

Recognizing Achievements Great and Small¹

Karl E. Wieggers

Process Impact

www.processimpact.com

When I first became the manager of the Kodak software group in which I had worked for several years, I initiated a simple (and slightly corny) recognition program. When someone reached a minor project milestone or made a small contribution such as helping another team member with a problem, I gave him a package of M&M® candies, with a message tag attached expressing congratulations or thanks, as appropriate. Bigger achievements generated bigger bags of M&Ms, or something more tangible. It wasn't much, but it was more than we were used to.

As I expected, the candy disappeared immediately, but I was pleasantly surprised to see that some people kept the message tags visible around their desks. To them, the important thing was not the bag of candy, but the words indicating that their manager noticed and valued the progress being made. It soon became apparent that group members preferred to have the presentations made publicly at our weekly team meetings, indicating their desire for peer recognition of even small achievements.

M&Ms worked with our group, but some other social recognition technique might work better for you. We also gave this sort of micro-recognition award to people outside the group who helped us in some way. It brought smiles to their faces and goodwill to our relationships. However you choose to do it, appropriate praising and commendation helps to build the culture of teamwork and striving for excellence that we all want. Recognition can motivate your team members to do an even better job in the future, since they know you appreciate their efforts.

The form and extent of recognition and reward is a visible indication of an organization's culture. If managers believe the employees are lucky just to have jobs, they won't go out of their way to offer even small gestures of appreciation or congratulations. Conversely, in a market characterized by competitive hiring and high staff turnover, an effective recognition program can help retain talented developers. M&Ms won't make up for low salaries or unpleasant working conditions, but simple recognition is an important step in the right direction.

Like anyone else, we software engineers want to be appreciated, and we appreciate being wanted. Besides the internal satisfaction we obtain from interesting, challenging work and the tangible compensation in our paychecks, we want to feel that our efforts are noticed and valued by those around us. We all enjoy receiving compliments, especially when they come from various sources: peers, customers, team leader, senior managers, professional associates.

Praise for a job well done should be timely, direct, personal, and specific. If you are a manager, don't wait until performance appraisal or salary adjustment time rolls around to pass along some positive feedback. Tell the individual exactly what he or she did well and why you appreciate it. Though many people feel awkward when they receive a compliment, they appreciate

¹ Published in *American Programmer* magazine, May 1997. Reprinted with permission from Cutter Information Corp. Adapted from *Creating a Software Engineering Culture* by Karl E. Wieggers (Dorset House Publishing, 1996).

that someone took the trouble to say how pleased he was with their work. Recognition says, "I appreciate your effort," "Congratulations on your accomplishment," or simply, "I noticed what you did."

Recognition Recommendations

Recognition can take many forms [1]. Ask the members of your team what kinds of recognition are important to *them*. Do they prefer a public pronouncement at the weekly group meeting, or are they more comfortable with private ceremonies? Should recognition come just from you, or is it meaningful to have higher-level managers participate in certain recognition activities? Tailor the reinforcement you offer to be significant to the recipients. Below are some of the things the groups in which I have worked have done to express appreciation and to help grow a healthy software engineering culture.

R+. Spend a few moments at weekly team meetings to give everyone a chance to pass along some positive reinforcement ("R+") to others. Did a coworker help you solve a problem this week? Did someone take some action out of the ordinary that helped the team? If yes, say so! The group may be uncomfortable when you first try this, but they should warm to the idea over time. If group members are so isolated from each other that no one ever has any R+ to pass along, you may have some serious issues of team dynamics to address.

Traveling trophy. A traveling trophy that moves from project to project can be used to recognize team achievements. In keeping with the M&M motif, we used a framed three-pound M&M bag as a traveling trophy. Recipients displayed the prize in their office area until another project reached a milestone worthy of recognition. The trophy was ceremoniously passed from one team to the next in our group meeting. If you try something like this, be sure to keep the trophy traveling every few weeks, or its significance becomes lost.

