Honeywell/
Sweatt Lectures in

Technological Leadership

(17)

Understanding the Technology Balance Sheet - A Key to Leadership

••••••

Dr. C. Gordon Bell
March 11, 1993

Monograph 21

C. Gordon Bell

Computer Industry Consultant and Director
Bell-Mason Group
Former Vice President of Research and Development
Digital Equipment Corporation

C. Gordon Bell spent 23 years at Digitial Equipment Corporation as the architect of most of its products. He is now a director of Bell-Mason Group, supplying expert systems for venture developments. He has taught at Carnegie-Mellon University and authored books and papers about computer structures and start-up companies. He was the first associate director of the National Science Foundation's Computing Directorate. He received the IEEE's Von Neumann medal and in 1991 was awarded the National Medal of Technology.

Understanding the Technology Balance Sheet - A Key to Leadership

C. Gordon Bell

March 11, 1993

Technology change is causing rapid shifts in all industries. For example, the main computer markets are moving away from high cost proprietary computers to low cost standard computers. At the same time, these shifts allow even greater opportunities for new computers and their application. Established corporations are faced with making significant changes beyond being a distributor for innovative start-ups and high volume Japanese manufacturers.

The change can only be brought about through a shift to intrapreneurism, whereby hundreds of innovative, small companies must "start-up" within every large enterprise. Companies such as Apple, DEC, HP, and IBM have established venture units, but only a few of the efforts seem to be succeeding. Intrapreneurial driven new ventures can, in principle, be formed with the same speed and reliability as one can create a major software application — provided one has the right model, tools, and a supportive culture including the right risk-reward structure. In the past, start-ups (e.g. Intel) were formed by entrepreneurs escaping the bureaucracy of the large company. It should be fundamentally easier to form an intrapreneurial venture than starting a small company, especially at a time of decreasing venture capital.

The Bell-Mason Ventures Development Model is posited in the book *High Tech Ventures: The Guide to Entrepreneurial Success*[1]. The model is used two ways: to establish new ventures, and to evaluate or diagnose the health of ventures. Both Entrepreneurial and Intrapreneurial start-up ventures use the Bell-Mason Model and Diagnostic as a guide to plan and evaluate their progress.

The Computing Industry: Another Decade of Opportunities

For the first time the industry, taken as a whole, exhibits maturity in several segments. Equipment sales in mainframes and minicomputers have leveled off, even though technology obsolescence enabled a constant replacement revenue stream. Moore's Law[2] guarantees that there will be new computer classes - the forces that created the industry, but these will be grouped within $10 to $1000 price points since a single chip is projected to store 32

[1] Bell, C.F., and McNamara, John E., High Tech Ventures: The Guide to Entrepreneurial Success, Addison-Wesley, Reading, Mass., 1991.

[2] Semiconductor density and processor speed quadruple every 3 years or 60% per year.

Mbytes in 1999.

The industry was created by the constant evolution of enabling semiconductor and magnetic technology, and waves of venture-funded entrepreneurs who created the computer classes —minis in the 70s, and PCs and Workstations in the 80s. Each new class enabled new applications software. There has never been anything like this rapid change in industrial history — the smaller the chip geometries, the more they store, and the faster they go — and the more of the past they displace and the more rapid the decline of the old — especially traditional, lethargic, incompetent companies. With this progression to an all digital world embedded in low cost silicon, one can pretty much see the transformation of the computer into Japanese consumer electronics, and into every possible information appliance (e.g. HDTV, cameras, phones, faxes, smart cards, point of sale terminals, calculator sized computers) and into smart houses and offices.

With the successful transfer of the IBM conceived and DARPA funded research that resulted in RISC chips, into Japan by MIPS and Sun Microsystems, 100% of high *information* tech hardware is now offshore - i.e. microprocessors, memories, magnetics, and CRTs to be displaced by LCDs. One has only to add the software to build systems that replace substantially all of the systems that have taken an industry 40 years to create. All new applications are predominantly desktop and deskside based, simply because of the volume and the human interface. The fact that today's $25,000 deskside is roughly equivalent to yesterday's $2.5 million mainframe in performance is the basic cause of the change.

The entrepreneurial environment is the best place to create the plethora of software for every possible generic and professional application because the first version of a new software application is best created by a small, dedicated team. U.S. software producers are not safe. With software engineering being transformed to a highly structured, factory-like process, the country with the most highly skilled, disciplined workforce, and ideas will win.

The 90s: The Decade for Intrapreneurs

The pressure is on traditional companies for radical change beyond reducing salaries and laying off people — these are window dressing to attend to a decade of mis-management, the worst of which has been poor technology planning. The strategic choices are to become:

1. Distributors for Japanese-produced components or products if one believes a high priced sales force can compete with the evolving consumer-like distribution networks.
2. Systems integrators and servicers of evolving standard equipment and your "code museums" until the relevant data is removed from the mainframe or mini by "brain sucking" software.
3. Innovative, high margin, products and services in new markets.
4. Competitive, manufacturers of computers that will increasingly look like low priced, consumer electronics like Sharp

and Sony sell, and risk reduced margins.

5. Part of another company that may have an idea of what to do.
6. Parts of each of the above.

For example, when a company decides to buy and resell someone's low tech PCs, rather than make them — was it clear to the company as a whole, what buying out meant? Did the engineers and plants responsible for designing and manufacturing PCs know their competition? Did these folks have an intrapreneurial environment to remain a competitive supplier versus ultimately being laid off? Could they have organized as a group and offered the company an alternative? Were they terminated once the outsourcing decision was made?

This is where intrapreneurism comes in. Companies must engage in radical reform where they actually create new and unique, high margin products — not just being purchasing agents and distributors for creative start-ups. Entrepreneurs and the Japanese understand this is the only way to win. The venture capital community is not a solution in 1991 because it is focused on software and biotech companies. Investors see no easy "put-a-high-tech- microprocessor on a printed circuit board and call it a high tech PC" as Compaq does. The computer classes that required minimal capital to start-up have all been formed. Printed circuit mother boards are just the packing material for shipping Japanese components, and Microsoft will make sure the software is there. Therefore, the only solution, outside of Japan, is to manufacture and compete. However, outside of a few companies here and in Japan, the will and skill to manufacture is diminishing.

