



Getting Mo(o)re out of semiconductor R&D

Excellence in the R&D function is only one piece of a world-class product-development strategy. Via a new diagnostic and jointly developed plan, we can significantly reduce time to market for new chips while also improving overall quality.

**Harald Bauer,
Felix Grawert, Nadine
Kammerlander,
Ulrich Naehrer, and
Florian Weig**

Research and development is the lifeblood of the semiconductor industry—so it is no surprise that R&D tends to be the highest-pressure corner of this high-intensity business. Much of this pressure results from the fact that time to market is a crucial metric for semiconductor makers: speed, specifically on-time delivery, is a key success factor in a market characterized by tight design-in windows, shortening product life cycles, and relentless price deflation. For some, Moore’s Law still sets the industry’s pace; ever-rising investments and technology challenges, such as rising chip complexity, are also a factor. For others, the core challenge is mastering system design, which involves integrating hardware and software blocks, as well as various films,

coatings, and layers, and ensuring they are customized to reflect customer and end-consumer preferences. Indeed, a number of leading-edge wireless semiconductor players now employ twice as many software engineers as traditional hardware engineers.

The impact on the industry is significant: for the top 20 semiconductor players, R&D costs have continuously risen and now account for more than 20 percent of revenues. The ratio of product life cycle to product-development time in semiconductors is half that for a mobile phone and a third that for an automobile. And for the growing ranks of “fab lite” or fabless players, R&D excellence is the key differentiating factor.

Many institutions have a history of time and budget overruns, in addition to late adjustments to product specifications from the R&D department, resulting in angry customers. At the same time, changing customer expectations and requests for specifications lead to many new projects in the pipeline.

Unfortunately, the problem extends beyond a single department. Our experience suggests that at least a third of the solution relates not to R&D but to interfaces with other functions: marketing, sales, production, and even supply chain. Consequently, any response must begin with a broader diagnostic of the product-development process, in which all involved functions are engaged to develop a holistic picture of the situation. For example, a team typically

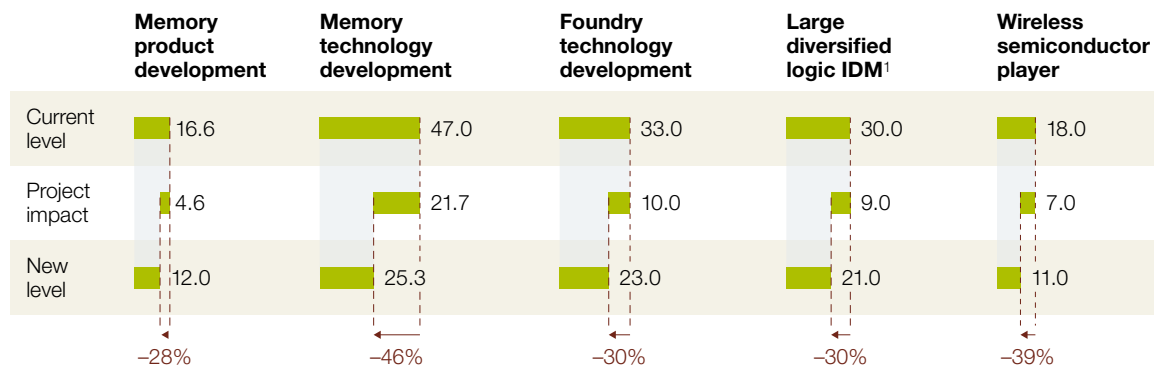
analyzes one or two end-to-end learning cycles (say, a new feature, a debugging exercise, or a technology process-of-record variant) within a single project; the team examines the process—including the initial idea, implementation, testing and analysis, and the final decision—while observing all touchpoints, loops, interfaces, and delays throughout the organization.

Our approach, which incorporates levers to improve product development and has been implemented with five clients (including fabless players, memory integrated device manufacturers, and a logic foundry), can make a sustainable and decisive difference in performance. On average, these clients reduced time to market by 30 percent and maintained or improved quality (Exhibit 1). We first conduct a thorough

Exhibit 1

A proper transformation program can greatly reduce development time.

Product-development cycle time, months



- 30% time-to-market reduction typical
- 30% effort reduction achieved by freeing up design team ahead of time

¹Integrated device manufacturer.

Exhibit 2

Specific challenges in integrated systems require tailored solutions.

