Examining the Criteria for
Best-in-Class
IC Development Process

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

Defining Best-in-Class

To win customers and dominate markets in today's fiercely competitive global environment, it's critical that semiconductor companies and chip development organizations deliver complete solutions. These solutions are often comprised of multiple chips, software, reference designs, development boards, and more. To gain a decisive edge it's crucial that these companies be best-in-class in their product development capability.

Unfortunately, "best-in-class" is bandied about far too easily, with no clear definition. For some, it means having the shortest cycle time or first-silicon success. For others, it's a very low spin count, the highest productivity, or some other criteria. At first glance, each of these metrics seems valid. However, none bears up under scrutiny as a viable, *stand-alone* definition of best-in-class, even though each is meaningful in its own right.

Very short cycle time, for example, means fast time to market. But what if a team achieves short cycle time by continuously adding engineers to the project? Should we view the resulting short cycle time as best-in-class? The team had an unfair advantage—a higher-than-average staffing level. Moreover, what if this large team had below average productivity, but its sheer size enabled it to accomplish more design work in a short amount of time?

It doesn't seem appropriate to say this team is best-in-class, because the consequence of excessive staffing is a higher development cost for a given chip, as well as the consumption of resources that are then unavailable for other projects. In other words, if a project sacrifices development cost and takes resources away from other projects in order to achieve very fast cycle time, can we really say that the team is best-in-class? The answer is no. Shortest cycle time can define "fastest in class," but not necessarily best-in-class.

Likewise, what if a chip development organization has the highest first-silicon success rate and the lowest average spin count—does this mean that it's best-in-class? Both are important metrics, but each represents only one view of the project. Moreover, they don't necessarily translate into fastest cycle time, highest productivity or lowest development cost. What if the team sacrificed cycle time and development cost to achieve its high first-silicon success rate? Just because an organization boasts spin count superiority doesn't mean that it's best-in-class.

What about using productivity—output per person-week—as the measure of best-in-class? At first glance, this seems intuitively correct, but there's a problem—a small team typically achieves high productivity far more easily than a large team because there's minimal overhead expended on communication, consensus building, decision making, and project management. On the other hand, small teams almost always produce much less *output* than large teams during an equivalent period of time—even if the small team's productivity is much higher—because small teams simply lack the staffing to do all the necessary tasks.

For instance, consider a design team of five experienced engineers working on a fifty-million transistor chip. It would take them many years to finish the design. In contrast, a team of 50 to 100 engineers could probably finish the job in less than two years. So, if a small team is tasked with developing a large, complex chip, the likely outcome is high productivity among team members but an unacceptably long cycle time. High productivity, although highly desirable, doesn't determine cycle time, as team size must be factored in as well. Many projects exhibit high productivity, but these are often smaller designs that generate low revenue or low profit margin. Therefore, simply having high productivity can't stand up as the definition of best-in-class.

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 semiconductor and electronics companies across the industry, Numetrics (NMX) has observed that two key metrics viewed *together* are extremely good indicators of best-in-class: *development productivity* and *development throughput.* **Throughput** is the rate at which a team, as opposed to an individual, accomplishes its work—output per week for the whole team. Whereas **productivity** is output-per-individual-team-member-per-week.

Projects that demonstrate higher-than-average productivity and throughput nearly always boast shorter cycle times, fewer spins and better schedule performance. They also have a higher probability of achieving first-time silicon success compared with projects that have lower productivity and throughput. Not surprisingly, the combination of high productivity and high throughput typically translates into very competitive development costs as well. These conclusions are supported by data from more than 1,400 IC development projects in the NMX Industry Database (see table below).

Metric	Comparing Best-in-Class with Average Projects
Development Productivity	232% more productive
Development Throughput	354% more output
Project Cycle Time	23% shorter
Schedule Performance	51% better
Silicon Iterations	10% fewer spins
1st Silicon Success Rate	111% higher success
Development Cost per Unit of Output	68% less

The greater the number of team members, the more the throughput per individual (per week) typically falls because of the additional overhead of the larger team.

