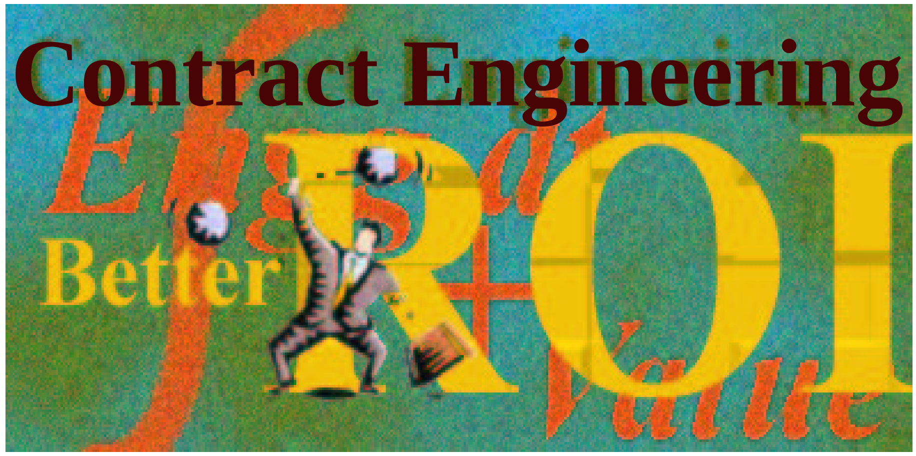
A WHITE PAPER



R. VIJAYARAGHAVAN

Competitive Positioning for the New Millennium:

Contract Engineering

This page intentionally left blank

Competitive Positioning for the New Millennium: Contract Engineering

By R. Vijayaraghavan

Executive Summary

All electronic product companies use Domain Knowledge and Enabling Technologies to build products. Architecting products for market niches, using their knowledge of application domains, is the job of the 'architects'. Building those products using the various enabling technologies is the job of the 'engineers'.

Traditional product development methods keep architects and engineers under one roof, often allowing them to cross their functional boundaries, due to the lack of a clear line of demarcation. This is the result of the legacy supply chain that is no longer suitable for the electronics industry, and its consequent faulty staffing process based on financial models, used by most companies. These models typically provide budgets for HR, infrastructure and other resources, as fixed costs. The organization first staffs up, and then figures out what work is to be assigned to whom; finds missing skills and scrambles for additional resources, or becomes inflexible and inefficient, eventually leading to periodic layoffs, and restructuring.

Contract Engineering saves money by sharing the cost of all enabling technology services with other users of such services from a Contract Engineering Company (CEC). This achieves conversion of the fixed costs of infrastructure resources to variable ones, based on the task on hand and paying for usage, under a contract of performance. It brings discipline to the product development process by isolating the architects, identifying product opportunities and clearly defining them. The new model offers several other advantages, including flexibility, speed, better ROI on R&D outlays, more stable employment to architects in product companies, and the same for the engineers of CECs. Most importantly, the ability to sustain consistently, a robust pipeline of new products is achieved by improved market and customer focus and simultaneous access to an efficient mechanism for Engineering Processes: the two features of an innovative supply chain.

Clearly positioned as one of the major macroeconomic trends, (see box on page 5: ('Efficiency Driven Value Migration') Contract Engineering will enable its early adopters to reap greater benefits, and build competitive advantage.

The leaders will position their companies for the next millennium, when the concepts of countries, governments, citizenship, employment, infrastructure services, and how we pay for them, will all change dramatically, in a 'value based' economy.

All electronic products, whether chips, boards, embedded software or systems, require two things to be built. These are:

1. Details of what functions the product will perform (called Application Technology)

2. Semiconductors like FPGAs and ASICs, EDA tools, flows, cores, targeted foundry technology, libraries, and software development environments (collectively called Enabling Technology)

This is, however, a particular view of the people who are responsible for product development from within a segment of the industry. A slightly broader view from the business perspective reveals that Application Technology merely defines the segment and is part of what we may call Domain Knowledge, including the details of :

- Customers
- Markets
- Products
- Features, and
- Competition

Similarly, Enabling Technology is part of a mechanism that is responsible for ***Engineering Processes***, including:

- ***Enabling Technology Tracking***
- ***Engineering Infrastructure Management***
- ***Training & Attrition Management***
- ***Project Responsibility, and***
- ***Process Guarantee.***

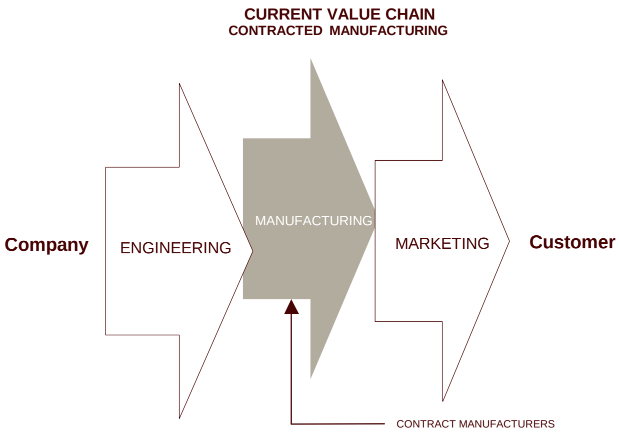


Fig. 1

Shift in Emphasis from Ownership to Efficiency

Product Companies have traditionally believed that they must own all stages of the value chain from Concept to Delivery. They have taken upon themselves all the tasks associated with acquiring their Domain Knowledge and maintaining engineering infrastructure/processes, and manufacturing capacity.

Recently, however, they have begun to let go manufacturing to specialists like Solectron (NYSE:SLR), Jabil Circuit (NYSE:JBL) and others. Contracting manufacturing out resulted in creating companies that are more responsive to market needs without having to worry about the manufacturing infrastructure and processes.

However, Product Companies still regard both Domain Knowledge and ***Engineering Processes*** as areas of their core competence.

For most companies, their current value chain looks like the one in Fig. 1. Engineering is bundled with concept and architecture and retained within the Product Company.

Therefore, the Engineering Processes still compete with Domain Knowledge for management attention, creating multi-dimensional demands on management bandwidth, as illustrated in Fig. 2.

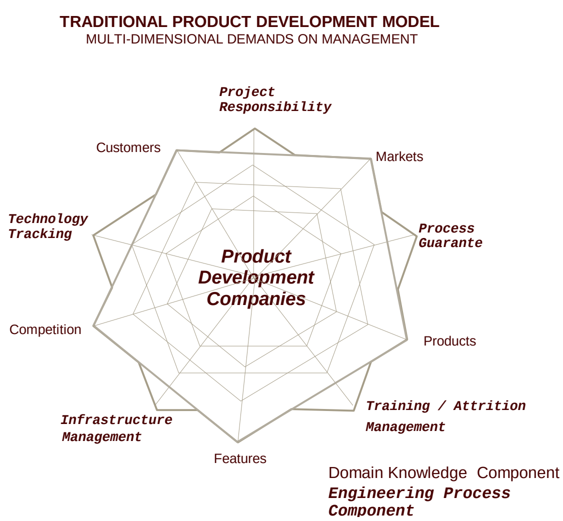


Fig. 2

The Case for Contract Engineering

Emerging research[1] in the area of Supply Chain Management recognizes that there can be two predominant types of supply chains: Primarily Functional and Primarily Innovative.

