Leadership for a mixed team of junior and senior developers

Adrian Vladu
European Master in
Software Engineering
Blekinge Institute of Technology
Karlskrona, Sweden
Email: adrian.vladu21@gmail.com

***Abstract*—Leadership is done by humans, not tools.**

I. Techniques used

Analyze each team member, using the available formal or informal information: history in the company, results, CV-s, social and cultural activity. Profile each team member through the personality characteristics. The profiling should be a continous process, as people change over time. Enabling the process will give a better insight on the current and future actions of the team.

Use agile methodologies, because are focused on communication and collaboration, which should easily level up the distance and knowledge between the team components.

Organize trainings held by both of the two parts - the young will give a training on a newly appeared technology and the seniors on existing technologies that they master perfectly

Organize a teambuilding once 6 sprints, considering a sprint of 2 weeks, so that people get to a better understanding of also the social, moral values of the other team members.

Associate a younger developer a senior developer as a supervisor and pillar of knowledge. When the junior is in neeed of help, the senior will be the first stop when seeking help.

Allocate to the junior developers 20% percent of the time spent at work to learn new things, without any pressure from the leader or management side.

Involve employees to set their own goals, as a team.

Security as a need - always have a peer on whom he can count on, a junior can feeel more secure. It will achieve predictability by delegation. Although the peer may not be in the position to always solve the junior's problem or to direct it in a right direction, it will suffice by giving more confidence.(page 43)

Acknowledge and address the danger - unwanted behavior, of undesired unofficial groups being created or alliances between the team members of same age. In the same time, people are social and gregarious beings, so group formation must not be interdicted or suppressed, only controlled and directed to organization's and project goals.

Competence - break things and then try to put the things together again. It will inspire the juniors and they will believe more in their capabilities and power. Achievement oriented developers. (page 50)

II. Ideas

The project manager has a fulltime job. Situational leadership. Create value as perceived by followers. Power changes from seller to buyers - Kickstart/Indiegogo. Technology is getting borders closer. What can be done will be done, sooner or later, if not by you, by some one else. Real leaders believe in change. Behavioural sciences deal with probabilities. Managers do things right. Leaders do right things. Performance starts from bottom-up, with the one-to-one relationships. Are leaders born or made? Diagnose, adapt and communicate. Plan, organize, motivate and control. Understand, predict and change behaviour. Control means manipulation and viceversa, depending on the point of view and the expected outcome. :) uuf, I am so relieved. A hammer won't always do the job. If you want to change your behaviour, you have to practice. Reading/learning will give you just a conceptual view. :)) At last, someone agrees with me. [1]
A goal that it is appropriated for a 6-year-old may not be a meaningful goal for a 7-year-old. Coworkers can participate in setting their own goals. Keep the carrot in the donkey's reach.(page 32)

Although *David W. Galenson*, in his work *"Old Masters and Young Geniuses: The Two Life Cycles of Artistic Creativity"*, refers to plastic artists, I strongly believe that the same ideas can be applied to programming and software engineering. As a consequence, software products can be considered as the unique resut of an art form. The team members' creativity and past knowledge can balance each other. That if the process is managed properly, with the leader acting as a middleman between the two molding forces - innovation and reuse. The ability to achieve a goal, to finish successfully a task can be native or hard-earned. One may innovate successfully at the conceptual level, but it requires a long time of trial-and-error or learning for experimental and solid innovation. The ability to radically innovate can decrease with age, as a person is inclined to self-establish a set of rigid methods and conventions. [2]

"When a situation requires a new way of looking at things, the acquisition of new techniques, or even new vocabularies, the old seem stereotyped and rigid....But when a situation requires a store of past knowledge then the old find their advantage over the young."

— Harvey Lehman, *1953*

V. Conclusion

Acknowledgment

The author would like to thank to his teachers.

References

[1] Paul Hersey, Kenneth H. Blanchard and Dewey E. Johnson, *Management of Organizational Behavior: Leading Human Resources*

[2] David W. Galenson, *Old Masters and Young Geniuses: The Two Life Cycles of Artistic Creativity*

[3] *Harvey Lehman, 1953*