Challenges	Approaches
<ul style="list-style-type: none"> • Insufficient breakdown of system features to requirements on individual hardware and software building blocks, leading to unclear development scope 	1 Structured requirements-breakdown process and tracking of breakdown via key performance indicators
<ul style="list-style-type: none"> • Serial development of hardware and software projects, extending development time 	2 Development model with parallel hardware and software development via virtual prototypes and step changes in development time
<ul style="list-style-type: none"> • Tests conducted only late in the development process and on the level of the entire system, leading to late identification of bugs 	3 Staged verification process starting with preverification on block level and a well-defined verification cascade, testing each module before integration into the system
<ul style="list-style-type: none"> • Separate projects and separate scope definition for hardware and software projects, despite need for close synchronization and interaction 	4 Joint definition of hardware and software projects and introduction of system milestones as “clock generator” that synchronizes hardware and software development
<ul style="list-style-type: none"> • Lack of accountability within the development of highly complex integrated systems that require the involvement of many experts 	5 Introduction of a hybrid organization with project deliverables handled by a virtual organization with functions responsible for delivery of work packages

diagnostic of the development process, and then we involve all affected departments in the design of a new methodology. Finally, we roll out the methodology, aiming to remove bottlenecks at various stages of the development cycle (Exhibit 2).

There is one important caveat: the approach deliberately focuses on product-development efficiency (number of products per input); it places less emphasis on effectiveness (impact of output products per input). It is, of course, better to develop a great product in an inefficient way than a poor product efficiently—but this is no excuse for poor efficiency. First, an efficient process simply provides the company with more outputs from the product-development process; increasing efficiency in time and effort by 33 percent allows the organization to develop 3 products instead of 2 using the same inputs. More important, by highlighting typical issues related to effectiveness,

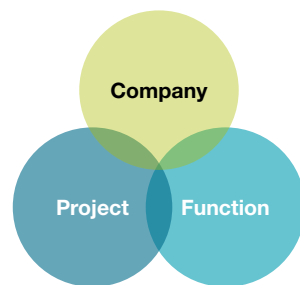
such as poor specifications or unrealistic time-lines early in the process, making processes more efficient can help identify changes that will improve effectiveness, too.

Designing a transformation program for product development

In general, the diagnostic looks at the end-to-end product-development process through three lenses: a company view, a line-function view, and a project view (Exhibit 3).

The company view focuses on basic higher-level facts about product development and the value at stake: R&D spend versus the competition; on-time, on-budget performance (versus the original plan) of projects considered both as an average and as a spread, or range; the basic organizational setup (that is, to determine whether the R&D footprint is fragmented inefficiently across several

Exhibit 3

A typical diagnostic considers three perspectives.**Relevant areas****Typical improvement levers****Company level**

- 1 R&D portfolio management
- 2 Organizational setup

Functional level

- 3 Predevelopment
- 4 Reuse
- 5 Definition of requirements
- 6 Development flow: methodology
- 7 Development flow: tool landscape
- 8 System for key performance indicators and output maturity

Project level

- 9 Resource allocation and capability buildup
- 10 Project management
- 11 Engineering efficiency and removal of bottlenecks

sites); and the general balance of the portfolio with regard to risk and time to launch. The team also reviews current key performance indicators and incentives for product development.

The line-function view looks at the way competence is created and leveraged in a sample of different projects. This includes an analysis of the quality of predevelopment, the reuse of architecture and intellectual-property blocks, the definition of requirements, and the basic development methodology and tool flow.

The project view assesses the quality of the competences deployed within specific new product-development projects. The team considers aspects such as project-management skills, especially in light of project planning, resource allocation, and links to capability-building programs and individual engineer productivity.

Viewing the product-development process from each of these perspectives allows the team to focus on strategic and tactical dimensions, highlighting improvements from the organizational level to the individual-employee level. The diagnostic is conducted by a group of consultants and client employees who represent all core functions and data providers. Both quantitative and qualitative analyses (for instance, structured questionnaires or surveys) are used.

The diagnostic yields many insights. For example, it is not uncommon to see several (or even all) of the following bottlenecks in a single company:

- The footprint of the R&D organization is driven by legacy and individual sites that are not aligned with an overall R&D strategy; sites lacking clearly defined missions are a particular problem.

- Performance varies significantly from project to project. The root causes are usually unclear, as postmortems focus on technical issues, not on project management, where problems more frequently lie.
- Predevelopment priorities are not transparent, so solutions are not mature, which leads to intensive follow-up work on live projects.
- The development methodology does not optimally distribute engineering work among the different involved functions. This may be especially problematic for companies that have recently acquired software competence—they tend to treat software as an element simply added on to hardware; such organizations must design a truly embedded flow.
- Milestones, also known as stage or quality gates, are defined but regularly missed; meetings are poorly attended and not formalized; and meetings often end without decisions being made.
- Planned reuse of intellectual property is often limited to one instance, as requirements for the next generation of products change in subtle ways.
- The project manager's overall plan is out of sync with a plan from a subteam manager. Individual engineers have no insight into how they fit into the picture.
- Star engineers may work on more than five projects at a time, and they may consequently be preoccupied with firefighting.