Bread and circuses. Food and entertainment are also good ways to recognize someone's contributions or special achievement. Taking the team to a celebration luncheon when a milestone is reached can be fun for everyone involved. A gift certificate for dinner at a restaurant gives an individual recipient a chance to celebrate privately with friends or family. On one occasion, I gave each team member a pair of movie passes as a symbol of how much I appreciated their time and teamwork when, during an intense period of hiring new group members, the entire team of ten pitched in on short notice to participate in the interviews. It was a small gesture but a sincere way of saying, "Thanks for the help, gang."

Thank your friends. Recognize individuals outside your group for their contributions as well. It's amazing how much future cooperation you can secure with a simple gesture of appreciation. As the recipient of a few such gestures myself, it always makes me feel good to know that someone really appreciated something I did, however routine it may have seemed to me. We have thanked our internal Kodak project customer representatives by taking them to lunch, giving them certificates to hang on the wall, and bestowing restaurant gift certificates upon those who shouldered the most responsibility. One customer even reciprocated, throwing a lunch bash for the development group. This sort of customer-developer interaction helps build a culture of constructive teamwork.

Interim milestones. Be sure to recognize people for attaining minor milestones, as well as when they complete a big project. Interim pats on the back help provide team members with the incentive to keep pushing ahead. Again, it says to the recipient, "Congratulations on making progress toward your goals."

Carpe diem. As a manager, you must actively look for recognition opportunities, and seize the moment as soon as you spot one. The manager who realizes an achievement is worthy of formal recognition but waits to figure out what he wants to do about it, and waits to execute his plan, may provide recognition too far removed from the achievement itself to mean much to the recipient. "Oh, so you finally noticed what I did," is a typical unspoken reaction to a belated recognition effort. Also remember that managers who fail to reward exceptional contributions are sowing the seeds of discontent. The absence of well-deserved recognition is highly demotivating.

Share the wealth. Be sure to distribute recognition awards equitably to your group members. Don't reserve recognition events only for project leaders, members of high-profile project teams, or your senior technical people. The scale and frequency of rewards does not have to be the same for everyone—after all, people and their contributions are different—but it is demoralizing for an employee to see the same coworkers being recognized repeatedly without anyone noticing his own achievements.

The Importance of Being Visible

The antithesis of being recognized for your achievements is feeling that your managers do not know who you are, what you do on the project, how you do it, or what your contributions are to the company as a whole. When was the last time your supervisor stopped by your office just to say hello and ask how things are going? How about a visit from a manager farther up the corporate hierarchy?

Some people are uncomfortable with an unannounced manager visit, but others welcome the opportunity to share their concerns and show the boss what they are working on. "Management By Walking Around" is one way managers express interest in the individuals in their organization; it can and should be practiced by managers at all levels. If you are a first-line supervisor and you never see your boss in the engineering staff's offices, chances are you'll conclude that he or she is hopelessly out of touch with the group.

People are more motivated to put in extra effort when they know the higher-ups value it. We have all worked for managers who represented the other extreme, having a limited awareness of the group's challenges and contributions. How excited can you get about trying to please such managers? As a supervisor, make sure you really know what the engineers in your department are doing. Who are the contributors, the innovators, the leaders? Who is just along for the ride? The team members will have no confidence that they can get fair performance appraisals if they rarely talk to their supervisor. Your employees need to have adequate opportunities to explain what they do and the problems they face.

One-on-One Meetings

Make sure you are accessible to the people who report to you. Schedule one-on-one meetings with those who desire them, at whatever interval is mutually acceptable to you and each individual. A general open-door policy is important, too, but your team members may be reluctant to bother you if they know how busy you are. They deserve a slice of your undivided attention at regular intervals.

These one-on-one meetings should be one of your top priorities. Avoid the temptation to cancel them, cut them short, or interrupt them for reasons like these:

- Someone else has already stopped by your office to chat.
- You need to work on one of your own projects.
- Your telephone rings.

- You have been invited to join a new committee that meets at that same time.
- Your boss calls another meeting for that time.
- You forgot about it.

Whenever anything short of a true emergency interferes with a scheduled meeting with someone you supervise, you are sending a clear message: "Everything else I have to do is more important to me than you are."

