

Countless studies have shown that in the software/tech industry your core skills are only the surface of what's most important for a particular career/job. What really matters is your ability to communicate, solve problems and empathize. Essentially your social skills and your emotional intelligence.

How do you screen for this when you're interviewing candidates for dynamic positions of varying levels across multiple disciplines? The answer is to **Start With Why**. Simon Sinek's book and philosophy on leadership and culture is an absolute *must read* for any leader or hiring manager in any capacity. When you *Start With Why* you are aligning your team(s) on the same mission and objectives. After establishing *why* and being honest and diligent (and patient!) you can focus on the soft skills that matter with an understanding and ability to focus on what really matters for doing great work on your team(s).

There are as many or more studies and conversations and blog posts that have for *years* attempted to dissuade organizations from the traditional whiteboard interview, and yet here we are continuing to see it and hear of it being done even at major companies.

Rather than ask a candidate to solve an arbitrary function or method on a whiteboard (insisting you are not concerned about syntax or precision) which you have undeniably vacillated on for countless hours, ask a candidate to explain to you a tricky algorithm or problem they solved in code. For example, if you really mean what you say, do not ask a candidate to "solve" an example exercise of writing a function `int is_anagram(string str)`, ask them to share and walk through a piece of code they actually wrote, and more importantly is actually of use.

First of all, any candidate worth a damn will have prepared the basic computer science exercises ahead of time, and will be able to recite much of the textbooks to you, so you're not really learning anything except for their ability to prepare and take things seriously; which while valuable and important, is not going to find you the "bar raiser" candidates.

However, if you ask a candidate to explain a problem they had, and gave them the time and freedom to explain the problem and their solution, you will gain immense understanding of not only how they think and work, but you can begin to poke holes and get a real

understanding for what they specifically worked on.

After all, if we're being honest with ourselves, any coding exercise is a solved problem that *virtually no* engineer will ever think about except to google search the function header for how to use it. And furthermore, if you're asking engineers to write code in notepad instead of a professional IDE, no candidate worth a damn will work for your organization, so you can safely assume basic syntax and formatting will be handled by the tools they use every second of their professional life.

Share this:

- [LinkedIn](#)
- [Twitter](#)
- [Email](#)
- [Print](#)