The picture that emerges is often sobering—but it can also be encouraging. The size of the opportunity becomes visible, and a straightforward change story emerges: introducing sound methodologies and new ways of working can eliminate substantial waste from products and reduce stress for engineers, project leads, managers, and executives alike.

Based on a quantitative assessment of the total opportunity and the contributions of the different levers that can be pulled, the team prioritizes core levers for the transformation program. There are typically two or three themes related to enablers (for example, site strategy, resource planning and deployment, and IT flow), as well as three or four direct levers (such as quality output of predevelopment, development methodology, and project management). When

Viewing the product-development process from each of these perspectives allows the team to focus on strategic and tactical dimensions

designing the transformation program, the team looks for a fair mix of line and project elements to ensure all stakeholders are involved in the change process.

It is also useful to touch different stages of the development process (for instance, predevelopment, the concept phase, the development phase, and ramp-up), which ensures that impact is widely visible. To quickly capture much of the opportunity, the team identifies debottlenecking adjustments among ongoing projects, rather than starting with one new project from scratch. These tactics help demonstrate broad impact early in the process, and they build enthusiasm to carry implementation forward.

Making change happen

Many programs to transform product development fail because the organization lacks a consistent method for the work, does not persist with the program as long as is necessary, engages in too much firefighting, and focuses too strongly on hidebound change processes. With regard to the last point, we generally insist that product-development transformations involve individual engineers directly or incorporate the views of colleagues whom the engineers respect. Although this is a basic rule of change management, it is essential in this kind of environment—after all, engineers tend to be skeptics until they see proof that a change will improve the situation.

As a consequence, teams overseeing the implementation of key levers must consist of both corporate management and important members of the technical hierarchy. Furthermore, they must act pragmatically at the outset of the program to convince colleagues of the program's worth by delivering quick wins. The first success factor will prove rather painful, because teams must

dedicate critical resources to improving the future rather than addressing day-to-day problems. (Whether this is happening in and of itself constitutes an important gut check for the project's sponsor.) The second success factor is almost as fraught, as immature innovations are deployed live and move forward in actual projects, not in sterile pilots. Given these factors, quick wins are crucial for building and maintaining morale.

A few examples demonstrate how clients have driven change in their product-development organizations. One client identified the need for a large-scale upgrade of its development methodology (for example, it needed to deploy a semi-automated design methodology more broadly, promote higher reuse, and significantly improve virtual integration). The change was debated fiercely in the expert community. The client organized biweekly town-hall meetings, to which a series of experts and project leads were invited to make presentations on the various aspects of the new methodology based on real project experience. The speakers included both supporters and opponents of the change; each was asked to display the pros and cons of the new approach. In the course of the series, the data increasingly showed the superiority of the new approach, and the engineering community embraced the need for change.

Another client needed to counter the lack of connection among the priorities of large projects involving more than 200 engineers with the tasks of individual contributors to these projects. The company introduced a standardized set of planning and alignment meetings to cascade through the project hierarchy, culminating in daily five-minute team huddles at the working-team level, to track progress against the weekly plan. Many, especially senior engineers, protested

the measure, because they felt it was a sign of mistrust and of a command-and-control attitude. However, the team leads were also trained to use the short huddles to identify roadblocks faced by team members that could be isolated and removed. Soon, many engineers had personally experienced measurable efficiency gains; this made the new way of working popular. For another company, a similar approach ultimately led to the development of a card for engineers to carry in their wallets that outlined the rights of an engineer on one side and the duties of an engineer on the other. Engineers were quick to point to their rights (which included having clear priorities, visibility into the project plan, and an efficient work environment), while team managers reminded them of their duties with regard to making work transparent, remaining committed to the plan, and speaking up about bottlenecks.

A fourth client had to deal with the ineffectiveness of its predevelopment department. The project team decided to organize a “predevelopment summit,” which brought together the company’s top 50 technical experts and R&D managers for a full day. In a series of preparatory meetings, the individual module and unit process owners pitched their proposals to a group of peers, project leads, and representatives from technical marketing and management. This helped them sharpen their proposals, anticipate questions,

and conduct initial prioritizations. The summit itself matched all proposals to the overall R&D strategy, forcing the technical experts to rank them (using an anonymous vote) before passing the final decision on to the management team. The predevelopment process was further improved by having the leader of the unit and the core program leads present together to the management team on progress each week.



These examples offer a glimpse of the innovation, rigor, and outreach involved in the improvement projects that make up a larger program. Although companies can generate initial success stories in as little as 2 to 3 months, it may take 6 to 10 months to realize the full impact from pilots. Given the time it takes for technology and product developments at larger semiconductor companies, it may be two to three years before the entire organization sees the total potential. By that time, however, the company will have profited significantly from the transformation, which will have freed up critical resources for more and higher-quality product development. ○