Opportunities in the 1990s

The 1990s will be the most creative for the industry, not because new, very low cost computer types will emerge, but because of new applications and because computers will be embedded in more appliances. It will be clear that these are computers because they process information, not just being simulators of simple objects such as watches, and controllers. Generic capabilities like voice and handwritten input will occur and create opportunity.

One application may "help" in dealing with the paper blizzard that computers created in the first place and now exacerbated by the fax machine. People in large companies may not see this problem because most of their written communication is done through electronic mail. The bad news here is that these folks spend their time focused inward, talking to themselves. The key is how much information is known to our systems so it can be filed and eventually retrieved without human intervention? The paperless office, that has more paper than ever, is a necessity.

An intelligent communications managing assistant for phone, fax, data, and videophone as part of our computers could enhance communications and free much time. This non-trivial assistant could provide more productivity in the increasing number of secretary-less offices.

Finally, digital video embedded into the computer has been introduced. Videophone in a window, videomail, and videofax will be generic capabilities for the PC by 1995.

High Tech Ventures describes many application possibilities in the technical area, but the commercial or generic ones are just as exciting. The key to success is establishing a relationship between the problem faced by the computer user and the engineers who can solve it.

Where Do Products Come From?

The classic business schools market process steps show why the Japanese and start-ups win in product creation: focus on the customer *through a marketing organization*, make a business decision *with no real knowledge since a product area doesn't yet exist*, and then set out to have a team, or *usually a committee*, design and develop a product. Engineers looking at all the new technology and problems, are sickened by this approach to product creation. None of us knew we needed Walkman or Watchman until Sony made and tried to sell them.

High Tech Ventures describes how new ventures can form to exploit technology, provided they have an entrepreneurial or intrapreneurial environment. In many cases, start-ups come from people and ideas within existing companies simply because the large company cannot deal with a new product, new market, or intrapreneurs. *High Tech Ventures* describes the push and pull effects that create the start-ups; large companies must change to support novel products in new markets.

Customer Involvement in the Creation

Notice that few of the above products have come from the customers, even though users play a key role in first product definition and testing. A company such as Cullinet formed based on software it obtained from a customer. Finis Conner describes his process as: "sell, design and build". This requires engineers to actually visit potential customers.

When the product developer can act as customer, it is easy to design great products. Timesharing, UNIX, C, and many of the programming languages happened this way. When product developers have little first hand understanding of the products they create, having direct links with the customer and problems being solved is crucial. In fact, without these links throughout the development process, it is impossible to design an effective product. In the 1990s, applications for specific professions necessitates having professionals who are expert in a user's problems.

How Intrapreneurial Ventures Are Formed within a Company

Given that the industry has a need for new ventures in order to survive and the above sources of ideas, the remaining ingredient is to support their formation. *High Tech Ventures* describes the critical organizational features and process for starting up new companies. Starting up within a corporation is similar because all ventures are constrained by: technology, markets, products, and people who can create and manage the venture. What differs is: start-ups have few rules, few resources, and less technology while large companies have too much of everything — especially people who want to control every aspect of a new venture. The key to a start-up's success is having fewer, but extremely competent people. Based on our experience, a well-run, dedicated start-up has an order of magnitude fewer people and delivers a product in roughly one-half the time of the large company! In addition, large companies are usually unable to nurture the creation of innovative products.

While the Bell-Mason Diagnostic described below was designed for start-ups, it applies virtually unchanged when used within a large company. A few changes are obvious: the CEO has some other title such as business unit manager; the team and venture's culture has to be focused like a start-ups; there is no board with fiduciary responsibility, but a board representing all the aspects important to the venture must exist; and most importantly, the issue of financing expectations must be clearly defined. Furthermore, the integration with the existing organizational entities (e.g. engineering, sales) must be clear and supportive.

The Bell-Mason Venture Development Model and Diagnostic

The Bell-Mason Diagnostic is a rule-based tool designed to characterize the status of a high-information-technology venture, at each stage of its growth. The Diagnostic posits a model for how every high tech venture starts up and grows. A venture is evaluated at a particular stage of its growth, by answering a series of questions that contain the rules (our expertise) to measure each of twelve characterization dimensions. The answers are tallied and plotted on a 12 dimensional relational or Kiviat graph, which is then compared with the "ideal" for that stage of development.

The four elements of the Bell-Mason Diagnostic are:

1. Segmentation in Time: The five stages of company growth
2. Segmentation to reduce and structure complexity: The twelve assessment dimensions
3. Evaluation of the dimensions: The rules used to evaluate each dimension
4. Visualizing the situation: A 12 dimensional relational or Kiviat graph is plotted against the "ideal" model for success

