

# IC Development Capability— A Prerequisite to Portfolio Planning

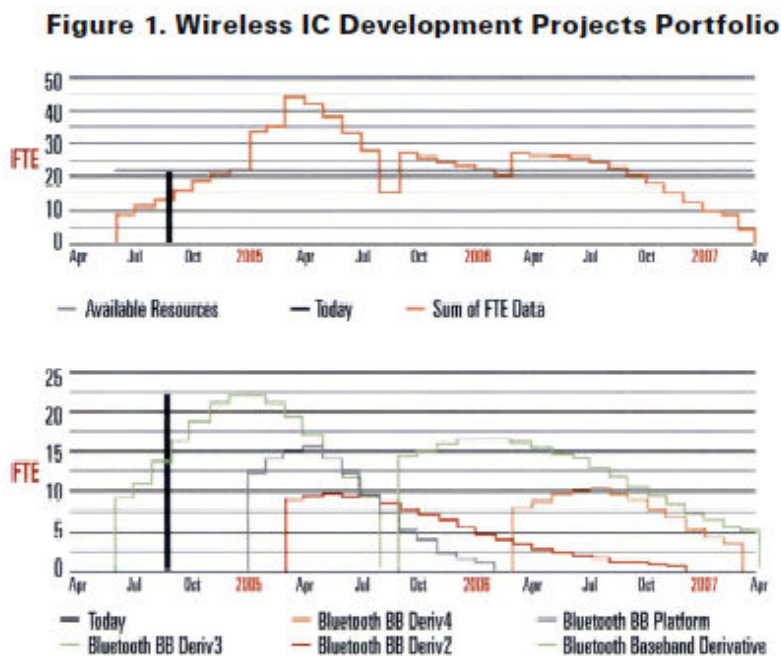
## White Paper

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](mailto:info@numetrics.com)  
Web site: [www.numetrics.com](http://www.numetrics.com)

## INTRODUCTION

A handful of chip companies are quietly and steadily establishing what portends to be a significant competitive advantage in the semiconductor industry. They're boosting chip development productivity, cutting cycle times, reducing spin count, improving schedule predictability and making optimal use of their engineering resources. How are they doing it?

Naturally they're investing in electronic design automation (EDA) tools, identifying product development bottlenecks and combating specification churn, but they are also complementing these traditional strategies with new thinking on portfolio planning (which includes on-going re-planning and optimization). Weak portfolio planning is an insidious problem bleeding millions of dollars annually from the financial bottom lines of companies across the industry—in many cases it's even tens or hundreds of millions. Financial losses occur when products are late to market and/or engineering resources are squandered on IC development projects that get cancelled. Both are symptoms of major flaws in the portfolio planning process, including insufficient staffing of projects and poor planning of the project execution pipeline. Figure 1 shows an example of a portfolio whose net present value (NPV) will fall short of expectations because of poor pipeline planning.



The table shows a portfolio of five wireless IC development projects whose net present value (NPV) is \$491 million—assuming they each finish by their planned End Date. However the top graph shows that the aggregate headcount consumption (FTE—full time equivalents) exceeds the available headcount (constraint) during several intervals of time. Therefore, the forecasted NPV will likely fall far short of expectation, as schedules will slip and some projects may end up getting cancelled because they will have missed their market window.

would deliver the expected boost. Both are critical, but neither has delivered the sought-after “unfair” competitive advantage. They've merely allowed companies to remain competitive, not achieve an advantage.

At first glance, other initiatives aimed at cutting cycle time and reducing project cancellations might seem to be potentially more efficacious than portfolio planning. In fact, IC companies have been virtually obsessed with the idea that competitive advantage, especially time-to-market advantage, should come from quantum leaps in product development productivity. Although they were right in focusing on productivity, they were wrong in thinking that initiatives such as improving design automation infrastructure and battling spec churn

## PATH TO COMPETITIVE ADVANTAGE

Until a few years ago, many chip makers gave portfolio planning modest attention, thinking that it was simply a matter of targeting the right markets and arriving first with products that outshine the competition. That was the beginning and end of it. Although such thinking persists in some companies, myriad others now recognize that it is much more. They are finding that sound portfolio planning practices, especially schedule predictability, lead to increased development productivity, shortened cycle time and fewer spins.

An industry-wide study of IC projects achieving best-in-class schedule predictability—those with 20 percent or less overrun—shows exactly that (Figure 2). Therefore, pursuing best-in-class schedule predictability, measured as (1) the percentage of projects that miss target schedule and (2) the difference between each project’s originally planned and actual cycle time, seems to be the true path to competitive advantage in the semiconductor industry. Moreover, an organization that has best-in-class predictability but merely average productivity will be a fierce competitor. Its advantage will derive from competitive cycle times across its entire portfolio, because engineers will roll from one project to the next as schedule . Achieving best-in-class schedule predictability means that an organization has control over its product development process. Furthermore, strong correlation between schedule overrun and development productivity suggests that perhaps development team performance improves if schedule targets are truly achievable, which would be the case if the appropriate level of staffing is allocated to each project.

