Editor: **Rafael Prikladnicki**
Pontifícia Universidade Católica
do Rio Grande do Sul
rafael.prikladnicki@pucrs.br

Evaluating Software Project Managers

A Multidimensional Perspective

Lawrence Peters and Ana M. Moreno

QUALIFIED, MOTIVATED project managers are key contributors to software organizations. Experts have identified a capable project manager as the most important factor in a software project's success.[1,2] Project managers' value to software projects, software engineers, and their companies is unquestionable.[3] Thus, getting the most out of those managers is critical.

The literature about knowledge workers' psychological profiles show that the best motivator is feedback about how well they've done.[4,5] This is consistent with our experience in software project management and consulting. Frequent, detailed feedback can also be a positive learning experience and an opportunity to improve skills.

Feedback and recognition require the evaluation of professionals, who must accept responsibility for their work if they're going to consider assessments as an opportunity instead of a burden.[6] However, the criteria needed to evaluate professionals aren't obvious.

Unfortunately, the literature provides little empirical data about evaluating software project managers. At best, the literature refers to assessing managers in terms of whether a project meets or exceeds its time and cost requirements.[7] Our combined 50 years' experience in software project management (see the "Our Experience in Software Project Management" sidebar) has revealed some best practices for evaluating software project managers.

Evaluating Software Project Managers

Evaluating software project managers should go beyond a hasty analysis of whether their projects met requirements and finished on time and under budget. The managers act in an organizational context, and their work impacts multiple subgroups within organizations. Additionally, each subgroup has its own value system regarding what a software project should accomplish and what constitutes an effective manager.

At least four parties play an important role in evaluating software project managers:

- Senior managers decide which software engineers have demonstrated leadership qualities and performed well enough to merit becoming project managers.
- Customers, either outside or within the organization (external or internal) evaluate whether the software produced met their requirements.
- Software engineers' expectations are different from the other groups. For example, technical excellence is more important for them than financial issues.[5]
- Software project managers have views about their own work and relationships with the other parties.

A holistic approach to evaluating software project managers must address each party's value system and expectations. As Figure 1 shows, each party's value system determines the criteria that the party evaluates. Various criteria are representative of each value system, with the specifics depending on the organization.

In this multidimensional system, the parties can evaluate criteria using a Likert scale, in which ratings run from 1 (worst) to 5 (best). Differences in various parties' evaluations of a software project manager

0740-7459/17/$33.00 © 2017 IEEE

OUR EXPERIENCE IN SOFTWARE PROJECT MANAGEMENT

Our background in software project management has provided us with information about how to assess software project managers' work in ways that provide learning opportunities for them as well as other project stakeholders.

Our experience is as follows:

- We've managed more than 40 software projects—ranging in cost from $10,000 to $20 million—for 25 customers.
- We've spent more than two decades in software-project-management consulting with firms such as Yourdon Inc., DMR Group, and Fujitsu Consulting.
- We've trained more than 1,000 software project managers in companies in Argentina, Australia, Canada, Chile, Mexico, Spain, and the US.
- Ana Moreno belongs to Spain's AEN/CTN 157/SC1 Project Management Committee, which contributes to the ISO/TC 258: Project, Programme, and Portfolio Management Technical Committee.
- We're key software-project-management contributors to international initiatives such as the IEEE Computer Society's and ACM's Software Engineering 2014 curriculum model and the IEEE Computer Society's Software Engineering Competency Model (SWECOM).
- We've taught software project management at the master's degree level in Spanish and US universities for more than 20 years and have supervised master's and PhD theses in the field.
- We've published more than 30 papers on software project management for leading journals, conferences, and workshops.

provide an opportunity for the software project manager to learn how to do better. Differences among stakeholder evaluations can indicate critical issues in the project managers' behavior, and analyzing this can clarify the issues and show where improvement is necessary.

This multidimensional system is based on the balanced-scorecard strategic planning and management concept, which evaluates an employee's or business strategy's potential impact on all aspects of the organization.[8,9] It goes beyond the typical one-dimensional approach, in which reviews of software project managers focus on profit or loss, a process that provides little opportunity to learn from mistakes and improve performance.