Element 1: The Five Stages of Company Growth

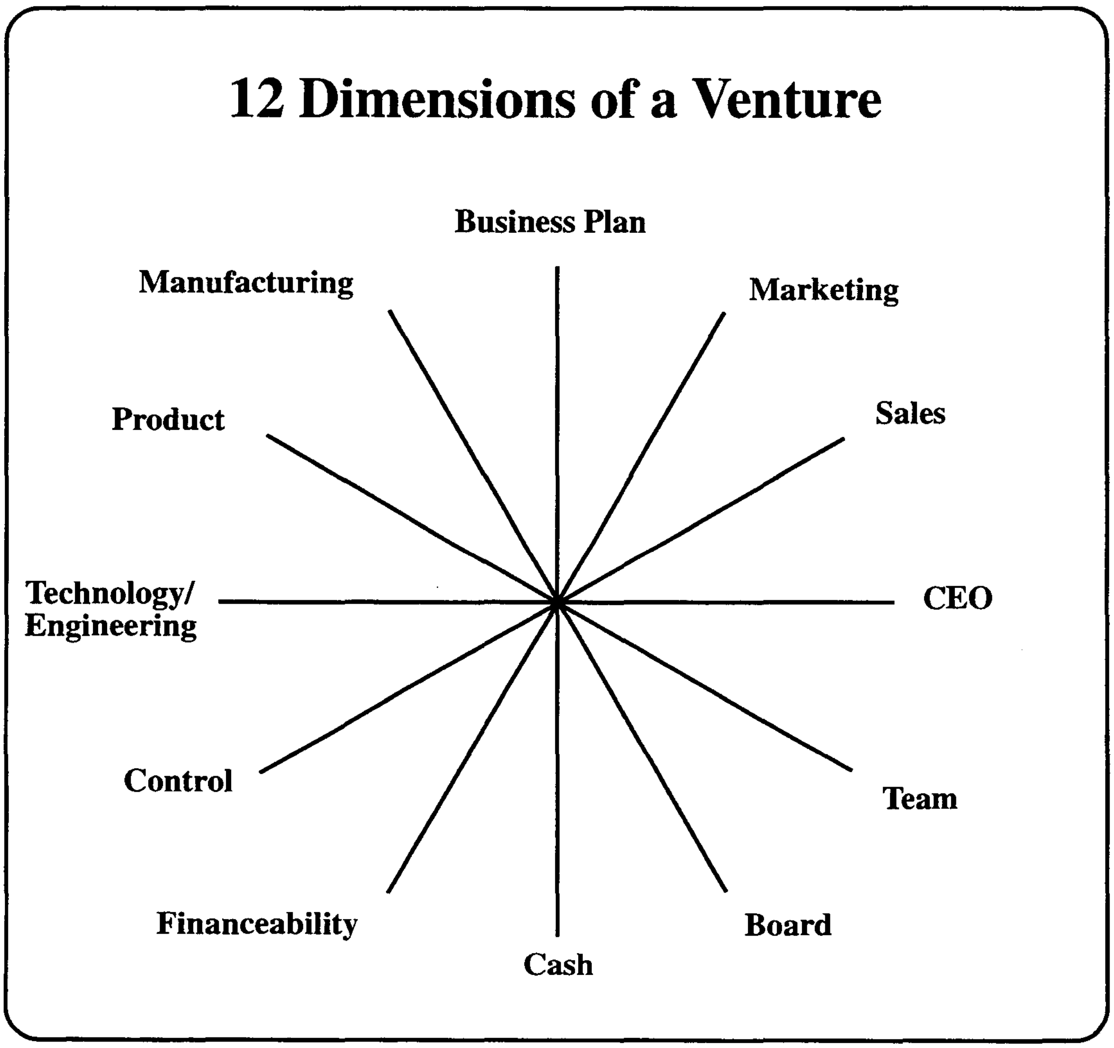
The range of computer- and communications-based companies is large. Hardware components start-ups produce and sell such items as integrated circuits and disks. Software components start-ups serve all computing power levels and deal in hundreds of software segments. Complete computer systems manufacturers may create anything from voice-controlled, credit card-sized PCs to supercomputers. End-user applications software start-ups bring us games, inventory control, word processing, and mechanical design. Distribution, service, customization, training, and operations also constitute a major segment.

Despite the variety of start-ups, all healthy companies starting up in the information-technology field pass through the following four, predictable, measurable, sequential growth stages before they reach steady-state.

The stages and their typical durations are:
Stage I: Concept (0-? months)
Stage II: Seed (3-12 months)
Stage III: Product development (6-36 months)
Stage IV: Market development (2-4 years)
Stage V: Steady-state

The concept stage is the company's starting point and can be initiated from any viewpoint—such as market, technology, or product—but it requires the drive of an initial group who have been infected with entrepreneurial fever. The seed stage is where the company proves technology/product efficacy. The product development stage has four substages, which correspond to four product development

Figure 1



phases: Hiring and planning, Designing and building, Alpha testing, and Beta testing. The market development stage, marketed by the product introduction has three substages: Market calibration, Market expansion, and Steady-state operation. While "cashing out" is a declaration of having entered the steady-state stage, a healthy company may choose to remain private and profitable, thereby entering steady state without the fanfare of an IPO.

Element 2: The Twelve Dimensions of a Venture

Innumerable factors, large and small, indigenous and exogenous, influence the course of a start-up. These can be distilled and categorized into only twelve key elements, or dimensions (shown in Figure 1), which determine the ultimate fate in the marketplace. By using the Diagnostic's rules to evaluate the strength of each of these dimensions at each stage of growth, the start-up's health can be assessed and its future outcome predicted and managed. In effect, the venture is compared against an "ideal." Of course, the very process of conducting the assessment (i.e., identifying and carefully scrutinizing critical issues) is likely to have a significant positive impact on the venture's outcome.

Many catchy formulas for how to conduct a successful start-up have been proposed that usually focus on a single dimension. One of the earliest venture capitalists, Arthur Rock, characterizes the entrepreneur's traits as follows: "a burning desire to start a company. A person has to be very, very honest . . . recognize problems, foresee problems, recognize shortcomings, and admit and learn from mistakes." He also reduces the whole issue to "People, people, people," while still others advocate a more balanced, but still simplistic, maxim: "People, product, plan." Bob Keeley's studies at Stanford indicate that without a very good first product, the company is likely to fail because it will run out of time.

Whenever stories of business success or failure are told, they almost invariably cite a simplistic formula like those above as the moral of the tale. The modern intrapreneur or entrepreneur must avoid such maxims, no matter how clever they are or how reliable their source or how true they may once have been, since none of them even begins to capture the challenge of the contemporary new venture.

Element 3: The Rules to Evaluate Each Dimension

Each of the twelve dimensions is evaluated at each of the four stages of growth by comparing the start-up with what we define to be an "ideal" for that stage of growth. This comparison is performed by having key participants in the start-up answer a series of questions, which, in effect, constitute a checklist. The questions themselves are the rules that define the "ideal." Thus, the company is on track across all dimensions if it answers "yes" to all the questions. The entire diagnostic consists of over 700 rules and a detailed stage or sub-stage may embody 70 questions. Diagnosing a company or venture requires roughly 1/2 day with two people and the venture's management team.