Functional supply chains cater to meeting repetitive, predictable demand cycles and are, structurally, replenishment oriented. An example is the supply chain management system for a chain of grocery stores. Functional supply chains are now fairly well understood, and can be dealt with through existing systems like MRP-II, ERP, etc.

Innovative supply chains, on the other hand, deal with managing unpredictable demand by responding quickly to rapidly changing market conditions.

The electronics industry does not appear to fit into a functional supply chain. Constant improvement in enabling technologies, product innovations and new product introductions create new demands and new

[1] See Marshall L. Fisher: *'What is the Right Supply Chain for Your Product?'* **Harvard Business Review, Mar.-Apr. 1997, pp. 105-116.** Reprint 97205.

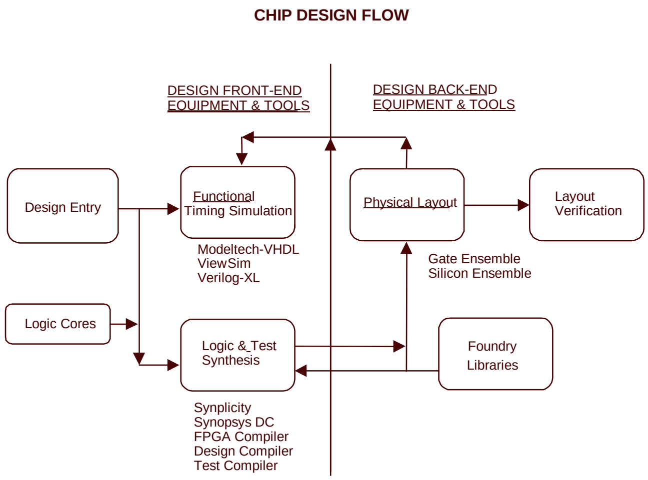


Fig. 3

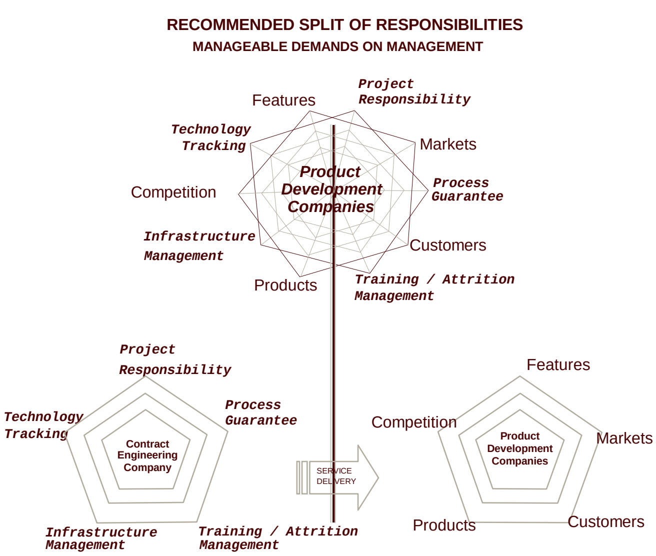


Fig. 4

markets, even as the old ones die. Product demand patterns and life-cycles in this industry are not consistently accurately predictable to fit into a traditional functional supply chain.

A large number of competing tools, flows, cores, FPGAs, and foundry technologies make it increasingly impossible for Product Companies to track, install, and maintain them, let alone train and re-train personnel.

In the area of Integrated Circuit chip design, for example, design flows have become extremely complex. A typical chip design flow is given in Fig. 3, with just a few of the competing tools mentioned.

Similarly complex flows and processes exist for designing boards, integrating chips, and other components into subsystems, developing and installing software, and having the entire system work smoothly. These activities are often interdependent, involving a fairly long chain of a large number of people.

Constant technology tracking and re-investing in design-center infrastructure are required, as newer technologies render older versions of tools obsolete. Training and attrition management of engineers are tasks by themselves.

For electronic/semiconductor products, development cycles are becoming longer than life cycles. Therefore, the ability to implement an innovative supply chain, i.e.:

- manage unpredictable demand well enough to survive and succeed in the marketplace and
- provide quick response to constantly changing market conditions

has become the primary imperative for long term survival.

The issue of managing unpredictable demand can be addressed by focussing better on Customers, Markets, Products, Features[2] and Competition**,** collectively defined earlier as Domain Knowledge.

The need for Quick Response requires an efficient mechanism for delivering effective Engineering Processes.

---

[2] **Products and Features:** Recent studies in creating market space (See W. Chan Kim and Renee Mauborgne*: 'Creating New Market Space'* **Harvard Business Review, Jan.-Feb. 1999, pp. 83-93.** Reprint 99105.) suggest that companies must focus on features to define their competitive positioning. Therefore, many times the features presented redefine a product, or create a new product or create new, originally unintended, uses and so on. This leads to a situation where features may define a product, as product used to define features. Hence the emphasis on both.

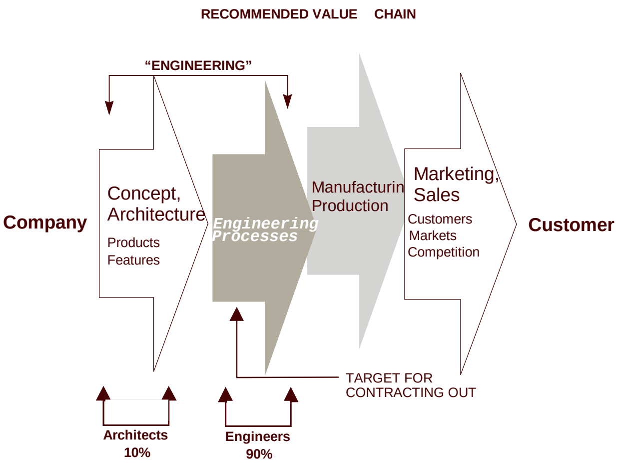


Fig. 5

The traditional product development model with its multidimensional demands on management has not been able to deliver on either, leaving Product Companies vulnerable to market forces.

Management is spread thin between the quick- witted and nimble-footed activities like understanding and responding to customers and markets on the one hand, and the essentially long gestation activities that require patient building, and nurturing of people and infrastructure on the other.

The problem of implementing an innovative supply chain can be reduced to a manageable level by creating Contract Engineering Companies or CECs (Fig. 4) that will split the responsibilities for product development with the Product Companies.

It will then be possible for the CECs to create an efficient mechanism for delivering effective Engineering Processes, just as the Contract Manufacturers did for the Manufacturing Processes.

The CECs' services are, therefore, Application Technology independent and apply well across the industry segments, whether it be computing systems, networking systems, or entertainment electronics.