Combining Productivity and Throughput to Determine Best-in-Class

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) by above average and below average productivity, yields four quadrants:

(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), and
(4) High Throughput and High Productivity (the upper right)—
 the region where the best-in-class reside

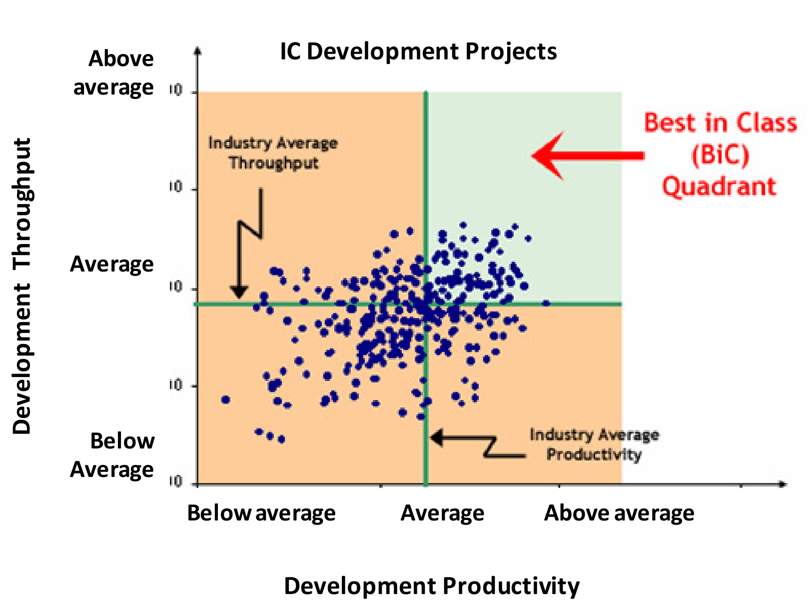


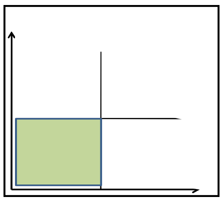
Figure 1: Identifying Best-in-

By itself, plotting throughput versus productivity doesn't make it clear which projects have achieved best-in-class status. Dividing the graph by the average throughput and productivity yields four views of performance. It's easy to be in the upper-left quadrant if you put together a very large design team or to be in the lower-right quadrant if your team is very small and skilled. Projects in the lower-right quadrant exhibit high productivity, but in most cases their low throughput puts them at a serious disadvantage.

The upper-right quadrant indicates best-in-class projects: those that are highly productive and have high throughput, even though their teams are typically larger than average. Achieving both simultaneously is extremely difficult. For a small team it requires enormous productivity from each individual on the team. For a medium or large-size team, it requires outstanding management skills.

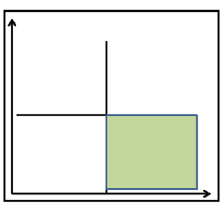
This framework enables us to identify best-in-class projects, as well as to categorize IC designs based on development goals and business strategy. For example, a particular market opportunity and set of business conditions might warrant that a project be in the upper-left quadrant. Likewise, a different market opportunity and set of business conditions might warrant being in the lower-left quadrant.

Quadrant 1: Worst-in-Class



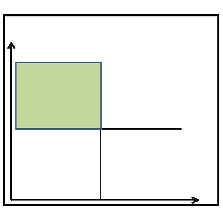
Obviously, nobody wants to be in the lower-left quadrant—below average productivity and below-average throughput—which could be called "worst-in-class." Low productivity can often be explained if the team size is large, because the communication and project management overhead would likely be high. Although not ideal, low productivity is often a reality of large teams. It's often acceptable only if the larger team yields above average throughput. However, low throughput for a large team is an unfortunate but common outcome that must be confronted. It often means that there is excessive overhead, mismanagement of some kind, and/or that the team members are just not up to par. Although the large teams produce greater overhead, they should be able to achieve a high throughput—in other words; they should be able to achieve "Quadrant 3" status (high throughput but low productivity).

Quadrant 2: High Productivity, Low Throughput



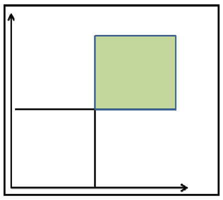
It's often easy to be in the lower-right quadrant. You can be highly productive if you assemble a small team of skilled engineers (e.g. to work on a small project). Less time and effort is consumed on communication, consensus building, and decision making. But a small team can produce only so much output in a given period of time, even if it's highly productive. That is, its throughput is limited, which negatively impacts cycle time. So if a small team works on a large complex project, the result could easily be high productivity but low throughput, which means a long cycle time.