The "Laws of good practice" come from our observations, with review and testing. These observations result in "heuristics," used herein to define our "ideal" start-up in the form of questions. Note how the product development rules (questions) evolve and become more stringent as the company progresses through the stages as shown in Figure 1.

Element 4: Visualization —The Relational Graph

Figure 1 shows each of the twelve dimensions as a spoke in a polar graph, with the spokes separated by 30 degrees. Plotting the scores for the answers to the twelve sets of questions produces the "value" for each dimension. The dimensions grow in value from the center of the circle to its circumference as the company progresses through its stages of growth. The figure shows three of the four elements of the Diagnostic: the stages of growth, the dimensions that are measured, and the "ideal" model for success at each of the stages. This enables the user to see at a glance how the company's current status compares with the ideal values for all of the dimensions at a particular stage.

Example Concept Stage Questions for Diagnosing a New Venture

The CEO (or head of the venture): Does the chief executive or (head of the venture) possess the level of intelligence, energy, ethics and quality that is required to establish the clock and culture for the proposed company?

The Team and Culture: Is there evidence that the founders can function as a team? Have they worked together for three to six months? Do they respect one another? For intrapreneurs: Is the venture's culture solely focused on winning in the marketplace?

Financeability: Has the start-up gained the respect of at least three reputable outside persons whose backing lends credence to the technology, product, market, and company concept? For intrapreneurs: Are the returns and business plan compatible with the corporation?

Technology: Can the team show how the technology can be developed while requiring less than three breakthroughs in the state of the art?

Marketing: Have a set of customers been identified for the product? Has a simple estimate of the the market size been developed, supported by article and extrapolated market numbers?

Example: Ovation - The Case of the Missing Product

Ovation was a software company founded in 1982, that declared Chapter 11 bankruptcy about two years later. The company spent about $7 million without producing a product. Ovation was created to build "the next generation of integrated microcomputer software". The Ovation product was defined as a package that seamlessly integrated word processing, spreadsheet, database management, and communications functions. The market for the Ovation product was cited as Fortune 1000 companies at a price of $695, commensurate with Lotus' 1-2-3. Healthy funding was secured from choice venture capital firms. What happened to cause the company to stall and ultimately fail in the product development stage (Stage III)?

While the company was perceived to be at the end of Stage III, the product had just reached the seed stage, as indicated by its definition; but no technology or a seasoned development team was in place to build a product of inherent and protectable value. The lack of product caused analysts to create the name "vaporware".

The business plan was clearly not a dynamic control document. The company was stuck at seed, and reflected the realities of product schedule slips and re-scaled forecasts. The marketing and sales departments, however, were completely staffed and running at full tilt, producing all the sales collateral, demos, sales materials, and communications programs. The group spent at a high rate, in hopes of generating indeterminate future revenues on a product that didn't (and ultimately never would) exist. Sales was substantially over staffed for the product development stage. The marketing and sales approach was impressive, and should have proven effective if the product could have been built. For example, the company had already won an award as the most important and innovative new product.

The CEO gets an alarm rating for his inability to re-balance the company, once it went off track. The Team was clearly not operating as a coordinated and unified group, with complementary functions making steady progress. The Board, in its capacity as primary reviewer and counselor to the company, also failed to identify and rectify the problems before they were fatal errors . This last point is not surprising, since the board had virtually no outsiders, and was made up entirely of marketing and sales staff.

The company's substantial cash, more than what a normal start-up might obtain, was aggressively spent at a time when the chance for future financing was diminishing. The company's poor track record for achieving any externally-measurable result like alpha testing, beta testing, bookings, or sales, negatively impacted further funding. Control and operations were primitive in relationship to where the company was supposed to be in product development, hence the company was, and had been, constantly out of control.

Even with 20-20 hindsight, it is not certain that early intervention would have saved the company or made it viable. However, careful monitoring and constant use of the Diagnostic would have served as an 'Early Warning System', almost at Ovation's start, against the expensive eventuality of failure of what was simply a case of not having a product.

Radical change is required by companies this decade in order to lower costs, decrease time to market, and create new products. Given the need for such a radical change, the only solution was to become creative through intrapreneurial units. By using the right tools a large organization should be able to permit the establishment of creative intrapreneurial units as reliably as the creation of a new software product.

The Technology Balance Sheet

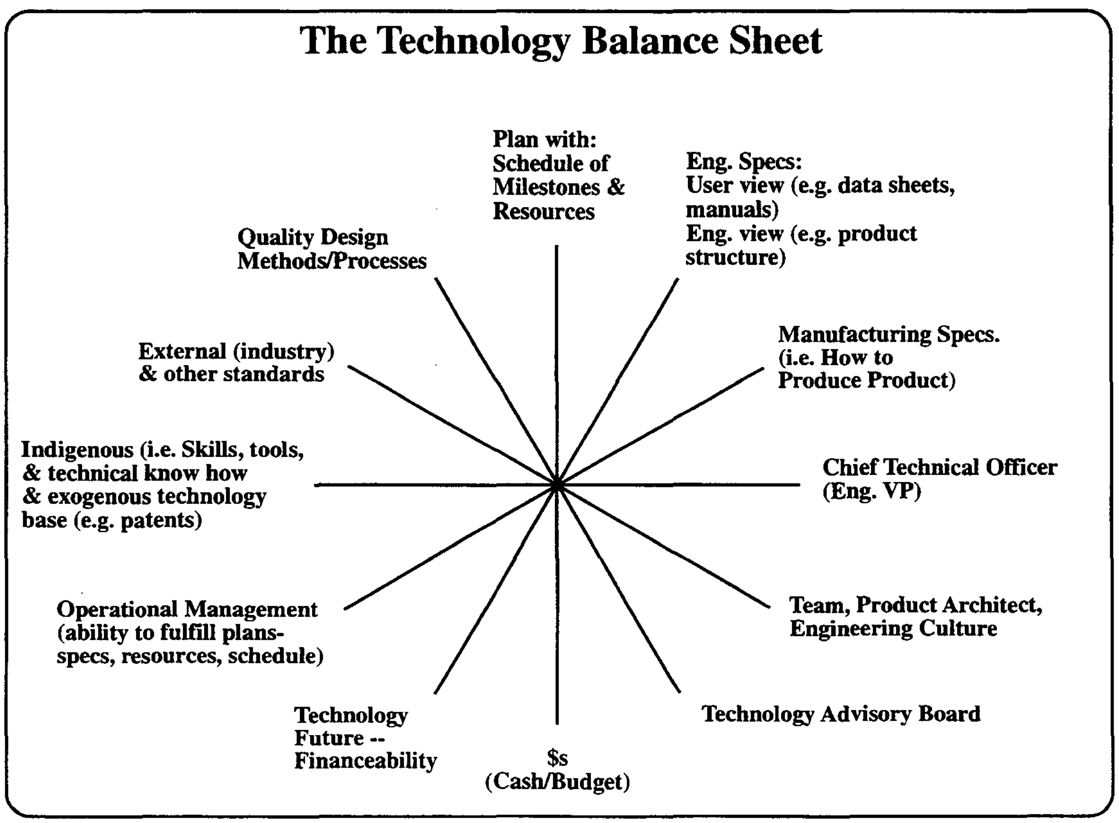
Just as it is essential to understand an organization's financial health, it is equally necessary to understand and measure its technological health. The technology balance sheet measures the company's technology. A second section presents classic flaws in technology which range from requiring infinite technology, because a fundamental discovery is needed, to having no fundamental technology. The final section describes the rules for evaluating the company position at the concept and seed stages.