The CECs must strive hard to demonstrate an effective engineering process that is transparent, by way of adherence to standards and quality. Equally important is their appreciation of the Application Technology involved for particular customers, in order to facilitate reduction in perceived risk of parting with what has, hitherto, been thought of by every Product Company as its proprietary process or core competence.

Product Companies must now examine their value chains, to identify how the value-chain components map to the recommended split in responsibilities that is required to achieve an efficient product development process. Identifying Engineering Processes and contracting them out is that next logical step (Fig. 5).

In the traditional value chain (see Fig. 1), companies tended to regard concept, architecture, and all the Engineering Processes such as design, development, and test engineering as one integral unit, called Engineering. This "unit" consisted of both the product architects (probably 10%), and the engineers (possibly the other 90%).

The Contract Engineering model attempts to split this perceived single unit into two distinct ones: *architecture* and *engineering*. It advocates that Product Companies focus on their Domain Knowledge to architect products,

Which Race are You Running?

Another way to think about the traditional model with its multi-dimensional demands on management is to liken it to a race. The long-gestation activities of resource building and maintenance are like running a 26-mile marathon, whereas attempting to succeed in the marketplace with a slew of new products in quick succession is like running a 100-meter sprint. If you are a track athlete, you had better make up your mind about which race you are running; otherwise one of your legs will be running one race, and the other, another. Would you expect to win either?

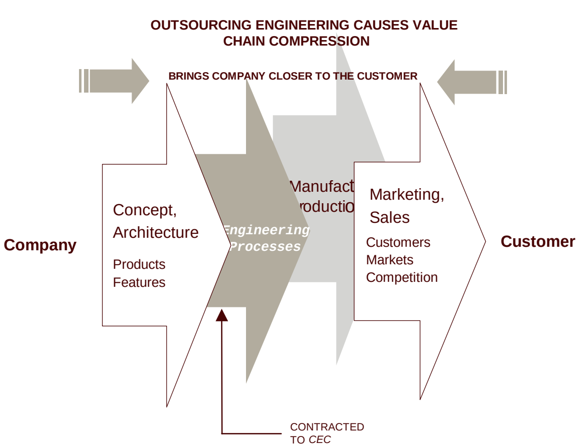


Fig. 6

and farm out the resource-intensive work of tracking and learning the changes and advances in the enabling technology areas, and the attendant responsibilities over engineering resources and processes.

Contracting out engineering causes value chain compression, bringing Product Companies closer to their customers (Fig. 6), resulting in improved market and customer focus, and greater ability to predict more accurately the demand patterns in their industry segments – the first half of what is required to implement an innovative supply chain.

As the Contract Engineering model becomes more prevalent, Engineers will get bifurcated into two categories, namely:

- Architects or the Product People, who use their Domain Knowledge, for product architecture

- Engineers or the Process People, who, with their knowledge of enabling technology processes, carry out engineering implementation.

The mature Contract Engineering model calls for a CEC separate from the company architecting the product, which will hire, train, retain and re-train the engineers and have them work with multiple architects from different companies.

The CEC will provide Product Companies with an efficient mechanism for effective enabling technology processes - the other half of what is required to implement an innovative supply chain.

Contract Engineering is, thus, aligned to a macro-economic trend, proven for other business processes, now moving to the engineering process of the electronic industry (See box: *'Efficiency Driven Value-Migration'*).

Efficiency Driven Value-Migration

Companies like Nike and Sara Lee are examples of successful product companies that don't make products.

Low-tech industries like Apparel Manufacturing have come the full circle to reclaim their first outsourced function, namely, "selling", either by retaining their own stores or by becoming 'direct merchants', or doing both. Motive: obtaining valuable feedback on customer needs and preferences. Low-tech industries also pay closer attention to their supply chains for survival, because of low barrier to entry. High tech companies entertain the illusion that 'technology' will protect them, though companies like Dell are supplanting other PC companies by focussing on the supply chain, measuring inventory in hours.

As enabling technologies become commoditized and entry barriers get constantly lowered, only innovative supply chains offer longer term protection. Knowledge of products, markets and customer relationships seem to be the only valid core competence. When that happens, product companies will be known as "Niche Owners". They will collaborate with others, like Enabling Technology companies, Contract Engineering Companies, Contract Manufacturers, Business Services Providers, Financial Services Providers, Health Services Providers, Governments, and other non-profit organizations (Fig. 7).

Any number of virtual links will be established amongst them. Even governments and non-profits must fall in line. Even as the pressure to cut taxes mounts, non-profits scramble to justify their existence. Hence the emphasis on 'same-store-sales' etc. for retail chains; franchise stores, of course, must stand on their own feet.

In the 'value based economy' every organization must become a unique value provider. Every member of society must belong to at least one such organization. This requires shedding of responsibility for functions that cannot be managed efficiently from within, 'spinning' themselves out to do whatever it is that they do best.

These entities are represented, in the diagram, in the form of spheres. Just as the sphere is the most efficient form to hold a given volume, these spheres of knowledge and expertise must be self-contained. Hence, whatever an organization's current form, it must become spherical. Otherwise, it will not exist.

Why is Contract Engineering more effective and efficient?

This is a reasonable question, especially because individual engineers are about the same whether they work in Product Companies or for a CEC.

Observations and analyses of several projects that we have been associated with over the years in the Silicon Valley, lead us to conclude that it is the *staffing process* that causes the problem.

The staffing process itself stems from the assumption that the engineering function, as part of our value chain, belongs in-house with product architecture. Therefore, the engineering infrastructure is regarded as part of the

