

Managing High-Tech Teams

By John V. Levy, PhD, CMC

Version 0.81

Copyright © 2002 John Levy Consulting LLC

info@johnlevyconsulting.com

415 663-1818

Introduction

Managing a High-Tech Team successfully requires knowing what the team needs. Understanding the teams' viewpoint makes it easy to fulfill the needs. This article describes some of the basic needs and desires of typical high-tech teams, and attempts to put those needs in context. Examples of what happens when the needs are met and when they are not met clarify the nature of the needs.

This is not a technology story. It's a story about technologists and their managers, people interacting with people.

Many high-tech firms didn't get anywhere because their projects didn't complete on time or their products didn't work right. Very few companies fail due to shortcomings in scientific and technological capabilities. The cause of failure is often interaction problems between the people doing development and the people managing them.

Experience

These needs are based on observations I've made while working in the computer industry, both as an employee and as a consultant. My colleagues who love to design computer hardware and software spent many hours complaining about the people who managed them. And my clients, who were managing companies or departments with design engineers in them, frequently asked for help in understanding how to keep the engineers productive and happy. Sometimes they only asked how to get the project done or the product delivered. The underlying issues were always related to

interactions between the first and second level managers and the engineering teams.

Not every engineer needs exactly the same thing as the next person. But the needs listed here are a place to start in diagnosing your high-tech team.

Summary

These are the key needs of high-tech teams:

Support

High-tech teams need to be supported in many ways. The obvious ones are physical space & equipment. The less obvious include information about other internal projects and external competitors.

Being in the loop

Teams need information to be flowing up and down the management chain. They need to be reassured about stability of their project (even if the reassurance can only say, “you’re good for another couple of months”) and the intentions of the top management with regard to the product.

Credibility

Above all else, teams need clear indications that management knows how to do its job. Is there a business strategy? Do the executives know how to keep the company funded? Are they hiring good people? When these questions are answered in the affirmative, teams can stay focused on their work.

Confidence & appreciation

Does my manager have confidence in me? Does he or she appreciate my value? These are critical questions for the individual contributor and important questions for all team members.

Need Number 1: **Support**

The employer provides the work environment. This principle has been around since the Industrial Revolution. Today, in the Information Age, the requirements of the high-tech workplace have evolved to include the office, laboratory space, a personal computer, local and worldwide networks, and everyday application software for communicating and managing documents and appointments.

Specific software tools may be required to do design and development work. These may include design automation tools, drawing tools, simulation, emulation and performance prediction software.

Intangible forms of support for high-tech teams are also needed. These may include

- ☞ team communications media, such as on-line bulletin boards, PERT & Gantt charts, to allow members of the team in one area to monitor progress of other parts of the project;
- ☞ problem logs to communicate bugs & solutions to problems during development;
- ☞ information about other projects and products in the same product line or division; and
- ☞ information about competitors' products

But this doesn't mean that high-tech teams really need everything they ask for. Some of the things that are not necessarily needed for successful projects are:

- ☞ Plush office or cubicles
- ☞ The fastest PC available
- ☞ The latest lab instrument
- ☞ Assurances that no matter what product the project team produces, it will be sufficiently competitive with what other companies will be selling

And finally, the project team may have to compete with internal competitors.

Whether the team should be defended against internal competition or not is a decision that should be based on an evaluation of the team's capabilities and a realistic assessment of what is best for the company.

*Case A: Dragon Project [Digital Equipment Corporation 1974-1975]
"internal competition", "realistic schedules not expected?"*

I learned a lesson about internal competition early in my career as a computer designer. Two years after I had joined a minicomputer manufacturer on the East Coast, I was named Project Engineer on a new computer, intended to be a third-generation "mid-range" replacement in a very successful computer product line. Both the Engineering Manager and his boss, the Director of Mid-Range Systems, were new to the company (from the same company), and most of the staff were new hires. Our project was ambitious – a multi-processor-capable design using a wider data bus than previous designs – and I had recruited another young Ph.D. in Computer Science to be chief architect on the project, a role not usually distinct in this company.

While the project was ambitious, we were fully staffed and we resolved to create a realistic schedule – one that could actually be met; this was not the standard practice in this company, where projects commonly overran their schedules by 4 to 6 months. We anticipated delivery of the product approximately two years from the launch of our project. The first 4 to 5 months were focused on design and testing of the new multiprocessor-capable bus.