The technology balance sheet evaluates each of twelve dimensions of a start-up's technology. Readers may notice that the dimensions used on the technology balance sheet to assess a firm's technology are very similar to the dimensions used to assess a firm's overall status. Figure 2 shows the twelve dimensions to be considered and measured.

Technology

The technology dimension includes internal and external sources of components, plus "know-how," as represented by critical personnel, patents, processes, etc. The company must examine every facet of the technology that it needs to build a product

Figure 2



and then rank each technology source as objectively as possible.

Standards

Standards should be regarded as a critical aspect of product design. Establishing uniform ways of doing things (such as having an exact dialect of a language for expressing a program and having programming style conventions) permits rapid progress because standardized components can be interconnected and built on one another. Although standards are inherently constraining, Dave Nelson (one of Apollo's founders) believes that constraints are what really breed creativity. In designing a product, it is inherently more difficult to start with a blank slate than to start with some constraints, because in the absence of established criteria, nearly anything is possible. Effort will therefore be squandered exploring an almost infinite number of options rather than channeled and focused in the most productive directions.

A start-up (or a company of any size, for that matter) must understand and implement both external and internal standards. Major aspects of product design are determined by external (industry) standards covering such areas as inputs, outputs, cost (in memory size), and speed. For example, a compiler may be specified as having to accept ANSI (American National Standards Institute) standard C language input, produce code for the Motorola 88000 chipset that is better than the existing compilers, occupy no more than 100 kilobytes of memory, and compile at over fifty thousand lines of code per second.

Internal standards are equally important and range from how logic design or programming is done to line width on printed circuit boards. Internal standards must be specified, published, and enforced in a formal manner. For instance, when Digital Equipment Corporation (DEC) first started, the engineering committee took responsibility for creating a set of design standards that covered everything from how a physical environment would be specified and tested (power, temperature, humidity, etc.) to how a copyright statement would be placed in memories and programs. Internal standards also include a list of the components that are permissible in new designs.

Upon seeing such standards and component lists, the first reaction of most engineers is that they are bureaucratic and constrain creativity. However, standards are simply a statement of decisions that have been made regarding good practice, which means the designer doesn't have to think about these more mundane aspects of a design (such as the temperature at which the product should be designed to operate) and is therefore free to concentrate on the truly creative aspects.

Design Process

The design process, which specifies what tools engineers use to create and check each part of their product design, must be documented and managed. The design process is intimately tied to the resources a company has to aid designers. Much has been written about software engineering, and there are any number of valid models for how programming should be done.

The important thing is simply to pick a model that is appropriate for the team and operate according to it.

The Software Engineering Institute has established a five-level ranking scheme, derived from Deming and Juran, to characterize how effectively a team is functioning in terms of its process capability:

1. *Initial:* There is an ad hoc process. Formal procedures may exist, but there is no management mechanism for tracking results against the procedures. The team rarely makes and meets plans.
2. *Repeatable:* A process exists that deals effectively with routine programs but produces unpredictable results with new programs or new tools.
3. *Defined:* A qualitative description of the process exists.
4. *Managed:* The process contains a minimum set of measurements to define quality and cost; a process baseline exists; etc.
5. *Optimizing:* There exist sufficient quantitative measures for each part of the process to allow the process to be completely understood and fine-tuned.

Humphrey describes a method for evaluating a company's process capabilities and also recommends various processes, standards, and methods for attaining software process control. The organization with which he is associated, the Software Engineering Institute, can audit a firm to determine its level of process control, and some members of the institute did so as part of a 1990 trip to Japan sponsored by Ministry of International Trade and Industry. While there, they found that many large Japanese companies are operating at the highest level in the above ranking (level 5, optimizing), enabling them to achieve quality and productivity levels more than twice that of their U.S. counterparts.

Engineering Plan

The engineering plan includes the schedule and a list of the resources required. The resources list must cover both the resources for developing the product itself and those for developing any of the manufacturing and design processes that the product requires. The important thing about an engineering plan is that the schedule be realistic. Developing a truly realistic schedule is almost impossible if the product has never before been attempted; it is merely very difficult if the product has been attempted previously but the team has never before worked together. In the latter case, each team member's ability to establish a realistic schedule for his or her portion of the work will probably be untested. In terms of the process-capability levels outlined by Humphrey, it is unlikely that such a software team in a start-up could get above level 2 (repeatable) by the time it ships its first product.