EFFICIENCY DRIVEN VALUE MIGRATION

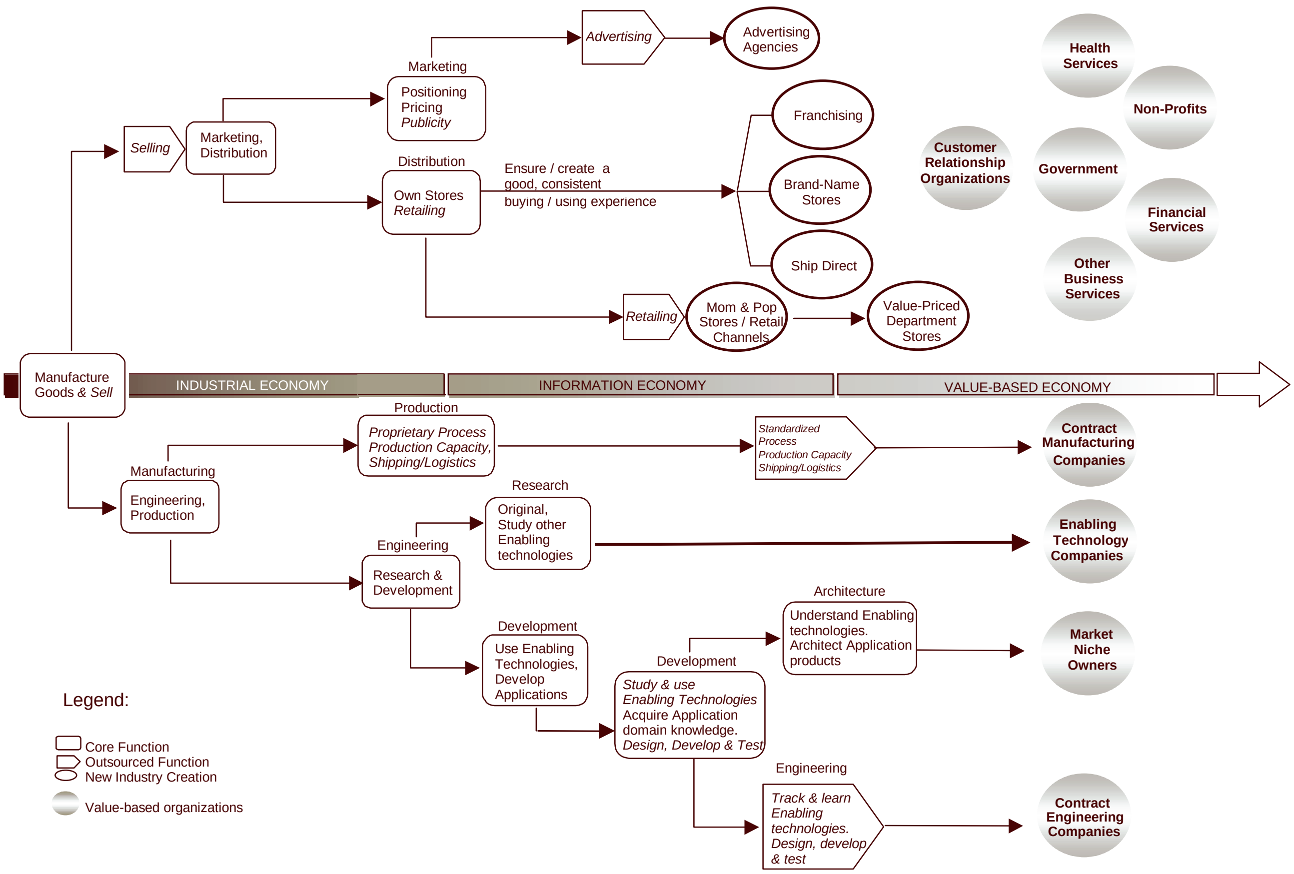


Fig. 7

in-house product development organization, treated as a fixed cost, and allocated a budget.

The Engineering group leader uses her best judgment as to what kind of skills will be needed, both architectural and engineering. Since they are usually under one roof, and work together without a clearly demarcated line of responsibility, the product development process is badly muddled. She makes compromises and trade-offs with what is available, hopes that, as a team, the group will deliver, and goes forward with the staffing process keeping the product development deadlines in mind.

Those who come on board are usually excited by the technology, and the opportunity to satisfy their own craving for technical experience etc. This group may include true architects, who either refuse to let go engineering, or carry that responsibility because the function is in-house, and the existing process dictates it.

The group also includes engineers, who are architect-wannabes. They have a legitimate desire for growth, hoping to participate in the architectural process of the product some day, again because the function is in-house, and the existing process dictates it.

There is nothing explicit about performance in their employment contracts, top down. Yes, there maybe some general understanding that product P must be launched by MMYY, oops, MMYYYY, but that is not part of the employment contract.

Therefore, the typical *contract of employment* reads somewhat like this: “We have bought 2000 hours of your time per year for the foresee-able future at $41.50 per hour; we may hike it at our option once a year, which we call merit increase; please show up for work at 8 a.m. everyday, and we will tell you from time to time what to do”.

An engineer, in this atmosphere, whether an 'architect' or 'engineer' may deliver what was assigned to her, but still, the product may not get out on time or work well.

On the contrary, a CEC enters into a *contract of performance* with its customers. The pay-for-performance system built into the Contract Engineering model ensures the success of contracted tasks. The tasks are usually well defined; if not, the CEC makes its customers define them in order to avoid failure due to non-performance. Thus, a CEC brings discipline to its customers’ processes. The CEC will then assign resources to the project based only on the task at hand and the schedule of delivery. The engineers of the CEC will now deliver based on this contract of performance. There is a measurable ROI, as payment is made against performance or progress. There also is a clear and present negative consequence of failure in terms of loss of future business, and bad references in the marketplace, for the CEC. Hardly ever are the consequences so severe for in-house teams in Product Companies.

Since every product needs different skills at different times in different quantities for its development, if we staff our in-house organization adequately for everything, we will be grossly inefficient. If we are too lean, we will never get the product out on time. In the mean while, Parkinson’s Law takes hold and all types of unproductive work is generated to keep the engineers busy.

Even if we create a perfect mathematical model to predict exactly what skill in what quantity is required at which time, it is not implementable under the current contract of employment, because, human beings are not machines that we can turn on and off at will. Therefore, no matter what we do, it is impossible to achieve higher levels of efficiencies in the current model of staffing for product development.

In other words, engineering process delivery through an in-house infrastructure becomes a high fixed cost, whether it is used efficiently or not. Contracting out engineering helps convert this high fixed cost into a variable cost, thereby enhancing the efficiency of the Engineering function.

Conflicting Goals

Product Development goals and Contracts of Employment are naturally conflicting, and work at cross-purposes.

Corporations have been, on the one hand, talking about the need for quicker turn-around and shorter time-to-market, and, on the other, offering traditional employment contracts. Yes, employment contracts are 'at will' in many states of the US. Yet, there is an unstated but implied contract, with every regular, full time employee, that has to do with long term association by way of matching/profit-sharing 401K plans, ESOP, training, nurturing, career development, and job satisfaction, that is much harder *not* to break.

The apparent and real contradiction between these stated and unstated positions is obvious, except to those who follow this model blindly. If not, it is soon to be found out when companies announce layoffs, RIF every three to five years.

The 'implied contract' needs are valid professional needs. CECs, however, can and must, meet them.

It is difficult for a CEC to reach 100% efficiency. There will, therefore, be time left, after meeting all service delivery obligations, for a CEC to do everything it can to retain and retrain its engineers, to be of long term value to its customers.

Surprisingly enough, the CEC can afford to have the traditional employment contract with its own engineers, because it works. Engineers indeed show up at 8 am, and know exactly what to do, for a project. When it is done, they know what to do next.

The hiring of fresh college graduates by Product Companies, for engineering tasks, is the ultimate betrayal of a good and effective product development process.

Hiring fresh college grads and training them, in various Engineering Processes, and some of the application domains must be the job of the CECs, to ensure adequate supply of trained engineers for the future. Systems engineering, whether it be hardware or software, can thus truly become a profession, akin to others like medicine, law, accounting etc, where a Post Graduate Degree plus 4 to 5 years of internship, training and specialization are mandatory, as a foundation of status as a professional.

When engineers have successfully delivered many tasks, many subsystems, and many products, they may, indeed, develop the necessary expertise to conceptualize new products and product improvements. Later, it is their alacrity, and commitment to their profession that will keep them upwardly mobile.

