in-class is a goal that IC development organizations throughout the industry aspire to. But what does it mean to be best-in-class? Surprisingly, having the shortest cycle time, the lowest spin count, or even the highest productivity doesn't necessarily equate with being best-in-class. At first glance, each of these metrics seems valid. None, however, bears up under scrutiny as a viable, stand-alone definition, even though each is meaningful in its own right.

In fact, no single metric can determine best-in-class, because being best in one category doesn't tell the whole story. Having collected and analyzed data from over 100 semiconductor and electronics companies, Numetrics Management Systems has observed that development productivity and development throughput, when viewed together, are extremely good indicators of best-in-class.

Projects that demonstrate both above-average productivity and above-average throughput typically have shorter cycle times, fewer spins and better schedule performance relative to similar-size teams developing comparable designs. In fact, projects satisfying these criteria consistently beat the competition on virtually all other key performance indicators.

Above-average productivity means that the output per individual team member is high. Above-average throughput means that the team has a high rate of output. In other words, best-in-class characterizes an engineering and management competency level in which the engineers are superior performers not only as individual contributors, but also as members of large teams.

A scatter plot of throughput versus productivity (Figure 1) shows that most projects cluster in the center of the graph. Dividing the graph horizontally into above-average and below-

average throughput, and then again vertically into above-average and below-average productivity, provides a four-quadrant schema that makes it possible to definitively identify best-in-class projects:

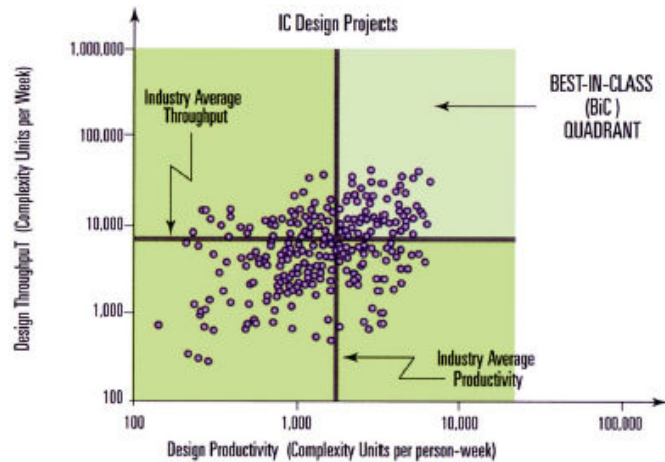
1. Low throughput and low productivity (the lower left)
2. High productivity but low throughput (the lower right)
3. High throughput but low productivity (the upper left)
4. High throughput and high productivity (the upper right), the region where the best-in-class reside

Quadrant 1: Worst in Class

Obviously, nobody wants to be in the lower-left quadrant (below-average productivity and below-average throughput). It means that the average productivity per team member and the rate of output of the overall team were both low. Lower-than-average team output, or throughput, occurs when the team is not large enough to offset the low productivity. It also occurs when the average productivity per team member is so low that even additional staffing—assuming it was applied—was unable to offset the low productivity.

Best-in-class demands high productivity and high throughput, simultaneously. Projects satisfying these criteria consistently beat the competition on virtually all other key performance indicators, including cycle time, spin count, development cost and schedule performance. (Note: each data point corresponds to a completed IC project).

Figure 1. Identifying Best-in-Class



Quadrant 2: High Productivity, Low Throughput

It's often easy to be in the lower-right quadrant. A small team of skilled engineers can be highly productive. Little time is wasted on tasks that take away from the actual development effort. But a small team can produce only so much output in a given period of time, even if it's highly productive. If a small team was put on a large, complex project, it would almost certainly maintain its high productivity, but its low throughput would result in a long cycle time.

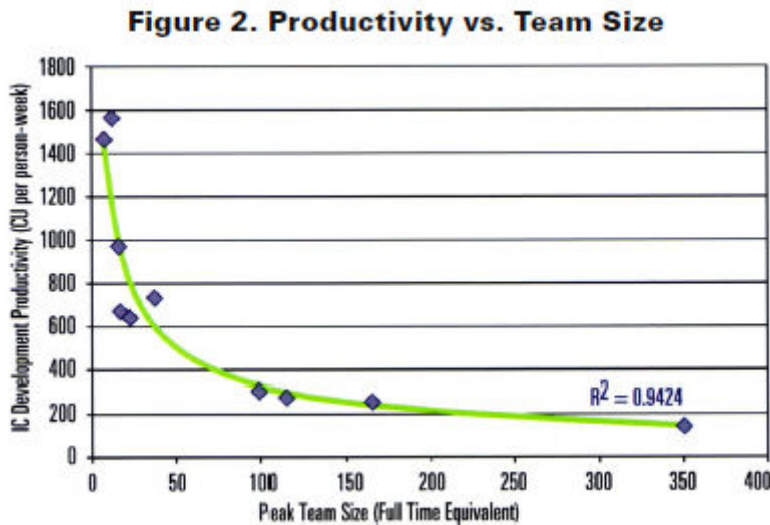
Quadrant 3: High Throughput, Low Productivity

If a team's productivity is low, cycle time targets can still be achieved by increasing the team size. High throughput is often achieved by assigning more resources to a project (if the organization has that luxury).

