

# Leadership Is About Responsibility, Not Authority

*Seth Dobbs*



I am often asked questions along the lines of, “The development team has more experience than I do, how can I be an authority to them?” This describes a difficult situation in which someone with less experience than the bulk of the team members has been promoted to a leadership role. A similar scenario is someone recently added to a team who has been working on a product for years and the newcomer architect has little depth of knowledge. In either scenario, the new leader needs to gain credibility in order to be effective or risk being undermined by the team. More broadly speaking, there can be a lot of discomfort in being a newly minted leader trying to tell a team what to do.

And that’s exactly where the problem comes in: we as leaders shouldn’t think our job is telling people what to do.

Thomas Chandler Haliburton, a nineteenth-century Nova Scotian politician and author, observed that, “Wherever there is authority, there is a natural inclination to disobedience.” It sounds a bit like he’s worked on one of today’s software development teams! In fact, it can be fairly common among knowledge workers in general to question authority and to sound out the depth of a new leader’s knowledge and then use their “superior” knowledge to discredit and/or disobey the leader.

So what’s a new leader to do? For me, the answer forms one of my core principles of leadership:

Leadership is responsibility, not authority.

In other words, we as leaders have a responsibility to make our team members succeed, not an authority to make our ideas hold.

This can be a difficult lesson. Some seek leadership roles because they believe it's about authority. And, sure, organizational authority is a thing. Authority can come with hierarchy and with depth of knowledge. Role power is real and can be used for good or ill. Indiscriminate use of authority and role power will often not yield durable results. Worse, it can't be used for situations in which you don't actually have hierarchical role power over the people you lead.

Looking back to the question that started this, the architect who first asked me this didn't have role power over the development team, which is often the case for leaders in flat organizations: the people they lead are matrixed in from other parts of the organization. The person also readily admitted to me that they had less experience and technical knowledge than the average team member.

So, the answer to the question, "How can I be an authority to them," is often, "You can't, but don't worry about that." Being a leader doesn't mean being the best "doer"; it means being a leader. So instead of trying to wield authority, I'd encourage you to tell the team that you're leading simply and directly that your responsibility is to help them succeed at their jobs. Get into a dialogue and find out what they need from you to succeed and help them help you by giving you what you need to succeed.

In the case of a new software architect, you're providing them guidance on how the broader system is coming together, and they in turn can provide you deeper details of what's been implemented. A new development manager could let the team know you're there to help ensure the team delivers value to the business and that you need to understand constraints, risks, and trade-offs in time, budget, and scope so that you can manage expectations. An organizational leader should help the team understand that your responsibility is to help them grow in a meaningful way that ideally also benefits the company.

It takes a little humility to approach leadership this way and perhaps even some rethinking of your role, but if you see leadership as a responsibility to help others succeed and take that responsibility seriously, you'll be one of the best leaders your team has ever had.