Meaningless Measures of Productivity

Many Product Companies highlight their R&D outlay as a percentage of Revenue to investors and analysts. A substantial part of this outlay is cost of engineering. Even successful companies do not wish this percentage to reduce. This has resulted in a higher and higher outlay for R&D.

This locker-room mentality has also invented questionable productivity measures, such as revenue per employee, routinely reported by several high-tech companies, now.

Skeletal research shows that for semiconductor and other high-tech companies, revenue growth is accompanied by growth in the number of employees, followed by a decline in revenue per employee after 4 to 5 years, to be followed by re-structuring. Some companies stop publishing this statistic, when it is going downhill.

First of all, measuring revenue per employee for a product company is nonsense. Yes, it is a measure, but not valid or relevant. It is similar to saying: “the ratio of the distance of my daughter’s school from my home to the length of my dog’s tail is 22567”.

Other than the fact that the ratio pertains to *my* daughter and *my* dog, it makes no sense. The dog can do very little about ‘improving’ this ratio! Marginal changes to this number can be made, if one takes different routes or shortcuts. My daughter has to move to a school farther away, to improve the ratio or I must get a dog with a shorter tail!

Similarly, setting up a product development process over which engineers (and many other employees) have little control and then measuring their productivity in terms of revenue makes no sense. Pray, tell us, how can an engineer in Bangalore, influence this measure in a $28 billion global product company?

Even when our processes are not clear, we end up making measurements of performance based on those processes.

This not only vitiates our thinking in terms of focussing our attention on making marginal improvements on those faulty processes, or becoming complacent when our numbers are better than the other company, but also prevents us from thinking originally, and constantly asking ourselves if our processes meet the needs of the ever changing market place.

Therefore, we must question: Why should revenue per employee be of the order of $200K to $511K? Why can't it be $2 million or more? When someone started publishing this number, others followed suit with theirs, especially if it looked better. As a relative measure it may be impressive, but is not a true indicator of what might be possible.

It is, indeed, possible, to change this measure dramatically, by reviewing current processes and their associated cost structures. Then, companies will be able to allocate resources depending on the projects on hand. Therefore, product development outlays will and must vary, based on what needs to be accomplished.

By measuring the yields of product people, in different product areas, we may get meaningful measures of how our people are doing, when they *can* do something to influence the outcome.

The tail, then, can wag the dog, if we may be permitted to mix our metaphors.

As for the engineers within a CEC, revenue per employee is a valid measure. An engineer's knowledge-base, currency of skills and experience are directly relevant to the revenue generated by that engineer.

The Intangibles

The CEC brings several other intangible benefits because there exists now an efficient mechanism for accessing effective engineering processes.

The most significant of these is the flexibility to adapt to changes in product development plans brought on by market forces. Putting a project on hold, moving one forward, or delaying yet another, is just a matter of a phone call, when we establish a long-term relationship with a CEC.

Earlier, with process infrastructure and related resources as fixed costs, we were either stuck with our excess resources, or scrambling for contractors, when we were short.

With a much smaller team of product people, we are now light-footed and nimble. We can move in and out of markets, and exploit product opportunities with speed. Luck, apparently, follows speed.

A Model for Contract Engineering

The CEC needed to realize this model is a vendor-neutral, independent Engineering Organization that will offer ***`Engineering Processes`*** at competitive prices.

It must be an engineering implementation center that clearly articulates the CEC's vision to be to Product Companies what, in effect, a civil engineering construction firm is to a firm of architects. To this end, it must establish full-fledged systems design centers that are structured for growth, invest in tools and infrastructure, and set up alliances with enabling technology companies. Its engineers must track and absorb the changes and advances in enabling technologies so that its customers can confidently buy efficient engineering processes guaranteed for success (Fig. 8). In order to aid understanding, as well as to capture the essence of what a CEC will do for its customers, the model has also been expressed as a formula, in Fig. 9.

THE CEC MODEL

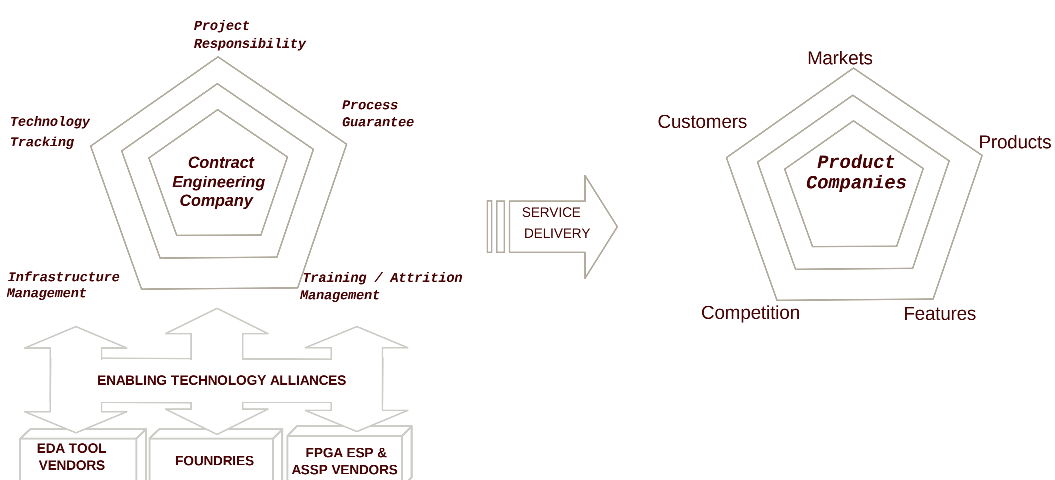


Fig. 8

$$\text{Contract Engineering} = \int_{t=\text{POST-DEFINITION}}^{t=\text{PRE-MANUFACTURING}} f\left[\begin{array}{l}\text{Engineering (Design, Development, Test)}\\ \text{Chips, Boards, Software, Systems}\end{array}\right] dt \; + \left|\begin{array}{l}\text{VALUE ADDITION}\\ \\ \text{Technology Tracking}\\ \text{Infrastructure Management}\\ \text{Training/Attrition Management}\\ \text{Process Guarantee}\\ \text{Project Responsibility}\end{array}\right|$$

Fig. 9

What are our options?

Maybe we are convinced that we ought to do something to our supply chain that will give us the ability to manage unpredictable demand, and provide quick response to constantly changing market conditions. Why simply adopt the CEC model? Why not look at all our options?

Our available options include the following:

- Temp Agencies
- Boutiques: usually owned and run by specialists well versed in a particular domain. The domain itself may be an application area like Wireless Networking, or expertise in a particular tool or methodology
- "Design Houses": a new breed of "Design Services" companies that are truly a cross between a Temp Agency and a Boutique, and offer engineering services in one or other area of product development.

We tend to assess these options on values such as Application Knowledge, Perceived Low Risk, Value Pricing, Methodology and their ability to take Project Responsibility

These traditional parameters are really those that we have been conditioned to look for in a services vendor.

It is now a recognized fact that companies don't compete, but supply chains do. In order for us to be competitive, our supply chains must acquire abilities to respond quickly to unpredictable demand. Failing to recognize that 'engineering' is part of the supply chain, because it is housed inside our own organization, we do not subject it to the same level of scrutiny and expectations, as we would any other component of our supply chain.