Of course, a large staffing level translates directly into higher overall development cost and cost per unit of output. Moreover, such a strategy means that the resources are unavailable for other projects.

Quadrant 4: Best-in-Class

The upper-right quadrant represents the best-in-class. Getting there poses a formidable challenge. As noted earlier, high productivity is generally achievable by a small development team, and high throughput is generally achievable by a large team, but achieving both simultaneously is very difficult because productivity declines as team size grows (Figure 2).



Relationship between team size and development productivity for a particular IC development organization. As team size increases, productivity declines. This is because more effort is expended on communication and other project management activities. Best-in-class organizations are capable of maintaining above-average productivity even as team size grows.

Achieving high productivity and high throughput most likely means an organization is capable of increasing team size without significantly compromising productivity, and that's generally a reflection not only of great engineering capability, but also of superior management practices and processes. That's why only the best-in-class can achieve high productivity and high throughput simultaneously.

Furthermore, since the development cost per unit of output for a given chip declines as productivity rises, and similarly the cycle time for a given chip shrinks as throughput increases, best-in-class projects score well on both cycle time and development cost and on virtually all other key performance indicators (KPIs). Best-in-class projects exhibit 2.5x more output, 23% shorter cycle time, 68% lower development cost per unit of output, 51% better schedule performance, 22% lower spin count, and 111% higher first-silicon success than the other projects. Though these statistics apply only to application-specific standard products (ASSPs), application-specific ICs (ASICs) are similar. The unmistakable performance differences between best-in-class projects and all others validate this approach to determining what constitutes best-in-class.

Best-in-class projects and companies are like championship baseball teams. Championship teams almost always score at or near the top in the two key team metrics, batting average

and pitching capability (earned run average), but they also score at or near the top in other key areas, like home runs, runs batted in (RBIs) and fewest errors.

THE REAL BENEFIT OF MEASURING PRODUCTIVITY

Striving to improve financial performance, semiconductor companies have begun placing greater emphasis on risk management, including rigorously assessing the likelihood of hitting product development schedule targets, since inaccurate schedule estimates are a primary cause of lower-than expected financial returns. As noted in [Benchmarking IC Development Capability-Why?](#), the semiconductor industry today misses its product development schedules 85% of the time, with the average overrun exceeding 50% of the originally planned cycle time. What's more, these statistics don't include those projects that are cancelled before completion.

The financial impact of project cancellations stemming from over-optimistic schedule estimates is substantial. Approximately \$2 billion is wasted annually on development of IC projects that are canceled. And of course billions of dollars in market share are lost annually by companies whose products are late to market. Much of this waste and loss is avoidable.

Measuring the productivity of a company's IC development process paves the way for generating reliable, fact-based estimates of cycle time and staffing requirements for future projects. Moreover, realistic estimates coupled with accurate measurement of productivity are the cornerstones of managing schedule risk and portfolio planning.

A FACT-BASED PLANNING METHODOLOGY

A fact-based project planning methodology together with a bottom-up approach for accurately estimating cycle time and resource requirements is suggested. The fact-based methodology uses quantified estimates of the chip's complexity, the team's productivity and a model of the rate at which effort will be expended on the project. Typically, estimates can be generated with just a few hours of effort if the proper infrastructure is in place. With the bottom-up approach, the project manager uses a traditional manual project planning tool or spreadsheet to create a detailed plan.

As an example of the fact-based project planning methodology, consider a development manager who has been allocated a project team that will have a peak staffing of 150 full-time equivalent members. The team is to tackle a wireless baseband IC design whose complexity has been calculated to be 3.5 million complexity units (CU). The immediate question is "how long will it take to complete the project?" Figure 2 shows the relationship between peak team size and productivity for the development organization. The curve is based on benchmarking some of the organization's prior projects. It shows that for a peak team size of 150, the baseline productivity approaches 300 CU/person-week. This can be further adjusted to reflect any major differences between the new project and the previously benchmarked projects—typical adjustment factors might include the team's experience level and the stability of the tools and libraries. Benchmarked projects can be used to determine

the impact these factors have on productivity. With this information, an initial estimate of the total effort required can be calculated as follows:

