

Conquering Micro-Management

A mini-case study of picking the right "management tools" for your environment

How do you pick the right way to do "project management fundamentals" like scheduling and status tracking for the size and culture of your environment? This mini-case study relates the experience of a small game company's founder and the decisions he made to make PM work for a fast-moving, bureaucracy-hating team.

The situation

At a software game development company, an inexperienced but highly conscientious company founder was trying to get a handle on the state of the company's projects and figure out how to manage these teams day to day, to ensure they'd meet key customer dates.

The challenges

This type of technical work is very iterative, judging completion is very subjective, and the finished product involves many small detailed pieces of work that must be touched by multiple team members: e.g., video footage, sound files, graphics objects, software modules.

These teams are full of creative folks who may be very resistant to anything that looks like a "limiting process" or bureaucratic management- especially if that management takes up their precious time. (Status reporting? Yuck! Low-value! Go away!)

Where our story starts

The company was undertaking a critical project for a new customer; payments to the game company (and the ability to make payroll!) would depend upon hitting monthly milestones for certain deliverables. To the founder, it was clear some kind of management was needed to ensure these dates would be hit.

The founder had never managed a project himself; but he tried valiantly to pick up tools from reading, talking to colleagues, attending local professional association meetings, and going to project management seminars. He would then try tools out on the team: Gantt charts, status report formats, checklists, executive status review meetings, etc.

What happened

The *technical manager's* reaction was to withdraw in the face of all this management flurry. He told the founder to stay away from his team; responded grudgingly to requests for updated schedules; and finally provided them with the caveat that they didn't mean anything because their work was too hard to estimate and express accurately in a project management tool.

The *team members* liked the founder but didn't get all his tries at various management techniques; they didn't understand what he was trying to accomplish. The overall feeling among these young team members was that he was trying to micromanage details in an ineffective manner, and do "management" just because some gray-hair had told him he was supposed to.

The *founder* was no dummy and not insensitive either. He didn't like the way any of this felt or worked, and he wasn't getting what he needed to manage the project with the customer and make tradeoff decisions that would allow them to hold their schedule commitment.

How the issues were resolved

- **Management by walking around:** The founder threw out the written status reports and started doing management by walking around. He got closer to team members, and understood the technical issues most affecting the project by talking to them briefly over breaks.
- **Value-add PM- facilitating better work among groups:** He started understanding some disconnects between groups, and contributed by showing them how they could better integrate and streamline work among the art and programming groups.
- **Milestone reporting:** He used the Gantt charts to communicate with the publisher for a contract re-negotiation, but then moved to milestone reporting to them, freeing him from a mandatory weekly update to a certain form of schedule file to please the customer.
- **Iterative development with time-boxing:** He reworked the schedule approach to be iterative, reflecting how the team was actually working and thus getting better buy-in to the information. He experimented with simple charts that allowed team members to quickly estimate schedule iterations on the fly, so that feature prioritization and time-boxing (scheduling an iteration of development and doing feature development in priority order until the time period is over) could be used to handle the schedule uncertainty.
- **Fast and easy status logging:** At the suggestion of one of the technical team members, he posted wallboards that let artists and developers quickly mark their progress on each "room" of the game as they worked, and see where others were as well. He was able to use the wallboard status to see overall trends and holes in progress, without bugging people for separate written status reports and having to compile the data.
- **A different kind of "management tool":** Finally, he realized that one of their most valuable management tools would be an asset management system that kept track of all the pieces- the video clips, sound files, artwork, etc., and automatically tallied completion status as people checked in their finished art and modules. He found the resources to get a simple starter system into place quickly.

In the end.....

The founder was able to come up with project management tools that were an appropriate fit for his technical environment. To do so, he just needed to

- not automatically accept anyone else's rules about "how it has to be done"
- listen well to his team and value their perspective, understanding what would help them, and what would get in their way
- get clear about what value he and they needed to get from any management tools they used,
- and be willing to experiment to find the approach that worked best for his company.