Product-gestation time gets ingrained in the people and the company. Their ideas regarding product-gestation time are often based directly on the lead times at a larger company, which are guaranteed to be much, much longer. One of the most important aspects of an engineering culture is to establish an accurate but responsive ability to schedule. There are four ways to schedule a project:

- *Optimistically:* Put enormous pressure on the team by preparing an aggressive schedule that the team believes can only be met if everything goes right.
- *Pessimistically:* Build so many delays and contingencies into the schedule that the schedule will certainly be met (an approach unlikely to be used by a start-up).
- *Realistically:* Allow for an appropriate number of contingencies, which will become possible when the team is mature enough and understands the project and each other well enough. However, it often happens that everyone up the chain of command then adds a contingency, and the net result is a bloated, pessimistic schedule (again, not typical of start-ups). With realistic scheduling, the company may end up with two schedules—the optimistic one and the one with the contingencies added.
- *Running blind:* Work on the project until it gets done. The company that uses this approach had better start with lots of money, be able to raise more money easily, and have plenty of extra resources.

In the final analysis, schedules really don't always work. Any critical schedule milestone must coincide with an immovable deadline such as a demonstration to the board, a trade show, a funding event, or a customer shipment. If customer shipment serves as a deadline, quality must always be used to control shipment.

Engineering and Manufacturing Specifications

The engineering and manufacturing specifications describe the product in several ways. First, they describe its external specification, or the product's function, including performance, as seen by a user. Second, they describe its internal specification, or the product's structure and internal function as seen by the engineering team (a set of components to be designed). Finally, when the product has been fully specified both externally and internally, manufacturing requires process and product specifications describing how the product will actually be built and tested.

Chief Technical Officer

The chief technical officer (CTO), or engineering vice president, is the technical leader in charge of implementing the product. This person is ultimately responsible for all products and is the CEO for the engineering organization. Thus, his or her general qualifications must parallel those of the CEO because the CTO is the "clock" and "standards setter" for engineering.

The company should have selected its CTO by the end of the seed stage, and if it is tackling a technologically difficult product, the CTO must be on board from the start. Funding a high-tech start-up without a CTO is extremely risky because he or she is the individual responsible for ensuring that the product is really feasible at the price, quality, schedule, and resource level specified in the business plan.

Engineering Team and Culture

The engineering team and culture are just beginning to form by the end of the seed stage, since at this point, a complete team has yet to be hired and the head of engineering may not even be on board. The organizational structure is quite important because the CTO may have positioned himself or herself as a bottleneck by assuming responsibility for all intergroup problem resolution. As with any organization, theory X, Y, and Z will all work. I do not favor top-down engineering organizations because they do not bring out the creativity of the people doing the work. Furthermore, top-down structures eliminate critical intraorganization communication. Worst of all, top-down organizations usually do not engender commitment to schedule, resources, and product on the part of the responsible engineers.

Architecture

The term *architecture* was coined in 1964 by the IBM System/360 design team to describe the instruction set of a computer, or how the computer appeared to a program (or programmer). *Architecture* is now used in a broader sense that encompasses both "external architecture" and "internal architecture." The external architecture describes the general function of any computer component (i.e., what it does)—such as the instruction-set architecture, operating system, compiler, a network protocol, or spreadsheet—and how this component appears to anyone using it. The internal architecture (or "realization") forms the blueprint for how the components that create the external architecture are implemented; it is what a development team designs and builds.

It is therefore vital to have a product architect who can both define the product externally and be able to play a major role in decomposing it for realization and then engineering it. His or her responsibility for product architecture applies equally to all levels of hardware and software. Thus, the product architect is likely to be the most critical person in the engineering group, including the CTO.

The architect's key job is to guide the product's implementation and evolution over its lifetime. The lifetime of a good architecture will be considerable, and the company fortunate enough to have chosen a good architect and architecture will profit immeasurably. Much of DEC's success during its first three decades (1957-1987) was based on constant and evolving architectures for its minis, including the VAX. System/360 hardware and software systems and their successors were the basis of most of IBM's revenues and profits for a similar period. More recently, the Apple II and Macintosh architectures have each prospered for over a decade. In 1990, Sun Microsystems has been attempting to repeat the success of these predecessor architectures by establishing the SPARC architecture as the standard for workstation-class computers.

Although most of the examples cited above involve hardware engineering, the same sort of architectural integrity must also be maintained for software. At Microsoft, every product, such as Word or Excel, has a single architect who maintains the product's integrity (and is usually its chief implementer as well). When responsibility for a product is diffused, as in the case of Fortran or UNIX, by placing it in the hands of some amorphous, committee-like group that is pushed around by numerous standards organizations, the product's efficacy declines and its ability to evolve may be stymied.

In my view, lack of a good architect, or lack of a suitable architecture, is the fatal flaw in many companies. Although at first, the product architect may be the CTO or even the CEO, ultimately, someone within the engineering organization must assume responsibility for maintaining the efficacy of the product's external specification, especially with respect to how that product is changed as it is implemented in succeeding product generations. In some cases, as in the Ardent Titan example described later, several architects may be required as a product is broken into various parts.

Not having an architect is quite risky, because it leaves the product's definition to some nebulous process or to a group that gropes with the product design, as I recently saw in a company building a multimedia system. Not having a way to manage the product's design and delegate it to those who must do the design is almost always fatal. One of the biggest dangers is overcommitment. When a technically difficult project begins, and one person functions as CEO, CTO, and architect, the company, engineering group, and product are all likely to be out of control unless the firm has a sufficiently strong staff, including a chief operating officer. In a start-up, such a project must have a full-time architect who will also play a major role in the product's design.

Technical Resources

The next dimension on the technology balance sheet is technical resources. This essential category includes people, equipment (both computers and networks), and software to run the engineering enterprise (i.e., operating systems, languages, computer-aided design [CAD] programs, and software licenses). Of all the resources, the technical staff is the most important.

The company's hiring ability determines the quality of the staff. All firms, regardless of how well they may be managed, find that hiring grade-A people takes much longer than anyone had planned. The key to hiring is having the right sources. The most effective approach is to develop a network of contacts with the best people working in each area such that recruiting is by word of mouth. A technical advisory board can be one of the company's greatest hiring assets.