## PORTFOLIO & PIPELINE: INEXTRICABLY INTERTWINED



The dials compare the execution performance of projects whose schedules were reasonably well forecasted against those that were not. Development productivity of projects overrunning their originally planned cycle time by more than 40% is approximately 30% lower than those that overrun by less than 20%. Big differences in design throughput and development cost also point to the value of having good schedule predictability.

Portfolio planning indeed includes the discipline of targeting the right markets and developing products that meet customer requirements. But the simplicity of this description tends to mask the underlying tasks necessary to achieving success. Missing in virtually every company’s portfolio planning process is accurate estimates of IC development cycle

times and project staffing level requirements. Inaccurate estimates lead companies to fund too many projects—because their appetites don't conform to engineering headcount constraints. It's a ubiquitous problem whose implications are obvious. Spawning too many projects means each project's resources are stretched thin. When resource managers deprive projects of the staffing needed to hit target schedules, schedules are missed. In fact, this is the primary reason why 85% of all IC projects miss their schedule. Funding too many projects naturally means many are doomed from the start.

Poor schedule and staffing estimates result from the methods managers use to generate them. They are largely subjective and based on the intuition of the individuals creating them. Intuition was perhaps acceptable in the pre-SOC (System-On-Chip) era, but today it's inadequate because the consequence of schedule slips has never been more severe.

Two other factors often contribute to unrealistic estimates. Cycle-time requirements are frequently driven by customer demands, and arguably there is little room for negotiation. Likewise, resource allocation decisions are often driven by reasons other than what projects need to meet schedules, and again rarely is there room for negotiation. The combination leads to over constrained and, therefore unachievable, project plans—not enough staffing allocated to meet schedule targets.

Without realistic estimates at the individual project level, portfolio planning is almost meaningless. That's because staffing dependencies invariably exist among the portfolio's projects. When projects fail to finish on time, they impact other projects that depend on receiving those resources. Therefore, a portfolio is viable only if each project can be executed according to schedule (which includes some margin for overrun). This demands that the development pipeline—the sequence in which projects are scheduled to execute—is viable and therefore realistic. The key point is that the pipeline and portfolio are inextricably intertwined, and the development pipeline's viability depends on whether engineers can move to their next project on time.

Many well-established consulting firms preach the value of good portfolio planning and have built reputable consulting practices around it. They have developed and promoted ostensibly powerful processes for implementing it. But almost without exception, they fall short of their promise because they gloss over the reality that accurate project level estimates predicate good portfolio planning. Therefore, inability to accurately forecast resources and cycle time is often a fatal flaw.

## **SCHEDULE AND STAFFING ESTIMATES FAILINGS**

Why do semiconductor companies struggle to generate accurate estimates of project schedule and staffing requirements? There are many reasons, some of which are technical, but others often stem from corporate behaviors that tacitly encourage grossly optimistic estimates.

On the technical side, inability to quantitatively assess the design's complexity is a primary

reason for poor schedule estimates. Historically, engineering managers have not had an effective way to accurately estimate the complexity of a design project. Without a quantitative understanding of the target design's complexity, generating reliable estimates is virtually impossible.

An overly aggressive estimate of the development team's productivity is a second cause of optimistic schedules. Not only do managers lack a methodology for quantifying productivity, but their subjective assessments are also typically way off. Optimistic expectations regarding development productivity are very common in the semiconductor industry.

Poor calibration of design complexity and team productivity is a deadly combination. It gives rise to numerous project cancellations and schedule overshoot—the average across the semiconductor industry is more than 50% of the originally planned schedule. In other words, a project scheduled to complete in 14 months will finish in 21 months, on average. Even worse, the standard deviation is quite large.

Implicit in every schedule that a manager creates is an assessment of the chip's complexity and the development team's productivity. Both are indispensable to creating a schedule and staffing plan and therefore must be contemplated—at least to some extent. If disregarded, there is little rational basis for either schedule or staffing plan estimates. Stated another way, the reliability of a project plan withstands scrutiny only if these two factors are considered in its creation. So either implicitly or explicitly, and either using intuition or some other method, managers planning projects must always attempt to scope the complexity of the design and the team's capabilities. Yet they fail repeatedly because they lack a quantitative methodology, tools and readily accessible databases.

Articles 2 and 3 of this series ([IC Development Capability-What to Measure?](#), and [IC Development Capability-Achieving Best-In-Class](#)) described the approach that Numetrics uses to measure complexity, quantify productivity and generate cycle time and staffing estimates. The solution is encapsulated in a suite of software tools that has generated fact-based estimates of cycle time and staffing levels more than 500 production IC projects from several dozen major semiconductor companies. The underlying models are continuously refreshed and are based on mining an industry database comprising more than 1,500 IC development projects Numetrics benchmarked during the past 10 years.