The line managers were supportive of our project, but their understanding of the company's dynamics were as naïve as mine. When the competitors began introducing next-generation mid-range products near the end of the first 9 months of the project, the marketing department began demanding an earlier delivery, even if the resulting product had no multiprocessor capability. We were all surprised when a small team of "old hands" underbid our schedule

by proposing to deliver a product in only 9 months, 6 months sooner than our predicted end-date.

They got the job, and our team was disbanded. They actually delivered their product in the month of our predicted end-date. Fortunately (for engineering), our multiprocessor bus design was incorporated into a successor product line.

Need Number 2: Being in the loop

Quote

“If you don’t commit to the idea of trusting people and to the free flow of information throughout an organization, developing teams isn’t just a wrong idea, it can be catastrophic.” –Robbins & Finley, *“The New Why Teams Don’t Work”* p. 221

High-tech teams need a free flow of information about the stuff that management is responsible for, such as

- ☞ Is the company OK? Will it be here for the next few years as a going enterprise?
- ☞ Will we actually sell the product that I ‘m working on? Are there customers for the product? Have customers given us any feedback on previous products?
- ☞ Are there competitors who may offer products like this one before we do? Or with better features?
- ☞ Is the company planning layoffs or planning to hire new managers?

Teams also need direction from management that is consistent with executives’ statements about business goals.

*Case B: Magnum Project [Quantum Corporation, 1989] “don’t overload”,
“cut your losses”*

I had been consulting for this peripheral equipment manufacturer for years, as an advisor on computer systems technology and software. As I was walking out of the building one day, the VP of Engineering grabbed my arm. “I’ve got another project for you,” he said.

The company had chartered a new development approach using a high-level language for firmware development for the first time. In addition, the firmware team included a Chief Firmware Architect and a Program Librarian, roles that were not usually present in this company. A Program Manager new to the company had just started.

The firmware manager was a good administrator who had little experience in management. The Chief Firmware Architect, who had an MS in Computer Science, was not very assertive. The two of them together did not add up to a manager. I moved into the firmware manager’s office with him, and we co-managed the project for 7 months.

In my first two days on the job, I identified two members of the firmware team who were not productive. One was the program librarian. The other, also with an MS in Computer Science, was responsible for the critical read-write code. He was also a good friend of the Chief Firmware Architect. But he had no sense of discipline in producing working code. This situation was confirmed by virtually all of the firmware team members.

At the end of the second day, I recommended to the Program Manager that he terminate the two unproductive members of the firmware team. He said, “I’d like to wait a while before doing that.” I decided not to insist on action.

That decision was a mistake. The project fumbled along for months before it became obvious to everyone that this team would never complete the firmware. No matter how well I coached the firmware manager, he could not get results from a team that included people who were not contributing.

There are two lessons in this story. First, don't overload a project with new tools & methods. Second, always cut your losses: remove non-functional team members as soon as you discover them. The rest of the team will appreciate it, and you'll be back on track much quicker that way.

Need Number 3: Credibility

This need is about the employee's belief in the capabilities of his or her manager. The ability to inspire belief or trust is a key attribute of the successful manager. The first-level manager represents all of the company's management to the project-level employee, so everything that comes down to the employee is associated in that person's mind with his manager. That's why you often hear project-level people talk about "the management," meaning everyone from the CEO on down, as a collective whole.

Here are some of the key things that a high-tech team wants to believe in, indicators that management knows what it's doing.:

- ☞ Are the executives keeping the company afloat? Without any accounting gimmicks?
- ☞ Do we have a viable business strategy?
- ☞ Is the company selling the right product into the right market?
- ☞ Can I count on the company to hire the right people?

☞ Is the management asking me the right questions?

In the following case example, both of the last two questions were clearly being answered in the negative.

Case C: Disk File System (Singer-Friden, 1969-1970)

Singer, the sewing machine company, acquired Friden, the calculator company. In the late 1960s, Friden introduced a series of electronic cash registers (called Point of Sale terminals) that connected to a computer in the back of the store. Programs in the computer kept track of the sales information generated by the terminals. Centralized computer systems would call the store's computer by telephone every night to get the store's sales totals. The availability of this information revolutionized merchandise management in retail chains.