Senior Management

Senior management's priorities for assessing software project managers are whether their projects satisfied the client, earned a profit, and finished on time.

Thus, senior managers must assess software project managers' planning and risk-management strategies, which are among the most prominent causes of software project failures.[10] They also must evaluate team management. For some time, experts have identified personnel turnover as a main cause of software project failures and software project managers as the main cause of turnover.[11] Therefore, project managers need team-management and -motivation skills. They also must be able to recognize their own limitations and ask for help when needed.

Senior managers' evaluations should be based on whether the software project manager

- controlled costs acceptably,
- maintained good client relations,
- developed a sufficiently detailed project plan,
- mitigated risk in advance,
- requested help when needed,
- motivated and got support from the development team,
- had relatively low personnel turnover,
- maintained or improved team performance, and
- complied with company policy.

The last item is particularly important. Company policy exists for a reason, but if it contributes to project failures, the evaluation process provides an opportunity to analyze and modify it.

Senior managers must work to be objective in their evaluations. Many evaluators could be the senior managers who sponsored the project manager's promotion. They might not want to criticize the project manager because it could reflect badly on them.

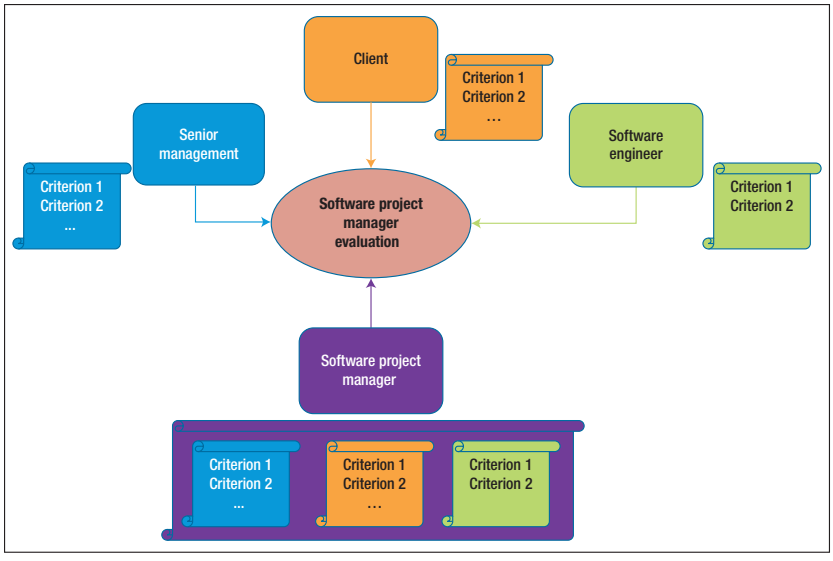

FIGURE 1. A multidimensional perspective on the evaluation of software project managers. Each party in the process assesses a project manager's performance using its own value system to determine its evaluation criteria. The project manager's self-evaluation should address criteria equivalent to those evaluated by the other groups. Each party's appraisal contributes to the overall evaluation.

Clients

External or internal clients pay for software projects, so their satisfaction relates directly to the final product's adequacy. The software project manager's attitude toward and relationship with the client are also important.

Additionally, consistent with other research findings, the client evaluator should assess the quality and effectiveness of communications with the project manager, which is particularly important when difficulties arise.[12] This can be a challenge because the technically adept project manager often must explain complex matters in terms a nontechnical customer can understand.

Clients should evaluate based on whether the software project manager

- produced a system that met or exceeded expectations and delivered value for the cost;
- kept the clients informed about progress, scheduling issues, difficulties, and possible solutions;
- explained technical issues understandably;
- avoided unpleasant surprises;
- produced a system that can be maintained not only by the original development team but also by any other development team;
- is someone the clients would want to work with again; and
- made the project a positive experience.