The organization's first hires have to be great because really good engineers like to be involved with other good (or even better) engineers and are intolerant of "bozos" and "turkeys". Great people hire even better people. Poor people hire even poorer people. (This is the pygmy theory of hiring.) Furthermore, because the company can expect to acquire its share of average people merely in the course of making minor hiring errors, it must never deliberately hire average people just to fill slots. Visix's hiring process is highly recommended.

Visix Software: Visix built a high-quality, high-performance platform for building graphical-interface, networked applications. Its desktop for UNIX, Looking Glass, extends that of the Apple Macintosh to handle networking. Visix achieved product quality by implementing a rigorous hiring process, by managing to keep a small team together over a five-year implementation period, and by having a single product architect. A key step in the hiring process was to review the code of each potential software engineer. Any engineer reluctant to show his or her work to fellow engineers is a likely loser.

Technology Future

The technology future dimension measures the company's ability to sustain the competitive viability of its technology. This dimension includes such factors as an assessment of the firm's products and architectures relative to the state of the art, morale, process technologies under development, and ability to hire critical people. Like financeability, the technology future dimension represents an overall look at the start-up's ability to build competitive products in the future.

For example, assume Company N introduces a Motorola 68000 workstation based on a CISC (complex instruction set computer) microprocessor, perhaps with an attached signal processor, while all the other workstation companies are introducing products based on RISC (reduced instruction set computer) microprocessors (such as the Sun Microsystem SPARC, MIPS, Motorola 88000, or Intel 80860). Because the RISC microprocessors deliver higher performance, Company N's product specifications suffer by comparison, at least superficially. Company N's ability to respond to the ensuing performance race by increasing the workstation's functionality with voice and video for example, and a wide range of applications software in the 1990s will determine its technology future.

Operational Management

Operational management is the engineering organization's ability to manage itself by meeting its product specification, budget, and schedule commitments. Management includes all the techniques of managing design reviews, management by objectives, staff meetings, team building, conflict resolution, etc. As a product reaches the final stages of completion, it will become clear that the team must compromise among the following three indigenous variables:

- The schedule, or when the product will be ready
- The complete set of resources that is applied toward meeting the schedule, including computers, consultants, other software, etc.
- The characteristics of the product itself, including performance, product cost, features, etc.

The best approach for the company is to pick two out of three, manage those, and be happy with the outcome. For a start-up, the schedule and resources are really fixed because of the incredible cost of raising additional funds. Furthermore, it is generally inadvisable to attempt to add critical design resources to a project that is already running late because the firm is then apt to become subject to Brook's law: "Adding resources to a late project makes it later."

Therefore, the function of the company's first product will inevitably be less than perfect. Faced with the need to cut function in order to meet schedule and resource constraints, it is best to sacrifice some of the product's features rather than sacrifice performance. Performance equates to quality in many systems and should not be sacrificed. Likewise, reliability is not a "feature"; it is a quality constraint that must never be sacrificed.

Tackling a Product That Requires Significant Research

A wonderful product that is clearly needed is just waiting to be developed. Designing the product, however, will require an unknown amount of basic and applied research. As of 1990, the estimate of when such a product can be produced ranges from now to eighteen months from now to never (although *never* is a word that cannot really be used when it comes to technology). The following example illustrates the slow evolution of a product whose development has required (and will continue to require) a considerable amount of research.

Requiring a Trilogy of Breakthroughs

It has been observed that a successful start-up cannot be based on more than two breakthroughs in the state of the art. Each of the areas requiring a breakthrough should have alternative technology as a backup. Clearly, a risk exists when three or more technologies have to be understood (i.e., researched to the point of being usable) and developed. It is almost assuredly fatal for a start-up to engage in research whose result cannot be known or scheduled, because the company's other functions must all be supported in the meantime, and the funding requirements are uncertain and often open-ended. The schedule for such a project contains loops, parallel and redundant exploratory paths, and conditional branches.

The following example discusses Trilogy, Inc., which attempted to develop a product requiring multiple technological breakthroughs. The "trilogy of breakthroughs" flaw is in fact named after Trilogy, since this flaw contributed greatly to the difficulties the firm encountered.

Trilogy, Inc.: Trilogy was started to develop an IBM-compatible line of computers with major subsystems packaged on a single semiconductor wafer. Unisys and Digital[3] invested in the technology as codevelopers. The risks included the following:

1. Interconnecting high-density, high-speed semiconductor circuitry on a single wafer.
2. Devising a scheme to ensure defect-free parts using redundant parts of a wafer.
3. Packaging an entire wafer such that power is input, heat is dissipated, and

[3] I made this recommendation. After Trilogy failed, Digital bought rights to all its technology. The power supply, heat sink, wafer-packaging scheme, and facilities were used as the basis for the VAX 9000.

the wafer is rewired to circumvent inherent wafer defects.
4. Developing a CAD system to manage the redundancy-based logic design and interconnect scheme.
5. Developing a computer design more complex than previous designs.

Some observers felt that Trilogy's pleasant facilities and large staff were fatal flaws. The real culprit, however, was that the requisite technology could not be developed in time to implement a product. The five risks listed above had the following outcomes:

1. The circuits were slower than specified, increasing the design's complexity while decreasing its competitiveness.
2. Not enough redundancy was available to cover wafer faults.
3. The CAD system was quite slow and decreased productivity.
4. The design was so complex as to increase the design time and adversely affect product competitiveness.

Although the preceding problems occurred during the product development stage, the issues were known at the concept or seed stage. In hindsight, an analysis of the situation should have produced an emphatic "no go" until the required breakthroughs were reduced to a manageable number.

When it became clear that Trilogy's technology was inadequate to build the product, the company acquired Elexsi Computer with its remaining capital and attempted to make it succeed. Unfortunately, minisupers from Alliant and Convex were also being brought to market at that time.