I was still a graduate student when an old friend of mine, whom I had known as an engineer on the staff at my university, called to ask my help. He was Singer Friden's manager of systems engineering for Europe, based in Brussels, and he was frustrated. For some reason, he could not get what he needed from the home office in San Leandro, California.

What he needed was software. Software that would manage files stored on disks attached to the Friden computer system. "Would you go over to San Leandro and find out what's going on?" he asked.

He arranged the introductions, and I soon found myself listening to the software team at Friden. They were competent people somewhat frustrated by the constraints of the computer. Because cash register marketers had defined the computer, it was slow, inflexible, and the programs were coded in binary-coded decimal, just like the data.

Then I met the programming manager. It was a short meeting; I found out he knew literally nothing about software. I asked to meet his manager, and was introduced to a busy man whose main concern was to delegate things to his staff. There were many things more urgent than planing for future software features.

Back with the programming team, I asked them what they thought about the request to build a disk file system, which had just come down from the management. They laughed. “How do you think we’ve been running our compilers and assembler?” they asked. “Of course we’ve got a disk file system. It’s just that the management never thought to ask for one until now.”

The combination of an unaware manager managing an incompetent one was deadly at Singer Friden. These two managers protected each other, and they kept any useful information from flowing either up or down in their department.

It was not a surprise that the product goals were ill defined. In this situation, the programming team did what any reasonable technologists would do: they built what they thought was needed. And they did their best to ignore the directions received from the incompetent manager. This situation usually leads to product failure – but not always. Sometimes the view from the bottom is good enough to get by on.

Need Number 4: Confidence & Appreciation

A high-tech team member, like every employee, wants to feel that the manager believes in the employee’s capabilities; is confident that he or she can do the job, is doing it, and has value to the company.

This sense of appreciation is not the same as praise, or high pay, or freedom to do what the employee wants to do. The high-tech team member does not mind receiving direction from the manager, as long as the manager satisfies the needs of the employee.

A sense of confidence and appreciation, however, must precede direction, which places the manager in a difficult dilemma. How to appreciate an employee who has not yet actually contributed something concrete to the company? This is where imagination comes in. As a manager, you must be able to visualize successful effort coming from every employee, before you witness it. Your vision helps to inspire the employee's confidence in his or her own abilities.

Case D: Lisa LAN (Apple Computer, 1979-1980)

The Lisa was Apple's first big step towards an integrated desktop computer, ultimately leading to the Macintosh series. In 1980, as part of the Lisa hardware team, I took responsibility for developing a local area network (LAN). In this role, I supervised a very capable hardware engineer named Bob.

I formulated the theory that engineers require their manager to be able to do the engineer's job. If they don't think you can do their job, they don't follow your direction. Or so I thought until I met Bob.

I was particularly impressed with Bob's creativity in logic design, and after one session where Bob was showing me his work, I told him so. He looked puzzled. "But I had to explain to you how this circuit works," he said. "I'm just curious. Why didn't you understand it just by looking at it?"

I explained to him that I had never designed computer logic as part of my job since I had started working in industry. And it wasn't a big part of my work as a graduate student in Computer Science, even though I specialized in computer architecture – the big picture in computer design. "Computer Science!" he exclaimed. "I thought your degree was in Electrical Engineering! No wonder you're not a guru in logic design."

Something changed in my relationship with Bob at that moment. Afterwards, he didn't mind explaining his designs to me. And he seemed to be satisfied with my appreciation of his work. Somehow, to Bob, a computer scientist is competent to grasp the significance of superior logic design, even if he isn't capable of doing it himself.

The lesson is this: top-notch individual contributors require their managers to be able to appreciate their work. That means appreciate it in a deep way, knowing what it is worth in relationship to the product. And what it saves relative to less-creative design work. This is a particular form of credibility that is worth cultivating when you are managing creative individuals.

The employee who receives your appreciation is willing to explain what he or she is doing; and is ready to ask for your input and direction on the project.

Conclusion

Everyone wants to be part of a team that is supported; receive the confidence and appreciation of their management; have access to information about the company and its competitors; and believe in their management. These are the things that enable teams to perform their best.

People in high-tech teams are always ready to talk about how they see their management's performance. If you can listen to them and understand the meaning of their complaints and compliments, you are well on your way to managing a successful team effort.

Are you listening?