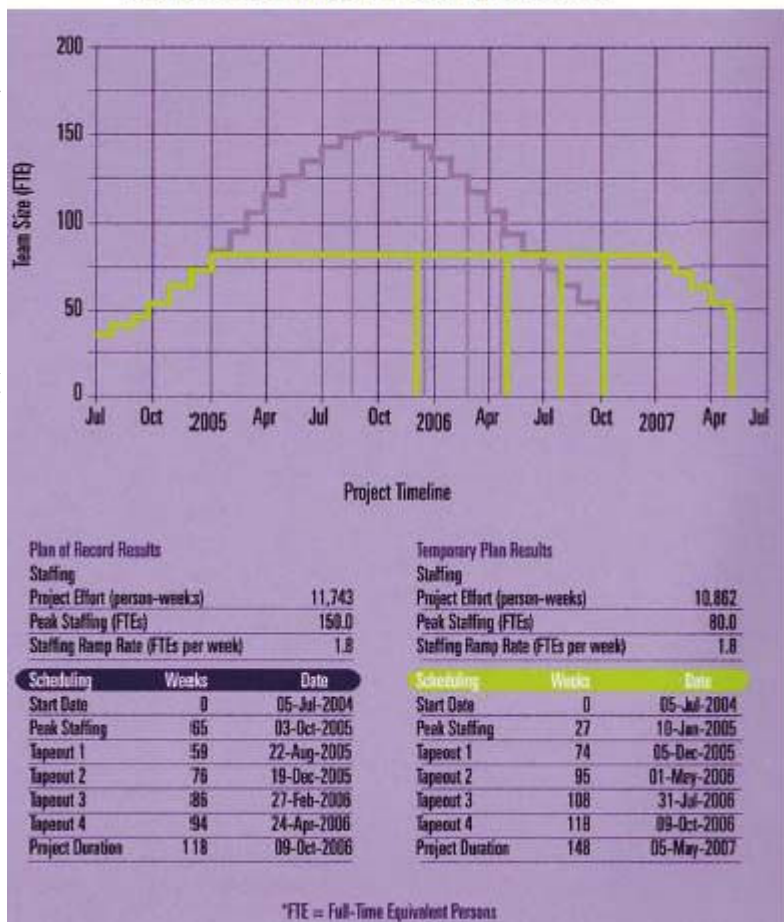
$$\begin{aligned} \text{Development Effort} &= (\text{IC Complexity}) \div (\text{Development Productivity}) \\ &= (3.5 \text{ million CU}) \div (298 \text{ CU/Person-Week}) \\ &= 11,743 \text{ Person-Weeks} \end{aligned}$$

The next step is to determine the rate at which the effort will be expended. This is expressed as a staffing contour, which is a representation of the staffing strategy—that is, the rate at which resources will be applied to the project. The staffing contour is based on the input from the project manager plus contours of similar projects. It is dictated by such factors as the IC development methodology, the particular application (e.g. wireless IC), the scope of the project (i.e. derivative or platform) and resource availability. The calculated project effort and the contour are inputs to the mathematical model that generate the estimates of cycle time and key milestone dates. Important milestones include the first tapeout milestone, as well as the tapeouts for possible re-spins.

Figure 3 shows two scenarios of the synthesized estimates of cycle time, tapeout milestones and project effort—one in which the peak staffing reaches 150 (“plan of record”) and the other in which peak staffing is constrained to 80 (“temporary plan results”).

In addition to generating estimates of cycle time and staffing, the methodology enables “what-if” tradeoffs among the critical constraints on the project: cycle time vs. staffing-level vs. chip complexity. In other words, it answers the question of “how many engineers and specialists are needed to develop this chip in the target cycle time?”

Figure 3. Project Plan Synthesis



Two alternative project plan scenarios for a complex baseband IC project. One plan allocates a team of 150 persons at the peak, whereas the other constrains the team to 80. Naturally the larger team achieves a much shorter cycle time— by 30 weeks, but the sacrifice is lower productivity and higher development cost.

Alternatively, it answers the question of “how long will it take to develop this chip with a team size of x and a productivity of p ?”

In many cases, the estimates reveal that the chip cannot possibly be completed within the desired cycle time unless either resources are added or the chip’s complexity is reduced. Since it is often difficult or impossible to relax the schedule constraint or allocate more resources, reducing complexity is the only alternative. Complexity can be incrementally reduced by the specific amount necessary to achieve target cycle times with the available resources. This can be accomplished by reusing functional blocks, as well as removing non-critical chip features—for instance, those features that easily could be included in a subsequent derivative version of the product.

This fact-based approach complements traditional bottom-up project planning methodologies. Bottom-up methodologies provide much greater detail than a fact-based approach. But they’re very time-consuming to create, and used alone, are typically inaccurate. With the combined fact-based/ bottom-up strategy, the manager works to achieve convergence, which yields an optimized, reliable project plan.

ADDITIONAL PAYOFF

Setting realistic schedules also improves competitiveness. When guided by realistic schedules, project teams usually work exceptionally hard to meet or beat them. That translates into greatly increased productivity and throughput, which results in shorter cycle times and lower development costs, paving the way to best-in-class status. Realistic schedules and staffing requirements also reduce the need to cut corners during development to meet the target schedule; which translates into a higher quality development process resulting in fewer re-spins. Realistic schedules and staffing also eliminate or reduce burnout, which helps minimize staff turnover. The net result is an improved financial bottom line.

In conclusion, the benchmarking methodology prescribed in [Benchmarking IC Development Capability-Why?](#), enables semiconductor organizations to determine whether their IC development processes are truly best-in-class. What’s more, companies can easily build on it to create streamlined processes and an underlying infrastructure that allow them to quickly generate reliable, fact-based estimates of cycle times and staffing requirements. That, in turn means better risk management in portfolio planning. Thus, companies can use the methodology as a springboard to rapidly improve their financial performance.