## THE USUAL SUSPECTS

Many development organizations mistakenly believe that accurate estimates of cycle time cannot be generated because predicting particular events that will impact schedule is impossible. Predicting specific events is certainly unrealistic, but generating reliable cycle time estimates need not depend on soothsaying. Rather, it is enough to simply acknowledge that (1) events will likely occur and (2) collectively they will impact productivity. Project effort increases when events spur an increase in cycle time and staffing (or if staffing level increases but cycle time remains constant). Both reduce productivity. Following are examples of typical events impacting productivity and, therefore, cycle time:

- (i) Specification changes,
- (ii) EDA tool problems,
- (iii) Cell library problems,
- (iv) Manufacturing process technology instability,
- (v) Problems with reused IP blocks and/or their delivery schedule,
- (vi) Coordination among multiple design sites,
- (vii) Reliability of new design services providers,
- (viii) Experience level of the development team, and
- (ix) Whether key resources are: pulled from the development team; delayed in joining the team; or leave the company.

When an organization benchmarks its productivity, the calculated average reflects (1) how many of these events typically occur on each of its projects and (2) their collective impact on each project's effort. Of course it also reflects its overall design capability, including finding and eliminating design bugs. What's most important is recognizing that predicting which particular events will occur is unnecessary and extraneous to the fact-based planning methodology. Instead, the lynchpin of accurately forecasting cycle time and staffing requirements is estimating development productivity, which can only be accomplished through benchmarking.

An organization's productivity is typically determined by benchmarking a handful of completed projects. After calibrating productivity, engineering managers apply the results to estimating cycle time and resource requirements for new projects. They typically generate a full range of schedule and staffing scenarios for each target project, from best case to worst case. For example, a best-case scenario might assume that productivity will be 30% higher than the highest productivity achieved on prior projects.

Numetrics often finds that an organization's productivity is surprisingly consistent among projects, especially the productivity achieved when implementing large SOC projects, including those with substantial analog or radio frequency (RF) content.

Stable productivity means that even though particular events influencing the schedule vary among projects, their collective impact remains fairly constant. Consistent productivity also means that an organization has control over its development process, and this translates to best-in-class predictability and, therefore, reliable estimates of cycle time and staffing requirements. And, as noted earlier, not only is this critical for sound portfolio planning, it seems to engender best-in-class productivity, cycle time, spin count, etc.

## ROOTING OUT RESISTANCE

Resistance to the idea that managers can consistently generate accurate estimates of cycle

times and staffing requirements is also rooted in the belief that chip complexity, especially analog circuit complexity, cannot be accurately quantified. A clear explanation of how complexity is calculated plus tangible evidence of its accuracy dispels the belief, but the initial skepticism is natural because such calculations have been an elusive goal of the IC industry. Skepticism typically disappears once engineering managers apply the fact-based methodology (described in [IC Development Capability-What to Measure?](#), and [IC Development Capability-Achieving Best-In-Class](#)), which includes complexity calculation, to their own projects. Statistical evidence of the calculation’s accuracy complements empirical evidence, which comes in the form of an automatically generated ranking of the development difficulty of each functional block on a familiar IC design project.

When examining the fact-based approach to generating cycle time and staffing estimates, it’s important to distinguish productivity from chip complexity. There’s often confusion between the two, because chip complexity, or design difficulty, is naturally a function of the skill level and efficaciousness of the development team. An unskilled team will find a particular design far more complex than a skilled team. So a frequently asked question is how chip complexity can be separated from skill level. The answer is straightforward. As described in the second article of this series ([IC Development Capability-What to Measure?](#)), Numetrics’ complexity index is an industry-wide index, not a company-specific index.

It calculates complexity based on the average development capability across the industry and, therefore, is a reflection of the complexity as seen through the eyes of the (hypothetical) average development team. Productivity, on the other hand, is specific to the particular development organization or team assigned to the project.

**Figure 3. IC Development Schedule Predictability**



The charts plot IC development schedule predictability against two key performance indicators: cycle time and spin count. Projects over-running their originally planned cycle time by more than 40% have (a) cycle times that are 20% longer and (b) spin counts that are 25% higher than those over-running by less than 20%.

Numetrics generates its complexity index by determining the quantitative relationship between project effort and each “complexity attribute” by benchmarking completed chip designs. Examples of attributes include circuit types (analog, RF, memory, etc.), amount and kind of reuse, process technology, clock frequency, power consumption and so on. Project effort for each attribute is averaged across the industry using a statistically significant sample of projects.

Attributes included in the complexity index are those having a statistically significant impact on project effort, as verified through extensive data collection. Averages are recalculated on a regular basis to account for improvements in product development processes, design skills, tools, methodology, design style and changes in silicon technology.

## CONFRONTING A CULPRIT

Semiconductor companies have given much attention to reducing the number of specification changes on projects. Everyone is painfully aware that spec changes stretch out cycle times and chew up precious resources, the combination of which lowers productivity. Spec churn is a systemic problem that directly impacts a company's financial bottom line. But many companies now recognize and accept that spec changes are inevitable—because end-markets are in a continuous state of flux and specs must be subject to change throughout the development lifecycle. Not having the right feature set in a product can be tantamount to missing the market opportunity. Therefore, instead of just working to reduce spec changes, companies are instituting processes and infrastructures that enable managers to quickly determine the true impact that each change-request will have on schedule and resource requirements. Insight into the implications of a spec change is indispensable to effective spec change management and integral to on-going portfolio re-planning.