Attitude Adjustment

Here's a radical idea: Think of you, as a manager, working *for* the people who report to you, as opposed to the more traditional view of subordinates working for the supervisor. The people you supervise are the customers for your leadership and management services, including

- coaching, mentoring, and career development
- setting project goals and priorities
- resolving problems and conflicts
- providing resources
- evaluating performance and providing feedback
- providing technical guidance when appropriate

Give your own people top priority over the demands of others for your time. Your priorities as a manager should be those shown in Fig. 1.

Unfortunately, too many managers are busy looking *up* the corporate organization chart, not down. The sequence in Fig. 1 is frequently inverted, with the key driver being what you think will make your own boss happy. In a healthy, congruent workplace, the boss should be thrilled if you are meeting the needs of your team and its customers. Not everyone is fortunate enough to work in such an enlightened environment; priorities are usually defined by the perception of who you think you have to please to keep your job.

Figure 1. Priorities for the enlightened software manager.



One way a manager can tell if his priorities are straight is whether he ever receives any recognition from his team members. It meant more to me to get an R+ from one of the people I supervised than to get one from my own manager. Remember to thank your own managers for special contributions they make or extra help they provide to you. Managers are people, too, with the same desire to be appreciated that anyone else has.

Rewards for a Job Well Done

The skillful manager will use recognition and rewards to reinforce desired behaviors, rather than offering tangible incentives to individuals or teams to achieve specific goals. Dangling extra cash as a carrot in front of a software engineer to try to get him to work faster-harder-longer can backfire [2]. If the ambitious goals are not met, and developers know the pot of gold at the end of the rainbow won't be forthcoming, how do you keep them motivated? Withdraw the incentives (thereby destroying morale), or renew the incentives (thereby showing that falling short of management's outrageous goals is just as meritorious as achieving them)? Either way, everyone loses.

Instead of offering incentives, design a reward program that matches your organization's culture and means something to your team members. Motivate your team through frequent interim recognition activities, and reward them for a job well done when the job really is done. People should also be rewarded for taking intelligent risks, even if a great notion or great effort doesn't pay off for reasons outside the development team's control. Public rewards indicate to the rest of the group those behaviors you feel are desirable.

If you make a verbal commitment to someone for a reward, be sure to follow through on it. Forgetting that you made this promise demonstrates a lack of sincerity. Be sure to reward the right people for the right reasons. If you aren't sure who made key contributions to a successful project, find out before you present any rewards. Few things are more infuriating than seeing a person receive praise (or more) for work that was actually done by someone else.

Rewards can be monetary or non-monetary. Talk to your people and understand what rewards they feel are significant. The corporate culture will have some influence in selecting feasible rewards. It may be easier to give someone a substantial, but non-cash, award, such as a trip to a conference or trade show. Some companies reward employees with frequent flyer miles, which can be purchased from airlines for a few cents per mile. Some developers might enjoy extra vacation time; others might want to buy a special software package with which to experiment, just because they heard it was interesting. If someone does an exceptional job on a project, he might appreciate a mini-sabbatical, a couple of weeks set aside to work on whatever he likes.

Another kind of reward is the opportunity to work on an exciting new project. In some groups, only the old hands have the skills and knowledge to keep the legacy systems alive. Less experienced people may be assigned to projects involving newer technology, which are more fun than maintaining ancient applications. This is a good way to drive your senior staff members out of the group, to search for opportunities where they can learn contemporary skills and work on the kinds of projects they read about in computer magazines. Creative and experienced engineers don't want to be mired in legacy code for the rest of their careers. Look for ways to reward your best workers by keeping them stimulated with new learning opportunities and project challenges.

Enhance Stability with Recognition

As two-way corporate loyalty slowly becomes an obsolete tradition, managers must become increasingly creative at building an environment that attracts and retains talented engineers. A fair and consistently applied reward and recognition program should be a part of your plan.

Spending one or two percent of your salary expenses on the gestures that say “thank you” and “congratulations” is cheaper than constantly replacing disgruntled employees who walk out the door to someplace where they might feel more appreciated.

References

1. Deepro, Donna. *How to Recognize and Reward Employees*. New York: AMACOM, 1994.
2. Whitaker, Ken. *Managing Software Maniacs*. New York: John Wiley & Sons, 1994.