Table 1

<table>
<tr><th></th><th colspan="5">How our Options Compare</th></tr>
<tr><th></th><th></th><th>Temp Agency</th><th>Boutique</th><th>Design House</th><th>CEC</th></tr>
<tr><td rowspan="5">Traditional parameters</td><td>Application Knowledge</td><td>X</td><td>*</td><td>*</td><td>✓</td></tr>
<tr><td>Perceived Low Risk</td><td>?</td><td>✓</td><td>*</td><td>✓</td></tr>
<tr><td>Value Pricing</td><td>?</td><td>?</td><td>*</td><td>✓</td></tr>
<tr><td>Methodology</td><td>X</td><td>*</td><td>*</td><td>✓</td></tr>
<tr><td>Project Resp.</td><td>X</td><td>?</td><td>*</td><td>✓</td></tr>
<tr><td rowspan="6">Non-traditional parameters</td><td>Scalability</td><td colspan="3" rowspan="6">Not Addressed by Traditional Options</td><td>✓</td></tr>
<tr><td>Range of Service</td><td>✓</td></tr>
<tr><td>Process Guarantee</td><td>✓</td></tr>
<tr><td>Enabling Technology Tracking</td><td>✓</td></tr>
<tr><td>Infrastructure Management</td><td>✓</td></tr>
<tr><td>Training & Attrition Management</td><td>✓</td></tr>
</table>

* Offerings vary depending on individual company strengths

If we did, we would recognize immediately that Scalability, Comprehensive Range of Services in all areas of product development, including chips, boards, software and systems, Enabling Technology Tracking, Infrastructure Management, Training & Attrition Management and Process Guarantee are all the non-traditional parameters that we need to consider while evaluating our options.

Our supply chain requires services that are responsible, knowledgeable, on-demand, scalable and guaranteed, in all areas of product development.

The adjoining table summarizes how each of the traditional and non-traditional parameters is represented among the offerings from the various options.

The CEC model is unique in offering to us values that enable us to respond quickly to constantly changing market conditions. That leaves us free to focus on our domain knowledge to better predict the demand patterns of our marketplace.

Implementing the CEC Model

To effectively integrate Contract Engineering into their product development process, Product Companies must critically examine their current mix of Product People, Process People and their associated cost structures.

Whereas it is fairly simple to differentiate between architecting a product and dealing with the details of implementation, it is not always easy to identify our own product and process peoples.

Who are the Product People? As a simple rule-of-thumb to determine who might be our architects and who our engineers, the following has merit: Suppose tomorrow we were required to pay all our engineering staff 3X of what we are paying them now; who would we retain and who would we let go? The people who may be retained, even at 3X their current costs, are probably our product people. The others, possibly, are our process people. Skills in enabling technology areas are of little importance when we do this exercise.

The product people are those who spend a lot of time face to face with the customer, listen to suggestions for improvements and new requirements, study the competition, and apply what they have thus learnt to identifying market niches and conceptualizing products. They are the ones who are capable of writing a tight product specification document before they worry about how it is to be implemented. The process people usually tend to worry about tools, equipment, software environments etc.

In the Contract Engineering model we want to retain our product people and nurture them. If they are relieved of the burden of implementation they are likely to be a lot more productive, creating a robust pipeline of products.

Product Companies will, therefore, retain the product people component as a lower fixed cost, and convert the process people component to a variable cost and pay for it as a function of usage.

For the Contract Engineering model to work, not only must the CEC become the engineering implementers, but should also hire some of the process people, if that is possible.

As the world moves towards a mature Contract Engineering model, there will be an inevitable reorganization of the workforce, with the product people remaining with Product Companies and the process people moving to CECs.

Price Justification for Contract Engineering

In models that emphasize process ownership, companies tend to staff adequately, except perhaps, for peak loads.

The HR, budgeting and staffing systems, all of which emphasize cost over value, tend to promote the hiring of in-house employees. Departments and managers hang onto the resources once hired; furthermore, they defend the hiring and, over time, firing becomes harder to achieve, for various reasons. While it appears initially cheaper to hire in-house, an inevitable lack of productivity and efficiency has serious, long-term consequences.

There is, of course, a half-way-house solution currently prevalent. It is called hiring temporary workers. This is only an HR solution to a business problem. It merely makes the HR managers' jobs easier, as they don't have to go through the painful process of firing. It serves to circumvent hiring freezes and other corporate controls that may be in place. However, it does not bring in any value addition in terms of Project Responsibility and Process Guarantee as supervision and coordination of their work remains our responsibility. There is also no reduction in infrastructure costs, as they end up using in-house resources.

Unlike a mechanical machine-center in a manufacturing shop, which, when idle, can be easily spotted, as it is not spitting out whatever it is supposed to, human beings in front of computer terminals are constantly busy, as there is always some work to be done.

Furthermore, idle engineering capacity that crops up regularly while some process is waiting for another, cannot readily be reassigned to some other task because “it is not [their] job!”.

Therefore, it is harder to see inefficiencies in Product Companies. The price is eventually paid, when the company does not succeed in the market place, resulting in layoffs. This is the penalty that market forces extract for the error of conceiving as a fixed cost what ought to have been a variable cost in the first place. Sadly, restructuring, when it happens, is usually across the board, causing much unintended damage to the organization.

Let us say that an engineer costs $100,000 a year (50 weeks X 40 hrs), her direct, notional cost per hour is $50. If she operates at 40% efficiency, her real cost is $125 per hour.

The following table shows real cost versus notional cost, per hour, for a spectrum of direct costs and efficiencies:

Table 2

Efficiency	Direct Cost		
	$100K	$125K	$150K
20%	$250 vs. $50	$313 vs. $63	$375 vs. $75
30%	$167	$208	$250
40%	$125	$156	$188
50%	$100	$125	$150

We can conclude, in this example, therefore, that contracting out engineering at $100 per hour is no more expensive than doing it in-house, if in-house efficiencies are as high as 50%. We have not taken into account other indirect costs, like infrastructure costs. Indirect costs will easily justify a higher price. Fully loaded real costs tend to be closer to 3X.

Therefore, there exist reasons to pay anywhere from $100 to $200 per hour, or even more -- which is admittedly a lot more than the internal notional direct + indirect cost -- and still come out ahead. This is *very similar* to the theory of quality guru, Deming, who said, “…build quality into your manufacturing process starting from design and it will reduce your cost.” It seemed so odd and counter intuitive at that time.

The CECs, in effect, will be telling their customers “Pay us more, so that it costs you less.”

How does the CEC Remain Competitive?

If full-time employment contracts are invariably inefficient, how does the CEC deal with its own employees?

The CEC must track its engineering productivity avidly -- in terms of two productivity measures called NRPE (Net Revenue per Engineer) and NBT (Non-Billable Time). By constantly monitoring these two parameters, and by working with several architects at a time, it can keep its productivity high. As the CEC is relieved of the responsibility of watching the customers, markets, products and competition, its only job is to ensure its own high productivity for its own survival.