Software Engineers

Like many other high-technology and knowledge workers, software engineers seek to participate in projects that are or that advance the state of the art.[5] This lets practitioners develop their own technical and professional careers in a supportive working environment. Thus, for software engineers, team management is particularly important.

Software engineers should evaluate based on whether the project manager

- empowered the team and didn't micromanage,
- expressed interest in and concern for team members' success and well-being,
- encouraged quality results and practices,
- was productive and results oriented,
- was a good communicator who both listened and shared information,
- was a good coach,
- had a clear vision and strategy for the team,
- had key technical skills that helped the team,
- made the project a positive learning experience, and
- is someone they would want to work with again.

Software engineers' concern for technical superiority frequently differs from that of the clients and sometimes even the project manager, who generally just want a system that provides the desired functionality, reliability, maintainability, security, and ease of use. This can create differences between the software engineers and the project manager.

Software Project Managers

How software project managers assess themselves compared to how other evaluators view them can be an important way to reveal differences in their appraisals, as well as potential issues with how the managers handle projects.

Software project managers should thus rate themselves using criteria

equivalent to the ones assessed by each of the other parties, based, for example, on whether they

- controlled costs acceptably and maintained good relations with the client;
- produced a system that met or exceeded expectations, and delivered value for the cost;
- kept the clients informed about progress, scheduling issues, difficulties, and possible solutions; and
- empowered the team, didn't micromanage, and expressed interest in and concern for team members' success and well-being.

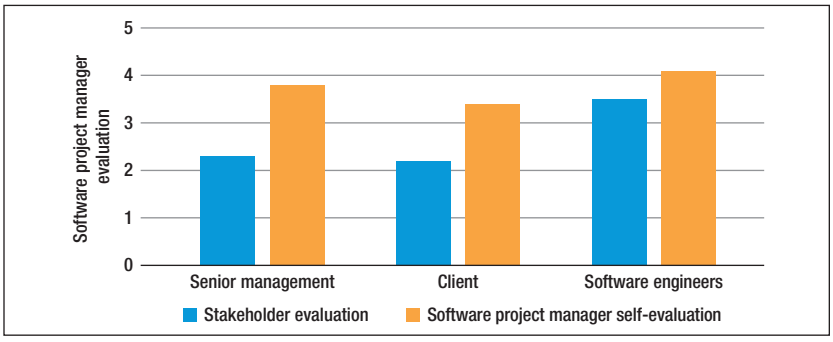

FIGURE 2. A comparison between stakeholders' evaluations of software project managers and the project managers' self-appraisals. The ratings on the *y*-axis are based on a Likert scale that runs from 1 (worst) to 5 (best).

Evaluation as a Learning Tool

People in charge of the evaluation could calculate the total score for each evaluating party's perspective as the mean value of the scores for all the criteria.

If there's more than one set of assessments for an evaluating party (for instance, from multiple senior managers or client representatives), the average of that party's questionnaires could be used. An organization could calculate a composite score for the project manager as the mean of the overall scores for each party, giving more weight to one party's score if deemed appropriate.

The results can be shown in simple graphs similar to those in Figure 2, which let organizations easily observe differences between stakeholders' evaluations and project managers' self-appraisals. Not surprisingly, project managers' self-assessments tend to be better than stakeholders' evaluations.

A large differential (each organization could define for itself what constitutes "large") between evaluations and self-appraisals might indicate a problem with the project manager's performance and self-appraisal. This should prompt the organization to identify the assessment criteria that showed large differentials, to analyze the causes, and to have stakeholders discuss ways to reduce the gap.

For example, if a development team says it's being micromanaged but the project manager says that isn't the case, the project manager might need to modify her planning style and perhaps her understanding of what micromanagement means. On the other hand, software engineers might have to understand why projects sometimes must be planned in great detail and to modify their understanding of micromanagement.

Senior management could work to improve a project manager's management style through successive projects. At the end of each undertaking, they could agree on a score that would be the goal for the next project's evaluations. In this way, managing a software project could become an ongoing self-improvement and learning experience.