Having Little or No Sustaining Technology

Offering just another commodity product of a particular type (i.e., "brand X") in a crowded field is usually a fatal flaw. Starting a company with commodity technology, such as a new chip, is the opposite of the trilogy-of-breakthroughs flaw. It comes from the belief that the firm has just a slightly better idea about the product or how to sell it. The minicomputer, PC, and workstation industries all began as technology companies to a greater or lesser degree, and the introduction of various components (SSI/MSI, 16-bit microprocessors, and 32-bit microprocessors, respectively) allowed dozens of no-tech companies to enter the market. In early 1990, the smallest PC electronics assembly costs $200, and within five years, just one or two very-high-tech chips (available from Intel and a memory supplier) will form the entire, minimal PC with 2 megabytes to 8 megabytes of memory. Dell Computer is an excellent example of how a company was able to get started and grow with PCs despite the low-tech odds, because Dell considered the whole environment of product, sales, service, and support.

The Not-Invented-Here (NIH) Syndrome

One of the most dangerous flaws is a form of technical arrogance in which a company feels compelled to reengineer every part of a hardware or software system because it believes that it can do a better job than any of its potential suppliers. For a start-up, inventing every possible component in order to make an ultimate product (instead of buying everything possible in order to get to market rapidly with a good product at the lowest development cost) is often fatal.

The other effect of the NIH syndrome is the incompatible-product flaw. A company designs a new interface, such as a programming language or a feature for an existing language, when an old one would have been just fine. In this case, NIH hurts the buyer, who has to change and adapt to something different. Needless innovations and changes that have the effect of rendering hardware, programs, and data incompatible are extremely costly for the whole computing enterprise.

The NIH syndrome is endemic among most engineers, especially in the United States, the United Kingdom, and France. NIH does not necessarily have anything to do with a team's competence, only its lack of business savvy, although the brightest teams are often the most unhappy about using less-than-perfect components. The NIH syndrome's effects on productivity and on profit and loss are devastating, and this syndrome may account for the reason that Japanese engineers are at least twice as productive as American engineers in a field such as automotive engineering. NIH often triggers the formation of multiple companies in one, a type of business plan flaw.

Even well-established and well-respected firms have exhibited this flaw. In the early 1960s, IBM found that every computer products group was building a computer based on each group's own logic circuits, requiring redundancy in design, manufacturing, and field spares. Gene Amdahl proposed that any group using components from another group be rewarded and given special recognition. One of his coworkers squelched the idea, claiming that "it's un-American."

The Missing Component

Every day that an organization depends on a risky part or a marginal vendor, it risks its life, because if a critical component (or process) fails to materialize as scheduled, the company may run out of time and, hence, out of money. Selecting poor vendors is a common and hard-to-avoid error. Only through experience will a start-up learn which firms can be trusted to meet their commitments.

Inability to Hire the Engineers

Hiring is absolutely critical, yet every start-up I know of has had more trouble hiring than it ever planned or imagined. This leads to an additional flaw—lowering the standards. By reducing its standards, the firm risks producing both a downward spiral in quality and a bloated staff that generates no meaningful output. A pygmy heading engineering will proceed to hire even smaller pygmies.

Failing to Get Rid of Poor Hires as Soon as Possible

If a person is found to be a poor hire, he or she must be dismissed at the earliest feasible moment. Negative producers[4] should be terminated immediately, placeholders very rapidly, and marginal producers as soon as possible.

A firm, "X", was having trouble staffing a new project with good people and made a borderline hire without proper reference checking. When the team discovered that the borderline individual was in fact a poor hire, they felt they could manage him by close supervision and checking. However, he refused to ask for help, chafed at having his work reviewed, and was late—all sure signs of a bad design(er). Simulation revealed continued bugs with no evidence of progress toward a correct design. In essence, bugs were just being moved around. When Company X finally conducted a design walk-through, the engineer quit and went to a competitor, where he may or may not have greater success. Although Company X did nothing to influence its former engineer's selection of a new employer, outplacing negatively productive people with a potential competitor can do wonders for a firm's lead.

[4] The principle of negative productivity is a theory I claim is worthy of a Nobel Prize. Normal principles of productivity assume that workers create positive output. Brooks refined the concept of software productivity to express it in terms of the "mythical man-month," and in software engineering, it is understood that different programmers vary in their productivity by several orders of magnitude. According to the principle of negative productivity, it is possible for an individual to produce bad results that others must then redo; hence, someone who is very negatively productive can keep a whole team busy with damage control, preventing the team from producing any output whatsoever.

Leaking Technology and Product Ideas

If a company permits its technology and product ideas to leak, it risks giving both established competitors and other start-ups an opportunity to respond. It is therefore important that the staff say no more than is absolutely necessary in order to sell new recruits. They should try to get recruits to tell more about themselves than the company tells about itself and avoid any mention of costs and schedules.

Preannouncing the Product

It is absolutely foolhardy to preannounce a product before it has been tested internally and passed its acceptance tests. At the very least, preannouncement is likely to be an embarrassment; at worst, there might be legal repercussions. In no case should a product be officially announced before it is operating well enough to pass formal tests that are comparable to actual customer use. Ideally, the product announcement is made at the end of beta testing at customer sites.

This is one flaw that is even more painful in large companies than in start-ups. In 1966, IBM preannounced a large computer that would compete with Control Data Corporation's 6600 in an attempt to get customers to wait for the IBM product, which, in this particular case, would never come. CDC sued IBM and was awarded $600 million in a consent decree that forbade preannouncement.

Conclusions

Semiconductor and magnetic densities, as measured by the number of bits stored per unit area, are likely to increase at their current exponential rates on into the 21st century. These increases will provide opportunities for new hardware systems, which in turn will allow new software products. Technology opportunity comes in many forms: components such as semiconductors, standards, customer application needs, and genuine inventions. Technology, i.e. the ability to design and build high technology products, comes from having trained resources that concentrate on discovery. A significant portion of the world's research and development is available to entrepreneurs and intrapreneurs in various forms ranging from papers, demonstrations, and consortia, to trained people. Technology transfer is best accomplished by the transfer of people, as occurs when people leave a laboratory where they have developed an idea and form a separate group or a new company to commercialize the idea with a product.

Having technology know how is a necessary prerequisite for a start-up, but not sufficient. In addition, it must have engineering mastery to enable the successful conversion of technology to products in a predictable and timely fashion. The technology balance sheet describes the evaluation of a company's technology and engineering abilities, and maps these into twelve separate aspects or dimensions.