By voluntarily giving up the potential profits from product development, the CECs also avoid the risk of losses in the marketplace. The CEC's cashflow is dependent, not on the success of the product it helped develop, but in successfully delivering the project and getting paid for its work. This is not to suggest that the CEC will do anything less than its best to ensure its customers' success.

As the Contract Engineering model matures, companies will sign multi-year contracts with their CECs. The CECs, assured of their revenue streams, will ensure profitability even at lower gross margins because of higher productivity, and will pass on the resultant savings to their customers.

By contracting out engineering, Product Companies make a smaller number of people dependent on the success or failure of the product in the market place. All the risks, as well as the rewards, go to the architects, managers and investors, who by virtue of their experience in markets, management and finance, are better equipped to handle the vicissitudes of the marketplace.

The working life and career growth of engineers, then, are likely to be smoother, even as they move from project to project, as long as they keep their set of skills current.

What About the "Control" Issue

We have heard this one before.

The Contract Engineering model's argument is that Product Companies have more control with the CEC than with in-house development, as a contract provides a lot more tools and remedies such as guarantees and penalties to ensure performance. How many times have we fired our own development team purely because the product was late?

What is our Core Competence, really?

As for 'core' competence, what is it, any way? Is it the knowledge of customers, markets, product definitions, competitive forces -- collectively defined earlier as Domain Knowledge, or, is it VHDL, Verilog, synthesis, simulation, libraries and software environments -- the enabling technologies?

The answer is obvious: the enabling technologies became part of our core competence because of our current product development process.

Our knowledge of, and relationships with, our customers and markets are special to us. This is, or ought to be, our core competence.

Everyone who can afford it has access to enabling technologies. As enabling technologies become cheaper, and more prevalent, upstart competitors will use the latest technologies to wage guerilla warfare against established Product Companies, who are locked into the older, clunkier technology infrastructure. While technology does act as a barrier to entry, for long-term survival, the ability to adopt enabling technologies and respond quickly and effectively to changing market conditions becomes critically important.

Therefore, for the already established, their real power is the ability to see the needs of their customers and the developing trends in their markets sooner than anyone else. For a new entrant, their strength ought to be the ability to get that product out without wasting time on building a huge infrastructure; and, without having to worry about monitoring, nurturing and protecting it.

What About Security Issues

Security is a legitimate concern in contracting out sensitive engineering projects: Who else is looking at my stuff? Security is no more a concern with a CEC than when we are working

- either, with any in-house employee who may leave
- or, with a contractor who, despite non-disclosure agreements, owes us no loyalty

In the Contract Engineering model, the CEC's very survival depends on being able to maintain client confidentiality. A CEC's reputation needs be sullied only once for it to go out of business. The CEC, therefore, has great incentive to preserve the confidentiality of customers' intellectual property. Any violation of client confidentiality will be at great costs to a CEC, including a potential shutdown.

We believe, going forward, CECs will begin to offer specialized services that address this issue. These services may include on-site development centers, dedicated sub-nets, dedicated clean rooms, dedicated account managers, open-office access, open audit processes, etc. creating more tightly coupled customer relationships.

Why is Vendor-Neutrality Important

Earlier in this paper we described the CEC as vendor-neutral. We also mentioned in the same breath that a CEC must set up alliances with enabling technology companies. How vendor-neutral can a CEC remain while it is setting up all these alliances?

In the initial stages it is hard for everybody concerned to understand that the CEC has really no axe to grind. In order for the Contract Engineering model to truly mature, the CEC has to be vendor neutral. The CEC must offer several competing enabling technologies to its customers. The customers may even dictate the choices due to legacy reasons. Therefore, the CEC has to be somewhat impervious to the competitive and market forces acting on its alliance partners.

In the short run the CEC may have to work with one or more tool or device vendors who are willing to work with it. However, as the model matures, all device vendors, tool providers and silicon fabricators will get treated as equal, just as the computer retailer today carries several competing brands of computing equipment.

The retailer's salesperson is knowledgeable about the competing brands, and may point out their differences in features. The choice depends on several factors including application, customer preferences, suitability, cost, etc. The shelf space that the retailer provides to the manufacturer, and the consequent mindshare that the manufacturer enjoys among the purchasing public, has a close parallel in the mature Contract Engineering model. Some enabling technology companies may choose to spend a lot of time with the CECs, educating them on their latest tools, devices, libraries and processes. Chances are that the CEC's engineers will, consequently, be more knowledgeable about their tools, devices and processes than those of the others who do not work with the CECs as much.

In the long run, the CEC's engineers will have acquired a lot of knowledge on various enabling technologies, and can better determine their applicability to a particular requirement.

Nothing, absolutely nothing, when we do engineering in-house, can provide the same breadth of experience of enabling technologies and tools available as when we work with a CEC.

What is the Process of Evolution for the CECs

One must wonder whether the CEC will survive in the long run.

As business models go, the closest to a nascent CEC, is a Temp Agency. In this model with very low fixed costs, one would make money by simply matching the HR costs to the revenue, back-to-back. See Fig. 10 (TA).

However, as the CEC grows, builds and upgrades systems infrastructure, hires, trains, retains, and re-trains human resources, what is variable cost to a Temp Agency will become fixed cost to a CEC. To this, one must add the amortized cost of systems and physical resources. Then, a CEC's Design Center will have to get staffed to a certain level to match the perceived market demand in that place. Now the CEC resembles the in-house engineering organization.

The CEC will make money when revenue goes above the fixed cost, and lose money otherwise. Overall, it may make money in any given year, purely depending on its overall productivity. See Fig. 10(CEC 1). This is the most telling truth of the CEC model and evidence that the CEC does, indeed, mop up the inefficiencies of its clients' processes. As customers begin to value their services, CECs will sign multi-year contracts, and staff up adequately to meet those obligations. This will assure a steady revenue stream, making them more profitable. See Fig. 10 (CEC 2).

As the model matures, CECs will, no doubt, attract competition. Competitive forces will make the more established CECs cut prices, and/or offer enhanced levels of service such as on-site development centers, dedicated sub-nets, dedicated clean rooms, dedicated account managers, open-office access, open audit processes, etc. See Fig. 10 (CEC 3).

There would be other developments and consolidations, including mergers, eventually creating a mature Contract Engineering Industry, akin to Advertising, Contract Manufacturing, Civil Construction, Accounting-Audit Firms, and so on.

