## Assembling a project team

Right up front many project managers (PM) find that they have a balancing act when it comes to staffing up the project. First as the project's scope is being put together the project's staff will be small. The PM, maybe an assistant or two, generally a chief engineer, and maybe a couple more engineers. Included in the project circle will be one or more stakeholders, such as product managers, sales, and marketing, production, and as mentioned maybe outside vendors and end users. Many project managers will take on staff offered by management before they are really needed because if the project is late it looks better politically than not to have done so. It is not always safe to stick with the lean and mean approach.

Conversely many small startups may not agree with taking on more staff than needed, as you have only so many dollars between infusions by the investors. Yes, it's true that if you don't hire enough staff and don't get the job done because of it you're at fault. It's a balancing act.

An early task that the PM should do is direct his team to incorporate design, processes, procedures if applicable that have been successful in the past. We like to think that we are in the middle of new technology. Only the researchers are there, the rest of us are appliers of their work. The methodology used on a new project tends to develop based on the training given and the tools at hand, and what others have done in the same situation. Very hard to develop new methodology and then tool up. A methodology is best developed by doing and then documenting. A good PM will know their companies' best practices and encourage his team to use them.

Most technical managers have very little management training. So, most of them are schooled in how to get the job done and not how to manage the job. PMs might be the exception to this although you'll see many engineers that move into the PM track. While the non-tech PMs will have to still be versed in the technical lingo of the industry, they often don't need to know the nuts and bolts. But good non-tech ones will need to rely on their lead or chief engineer for the project to sort out truth from fiction when it comes to technical matters. A major PM job is to track the finances of the project. So, a PM with a business background has a better chance of excelling here than a tech based PM.

An important part of the PMs job is to run interference or carry water for his team. Companies with large bureaucracies generally give little job satisfaction to their workers. Most know of Parkinson's law – work expands to fill the time allocated. Companies that give their workers satisfaction tend not to suffer from Parkinson's. In such companies' workers who loaf or don't seem to care most likely are overwhelmed and might not be up to the assigned task. A well-run company will have team members unhappy with that person, and not only management. Companies can also invoke Parkinson's law, increasing bureaucratic procedures to fill the available time. PMs need to filter as much as that out from their team as possible to keep the team focused on the goal and not checking boxes.

Other constraints placed on a team can be lack of trust in the team that puts the management in a defensive posture. There must be freedom of action. People want to accept responsibility, but they won't unless given acceptable degrees of freedom to control their own success.

The authors worked at a company where a new President started a review of all employees every six months. Those who are in the lowest 10% were let go. It might seem a way to keep the good people and remove under preforming employees, but it destroyed morale by making everyone a competitor. This caused employees to not want to help other employees since that could raise them above you and you might be out!

## Initial Considerations

Once we have a team, a defined scope, and a timeline what's next? The team should divide each issue or problem into as many parts as possible. Begin with those issues or parts of a problem that are simplest and easiest to understand. Often the project that has an unreal target date is the one that needs time to brainstorm and not just get the work done. We spend far too little time planning, investigating new methods, training, reading books, etc. Often the work at hand must be stepped back from and questioned as to if the work should be done at all.

Again, teams are formed around goals. At the start individuals might have separate goals. A good team will be bound around the same goal. Often the goal turns out to be a successful team. Often the company's or client's goals are not paramount to members of the team. Bound teams will have inside jokes, and common experiences, space they consider their own. They might even give themselves a name and have shirts and hats. But inadvertently teams formed this way end up helping the client and the company.

Outsiders might view bonded teams as cliques. They may come across as cocky and self-sufficient, irritating, and exclusive. The chemistry within a team can become an end in itself. Sometimes a team will adopt a higher standard than necessary to set themselves apart. If possible, a team should be kept together from one project to the next if they are successful. It starts the next project with enormous momentum.

In a good team the resultant whole is greater than the individual parts. That said every successful project needs a catalyst person. Often that persons' necessity is hard to discern but they bind the team. It is often the PM, or the lead engineer. But not necessarily. Especially if the PM and lead are spread out on other projects. This fragmentation of work on other projects means now they have more interactions to track, and one or more members of the team must become the project "mother."

A common thing that can make teams not bind is the physical separation of members of the team. With the advent of working from home this can make team building harder as there is no casual interaction, no team culture forming. Also, when team members are neighbors, they tend to all go into creative mode at the same time. Sometimes the best thing a group can do is to get itself isolated from the rest of the company.

In the book Peopleware it is stated that in software development there is a 10 to 1 spread among speed in programming between individuals and companies. Good performers tend to end up together while worst ones end up together also. So, is it that companies tend to attract good, or is the company making good programmers bad?

But in looking at that last paragraph would a project consisting only of the fastest programmers or hardware design engineers be ideal? Electronic technicians used to far outnumber engineers. Each engineer often had one or more technicians assigned to them. Now the technician profession is rapidly contracting. Engineers used to design, and technicians would troubleshoot and bring the engineers designs to life. Occasionally an engineer would have to get involved in debugging aboard.

The authors have seen many engineers that were great at design but couldn’t master the skill of troubleshooting their own circuitry. Also, engineers who could design individual circuits but not able to tie them into an overall system. We also once saw a team of senior engineering students, all with 3.0 or better GPAs struggle to turn the subsystems each had designed for a robotic vehicle contest into a working system. It took the student with the lowest GPA among them to lead them into turning the parts into a system. So, diversity often is the answer.