Quadrant 3: High Throughput, Low Productivity



The upper-left quadrant is the inverse of the lower-right. High throughput is often achieved by assigning significant resources to a project (if you have that luxury). Typically, it's done for chips characterized by high volumes and large profit margins—in other words, chips that promise to generate considerable revenue and profit. Of course, a large staffing level translates directly into higher development cost—both overall cost and cost per unit of output. That's usually justified by the high volumes and large margins, but it doesn't make sense for most other chips, which are much more cost-sensitive.

Quadrant 4: Best-in-Class



The quadrant on the right represents the best-in-class. Achieving it poses a formidable challenge. As noted earlier, high productivity is generally attainable if you have a small development team, and high throughput is generally realized if you have a large team, but achieving both simultaneously is very difficult because productivity declines as you add members **(see Figure 2)**.

Productivity vs. Team Size

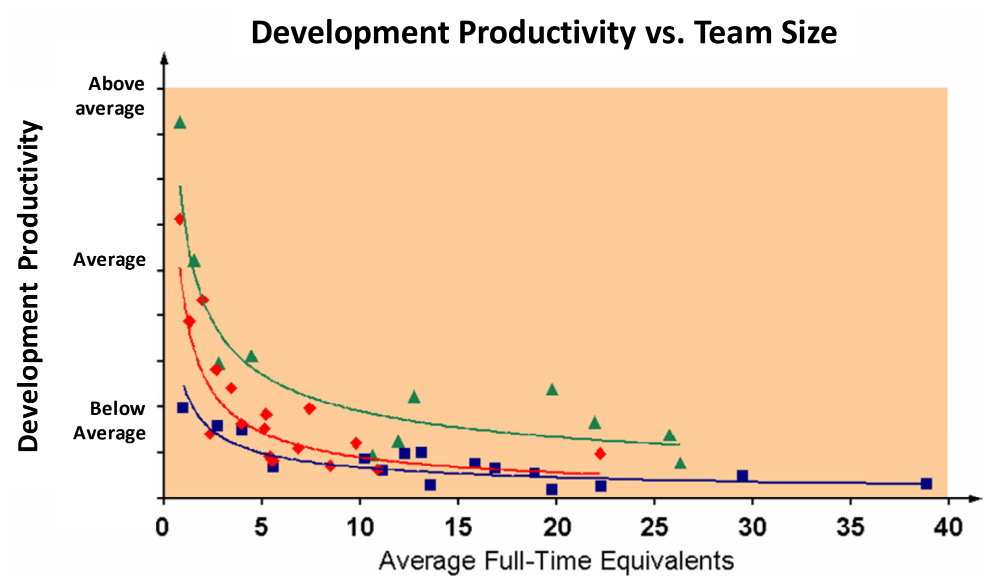


Figure 2: Productivity Degrades as Team Size Increases

Average productivity of a team drops as the team size grows, because the overhead expended on communication and overall project management increases. Generally, the best-in-class possess superior management skills that minimize the degradation.

Here, Company A (green **triangle**) has approximately double the productivity—across the board—of Companies B and C. This is a hallmark of the best-in-class. Company B's (red **diamond**) small teams do quite well, but its large ones don't—evidence of weaker management practices and methodology. Company C (blue **square**) simply does poorly everywhere, so it's probably among the worst-in-class. The tight clustering of the data around the trend line indicates clear structural and methodological differences among the companies.

Achieving high productivity and high throughput most likely means the company is capable of increasing team size without significantly compromising productivity, which is generally a reflection of great product development capability, as well as superior management practices, including project planning, risk management, portfolio management and program management. That's why only the best-in-class can achieve high productivity and high throughput simultaneously.

Furthermore, because the development cost per unit of output for a given chip falls as productivity rises and, similarly, the cycle time for a given chip shrinks as throughput increases, best-in-class projects score high on both of those metrics, as well and virtually all other key performance indicators. 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.

In fact, best-in-class projects and companies are like championship baseball teams. Championship teams almost always score at or near the top in two key team metrics: batting average and pitching capability (earned run average). When this occurs, we see the team typically performing very well in other key areas—home runs, runs-batted-in (RBIs) and fewest errors.

Summary

IC development organizations throughout the industry strive to be "best-in-class." But having a commonly accepted industry definition of what constitutes best-in-class is critical to success. As stated above, Numetrics has found that best-in-class demands achieving *high productivity* and *high throughput* simultaneously. Moreover, 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. But only when productivity and throughput are rigorously quantified can an organization or project team claim to be best-in-class. Numetrics' quantitative approach shows that being best-in-class requires superior IC development capabilities, as well as superior management practices.