The timing of evaluations is important. For example, evaluating project managers only at the end of a long-term project might be inadequate. Instead, intermediate evaluations—perhaps quarterly—might be necessary to identify and correct potential problems before they jeopardize a project's success. To reduce the overhead, the company could limit the number of participants and items under consideration during intermediate evaluations.

Organizations of any size could use the multidimensional evaluations we've proposed. To improve the process, they should automate evaluations via an easy and flexible software tool. Each organization should develop a tool that fits its practices.

Senior managers should also use the fewest criteria necessary to make each party's evaluation effective so that the process won't become onerous. Additional items will probably arise during the analysis and discussion of the data gathered, so organizations should refine the criteria list during successive evaluations.

Organizations should also provide project managers with the list of criteria they'll evaluate, to increase confidence in the process.

ABOUT THE AUTHORS

LAWRENCE PETERS is a project manager, consultant, and part-time lecturer at Universidad Politécnica de Madrid. Contact him at ljpeters42@gmail.com.

ANA M. MORENO is a full professor of software engineering and software project management at Universidad Politécnica de Madrid. Contact her at ammoreno@fi.upm.es.

In addition, they should discuss the results openly with everyone involved and use the process as a learning opportunity rather than as a basis for assigning blame.

And the organization should keep a record of all evaluations in simple but meaningful formats.

Senior managers should be careful when considering very high or low scores. Some evaluators might be difficult to please, and others might or might not like the project manager they're assessing. If senior managers suspect this, they must normalize or reduce the effects of such appraisals' results.

Finally, the value of these evaluations isn't in the project manager's score. Rather, it lies in the potential for the project manager to accept criticism and recognize weaknesses. The goal is to make the assessments a growth experience for all involved, while recognizing the project manager's efforts.

References

1. M.A. Cusumano, *The Business of Software: What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad*, Free Press, 2004.
2. J. Gulla, "Seven Reasons IT Projects Fail," *IBM Systems Magazine*, Feb. 2012; ibmsystemsmag.com/power/systems-management/workload-management/project_pitfalls.
3. D.A. Garvin, "How Google Sold Its Engineers on Management," *Harvard Business Rev.*, vol. 91, no. 12, 2013, pp. 74–82.
4. A.M. Grant and F. Gino, "A Little Thanks Goes a Long Way: Explaining Why Gratitude Expressions Motivate Prosocial Behavior," *J. Personality and Social Psychology*, vol. 98, no. 6, 2010, pp. 946–955.
5. R. Katz, "Motivating Technical Professionals Today," *IEEE Eng. Management Rev.*, vol. 41, no. 1, 2013, pp. 28–38.
6. C.G. Myers, B.R. Staats, and F. Gino, "'My Bad!' How Internal Attribution and Ambiguity of Responsibility Affect Learning from Failure," Harvard Business School Working Paper 14-104, Apr. 2014, pp. 1-53. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1025.7355&rep=rep1&type=pdf.
7. P. Ghazi, A.M. Moreno, and L.J. Peters, "Looking for the Holy Grail of Software Development," *IEEE Software*, vol. 31, no. 1, 2014, pp. 93–96.
8. R.S. Kaplan and D.P. Norton, "Linking the Balanced Scorecard to Strategy," *California Management Rev.*, vol. 39, no. 1, 1996, pp. 53–79.
9. R.S. Kaplan, "Conceptual Foundations of the Balanced Scorecard," *Harvard Business School Working Paper* 10-074, Mar. 2010; www.hbs.edu/research/pdf/10-074.pdf.
10. H.J. Thamhain, "Team Leadership Effectiveness in Technology-Based Project Environments," *IEEE Eng. Management Rev.*, vol. 36, no. 1, 2008, pp. 165–180.
11. E. Cone, "Managing that Churning Sensation," *Information Week*, no. 680, 1998, pp. 50–67.
12. M Chipulu et al., "A Multidimensional Analysis of Project Manager Competences," *IEEE Trans. Eng. Management*, vol. 60, no. 3, 2012, pp. 506–517.