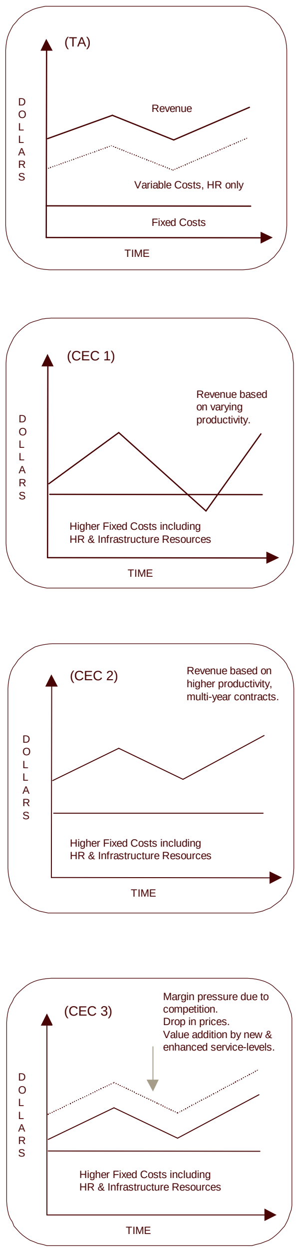


Fig. 10

Where does your Company stand vis-à-vis The Contract Engineering Model

For a quick feel of how things stand, for Product Companies that build application systems, we recommend taking a couple of minutes to fill out the questionnaire in Exhibit A.

The questionnaire comprises 30 questions. Against each question, check one of the boxes A through E for responses as they apply to your organization. Keys for responses A through E appear at the top of the questionnaire. Follow the instructions in the questionnaire to complete lines 31 through 34, to arrive at your final score.

If your score is less than or equal to 75, you probably have nothing to worry about. You may still consider contracting out engineering to build competitive advantage. If your score is greater than 75 but less than 120, you have a case for Contract Engineering. At scores of above 120, you may need to find a Contract Engineering partner fast.

For a more rigorous exercise, and if you can access the required financial data, you may use the worksheet in Exhibit B.

*   *   *

Large multi-billion dollar global technology companies, that are in the business of both creating enabling technologies, and building products, have a even harder time achieving efficiencies in their processes because of their attachment to their own enabling technologies, seriously mixing up their technology, process and product groups.

They first need to bring in clarity to their business models[3].

All organizations, whether an independent company or part of a conglomerate, must realize that one's Domain Knowledge maybe another's enabling technology. Clearly understanding one's own customers, markets, products, features and competition and using this knowledge to correctly position one's organization for the next Millennium seem to be the only core competence that leaders must strive for.

Everything else can be obtained from other entities, in a value-based economy. ■

[3] See John Hagel III and Marc Singer*: 'Unbundling the Corporation'*, **Harvard Business Review, Mar.-Apr. 1999, pp. 133-141.** Reprint 99205)

EXHIBIT- A

A= To a very great extent B= To a great extent C= To some extent D= To a slight extent E= To a very slight extent
Against each question, check one of the boxes A through E for responses as they apply to your company. Follow the instructions on the questionnaire to complete lines 31 through 34, to arrive at your final score.

NO.	SYMPTOMS	A	B	C	D	E
	DELAYS, COSTS, BUDGETS					
1	Specs for any product are not 100% complete either before OR after design implementation					
2	Add-ons to specs are done until the last minute					
3	Design or code freeze is a moving target					
4	Budgets/schedules are constantly revised upwards/ forwards					
5	Additional, unplanned capital requests are routinely made					
	PROJECT MANAGEMENT					
6	Managers feel a lack of “right” skills in the team					
7	Important but not “urgent” projects or features never get done					
8	High personnel turnover					
9	Projects are always in the crisis management mode					
10	Products are not fully tested before they are released					
11	High volume of initial field problems					
12	Test plans are afterthoughts					
13	QA complains that architects do not release specs on time					
14	Bug finding and bug fixing rates do not converge					
15	“let’s test it on board” is the design verification philosophy					
16	Engineers are busy learning enabling technologies like VHDL, Java, C++ etc.					
17	Not enough documentation on any past or current products					
18	Engineers spend more time in meetings than on their projects					
19	Your typical design cycle is “design, verify, redesign, verify, bugfix, verify…”					
	RESOURCE					
20	Idle human resources cannot be reassigned because ‘it’s not [their] job”.					
21	Sales-driven, short-term demands for product changes require additional resources					
22	Full-time employees cannot be hired fast enough (lack of qualified candidates)					
	TECHNOLOGY					
23	Multi-vendor tools create flow problems					
24	Library compatibility issues are time-consuming problems					
25	Components vendors are not brought into the design stage early enough					
26	Tool licenses are always in use or being shared					
27	Design tool flow is not verified					
28	Design modules are not reusable					
29	Different silicon foundries require different flows					
30	Newer foundry technology requires new tools					
31	Add the number of checks in each column for lines 1 through 30					
32		5	4	3	2	1
33	Multiply the contents of line 31 by the contents of line 32 and place the results in line 33					
34	Add the contents of line 33, columns A through E. This is your score:					

If your score is less than or equal to 75, you probably have nothing to worry about. If your score is greater than 75 but less than 120, you have a case for contracting out engineering. At scores of above 120, you probably need to find a Contract Engineering partner. Fast.

EXHIBIT- B

Worksheet to Determine Need for Contract Engineering

1. Gather information about past processes and costs for each product:

   a. Engineering costs from concept to pre-manufacturing over the last 3 years: C_E
   b. Cost of 'hired contractors on-site' and the justification for hiring them; such as, 'skills', 'peak-loads', 'not our core competence', 'short term' need etc. (These are powerful symptoms that reveal the inefficiency of current process.): C_H
   c. Infra-structure costs -- space, systems, tools, maintenance: C_I
   d. Delays from original planned release date to actual release date: T_D

2. Study future product plans

   a. Ask sales and marketing teams for their wish-list: what products, features, and upgrades they know customers need and the market demands (even if they believe that engineering won't do it).
   b. Present the wish-list to engineering, separately, and ask for what they could deliver, and when; and, what additional resources they would need to meet the requirements of sales and marketing.
   c. Estimate C_E , C_H, C_I & T_D based on past experience.

3. Identify your 'product' and 'process' peoples

   Compute the following ratios:

$$\frac{\text{Direct Cost of Process People}}{\text{Direct Cost of (Product + Process People)}} = K, \text{ and } \frac{\text{Number of Process People}}{\text{Number of (Product + Process People)}} = N$$

4. Calculate "notional engineering costs":

   Notional Engineering Cost: $C_{NE} = KC_E + NC_I + C_H$

5. Estimate Opportunity Costs

   Now assume that the past products were out on time, and worked well. Using T_D, use your best judgement to estimate the additional market share you would have had and consequent profits you would have made had you released your products as per original schedule.

   If the product people were free, what new or better things would they have done without wasting their time on implementation details.

   Do the same for future products, by applying a delay factor consistent with T_D.

If those additional profits, including lost opportunity costs, are more than the notional engineering costs, or almost even, you have a case for contracting out engineering.

This page intentionally left blank

COMIT SYSTEMS
The Contract Engineering Company

VER 5.0 R.10.00

Comit Systems, Inc.
3375 Scott Blvd. Suite 330
Santa Clara, CA 95054.

+1 (408) 988 2988 (Phone)
+1 (408) 988 2133 (Fax)

www.comit.com

About the author: R. Vijayaraghavan (vijay@comit.com) is the founder of Comit Systems, Inc. of Santa Clara, CA. His interests in the area of management include leadership, business models and longevity of companies.