

LEARNING HOW TO LEAD GEEKS

A Conversation with C2 Consulting's Paul Glen

If you've ever given what you thought was an incredibly passionate speech to a roomful of unmoved, unblinking developers, you can take solace in knowing that you're not alone. Indeed, C2 Consulting's Paul Glen has seen so many similar scenarios throughout his years as a management advisor that he felt compelled to write a book about them.

Leading Geeks (Jossey-Bass, 2002) doesn't spend all 253 of its pages examining uncomfortable silences in meetings, but it does offer plenty of insight about why working with developers is different than working with, say, your sales staff (a group of people much more likely to jump out of their seats and give high-fives all around after your big speech).

To learn more about what it takes to manage developers, we recently got in touch with Glen and peppered him with questions. Here's what he had to say:

SD: The title of your book, *Leading Geeks*, suggests that managing developers requires different skills than managing employees in any other field. What are some of the differences?

PG: I'm not sure that I would agree that it requires different skills as much as different attitudes, understandings and techniques. The effectiveness of a leader is not measured just by the sum of his or her skills, but by the quality of the relationship between leader and followers, and the collective results created by the group.

That said, there are three key differences between leading developers and other employees:

First, developers are different from other employees. Among those who choose to become developers, there are common patterns of behavior, attitudes and values. These patterns influence how one should lead. For example, developers tend to be more loyal to their technology than to their company or even project. For most, the technology draws them to a career in development, rather than an attachment to some specific industry.

Developers also tend to have what I call the "passion for reason," a strong sense that all things are or should be completely rational rather than emotional. So if you're trying to motivate a group of developers to

go out and work hard, the emotional, whip-up-the-passion approach so common in sales organizations usually falls flat.

Secondly, development work is fundamentally different from other work. This form of creative work doesn't conform to the manufacturing model that most management theories posit. What you would lead someone to do significantly affects how you should lead them.

Here, the inherent ambiguity of technical work makes traditional leadership quite hard to do. Since most leaders think that their job is to tell people what to do, they become quite frustrated by technical projects, where one of the biggest jobs is to figure out what to do.

In development projects, figuring out what to do is often harder than doing it.

And finally, power is useless with developers. This is not because developers are recalcitrant, but because power is about influencing behavior. But developers deliver most of their value through their thoughts, not their behavior. Traditional approaches to leadership are based largely on notions of power and aren't particularly useful when it comes to developers.

SD: One of the biggest problems with software project management is completing projects on time and within budget. What's your advice to project managers grappling with these problems?

PG: Collectively, we have a rather dismal record at delivering projects on time and under budget. Despite the fact that most studies indicate that while we have made significant progress over the last decade, we still only manage to complete about a quarter of our projects on time.

Examining the reasons for these failures, one discovers that the majority of projects fail not due to technical problems, but due to difficulties in leadership, management, client relationships and teamwork. In short, human problems doom projects, not technical ones.

This is why leading geeks is such an important topic. The failure of leadership is responsible for the vast majority of project disasters.

Of course, I have plenty of advice for project managers and other leaders in my book, but if I had to give only one piece of advice, it would be this: The focus of a project manager should be more on managing the people who do tasks and less on managing tasks.

SD: Every project has its stars and its underperformers. How should managers modify their management styles to more effectively work with both types of employees?

PG: Everyone wants to have star performers, but as I talk to my clients, I find that they often have very different definitions of what a star performer is. Sometimes it's the fastest programmer. Sometimes the best architect. Sometimes it's the most personable developer. Sometimes it's the most politically savvy person in the group.

What I recommend for managing both star performers and the less than stellar is the same: A manager must become very clear about what top-notch performance is and then become articulate in describing it. You can't expect developers to become stars if you can't tell them what a star really looks like.

SD: Clearly, many companies are still facing layoffs and budget cutbacks. How does one keep developers motivated and focused during these times?

PG: Motivating technical staff is difficult at all times. Unfortunately, most management literature written about motivating staff is targeted toward motivating sales staffs, not developers, who are quite different. Money has never been a key factor in motivating developers to produce their most creative work. Lack of money can serve as a de-motivator, but excessive money never made anyone more creative.

What tends to motivate developers most of all is engaging work, the opportunity to learn new things, fair pay and the prospect of a future filled with more of the same.

Further Reading:

Leading Geeks: How to Manage and Lead People Who Deliver Technology

(Jossey-Bass, 2002) by Paul Glen

<http://click.sd.email-publisher.com/maabpGHaa0f82bdnv4bb/>

COPYRIGHT:

SD People and Projects is a monthly newsletter brought to you by CMP Media LLC, publisher of Software Development magazine.

Copyright 2003 CMP Media LLC