



Problem Solver – Resource Management

Problem

You have a very talented and technically competent software developer on your team. While he is not formally the technical lead (no one is), he feels a significant ownership of the software product the team is developing.

He often does work beyond his role and reviews every line of code checked-in by other developers, changing things he does not like without telling them. He likes to work behind closed doors, does not interact easily with the team and often misses team meetings and takes days to respond to questions from other members of the team.

He is a valuable and passionate member of the team so you do not want to disengage him. Your attempts to coach him in improving his team skills have had little effect. What do you do next?

Solution

In formulating the solution I have presumed that this project is already well progressed and that changing staff at this late stage would be counterproductive. I have therefore assumed that retaining the services of the software developer is a desirable part of the outcome sought. The other outcomes sought are greater team cohesiveness, efficiency and effectiveness. In particular:

- ☞ eliminating unilateral code changes;
- ☞ improving the skills of the other developers;
- ☞ increasing attendance at team meetings; and
- ☞ improving the responsiveness of the software developer in question.

I would establish a formal code review process by which all code was distributed amongst all developers on the team and discussed at a code review meeting to be chaired by my very talented and technically competent software developer. However, before announcing this measure I would take the software developer aside and point out the impacts his/her practices and behaviours have on efficiency, team morale, team development and configuration control. I would then go on to say to the software developer that they had been

appointed as the chair of the code review meeting in recognition of their performance and their potential to lead and mentor the other developers. I would conclude that the purpose of the code review meeting was to give the other developers the opportunity to discuss the code and to hear the talented software developer's reason for any code change. In this way other team members could learn from the differences and ensure that they did not occur in the future. This would have the effect of reducing the amount of repetitive code changes currently done.

I would also point out that as the chair of the code review meeting he was required to attend all team meetings to report on the progress of all code reviews.

This approach would force the talented software developer to engage with the other developers and give the other developers a forum to ask questions not only about coding practices but also to answer any questions they might have.

If and only if this approach does not work would I take the talented software developer aside and remind him/her that their cosy office is not a right but a privilege earned. This would leave you with a place to go if the talented software developer failed to rise to the challenge.

Lessons

Human resource management is about ensuring that the people you value feel valued. But that does not mean you should be a push over... every carrot has a stick. As with every significant risk you should have not only a risk treatment but also a contingency plan.

This Problem Solver was summarised and published in the April/May 2012 edition of the *Project Manager* magazine