

Value Results, Not Just Effort

Venkat Subramaniam
Broomfield, Colorado, U.S.



DEVELOPING SOFTWARE TAKES A LOT OF EFFORT. However, if you hear someone brag, “I work on an application with over 3 million lines of code,” ask him or her how many of those lines of code are really needed.

Often, extra code is added with some perceived extensibility* in mind. Extensibility is important, but if not done correctly, it can have the opposite effect. It can delay your current project.

Extra, out-of-scope code is a symptom of software project managers who reward only extra time and extra effort. If you routinely insist that the programmers work long hours, be sure they are actually producing additional, useable results.

I like my lawn to be green, and rely on my sprinkler system to water it every day. My first summer in Colorado, I noticed that one of my maple trees had lost most of its leaves. Assuming that the hot and arid conditions were the reason, I watered longer but noticed no improvement. The expert I consulted asked me, “How frequently and how long do you water?” Hearing my answer, he said, “That’s the problem! Reduce the duration and frequency by half, and you will see improvement.”

I was killing the tree with excessive water. Having slightly less water actually helps these trees. It builds their resistance and helps their growth. Two weeks after following his advice, my tree was healthy and full of leaves.

Your programmers are like maple trees when it comes to work time. Give them small, but adequate amounts of time and fewer broadly defined tasks, and they flourish. Give them larger task chunks and ask them to routinely work extra

* *Extensibility*: A systems design principle where future growth is taken into consideration. The ability to create and implement additional features is maximized while coding the currently needed functionality.



hours, and they begin to wilt. Plus, they tend to overwrite and complicate the code, since they have too much time on their hands.

I worked for a manager who focused on how long people worked. Working a Saturday morning, or staying late in the evening, was more important to him than what employees were actually producing. It is impossible to be a productive and effective programmer for 12 hours or more a day.

In another group, the manager kept us to a traditional eight-hour work schedule. Yes, there were days we stayed late, but those were exceptions rather than the norm. Employees knew they were not required to work long hours but had to provide their committed deliverables on schedule. So, we were focused and less distracted, prioritized our work well, and used our time effectively. Even though developer capabilities were about the same in both groups, we got more accomplished in the second group than in the one where we worked to exhaustion.

Encourage programmers to report the progress they make, rather than how long they work. Let them know that you care about getting results rather than keeping track of how long they spent at the computer. Once your team members realize that you are a results-oriented manager and not a “put in hours” manager, their focus will shift to achieving results rather than merely clocking hours at work.