

The Plug and Play Staffing Syndrome (PPSS)

By Doug Tracy, Fairhaven Solutions, LLC

What is the Plug and Play Staffing Syndrome?

Success as a project manager (PM) has always been dependent upon the ability to manage change, minimize risks and control costs. We live in a dizzying world of rapidly evolving technology that challenges us to cope with a rate of change that never seems to stop accelerating, constantly forcing us to respond by adjusting the way we conduct both our private and professional lives.

In the world of project management, these technology changes have extended beyond simply hardware and software, and now heavily influence the way employees and employers interact, their expectations, their attitudes and, ultimately, their levels of commitment. Both employees and employers are keenly aware that the company they work for may not exist next year, that economic conditions can force job reductions at any moment and that the most skilled developers will more than likely have an array of better job offers to choose from at any given moment.

Not surprisingly, both employees and employers have adjusted their expectations, their conduct and their career plans accordingly. Many developers now view careers as a series of assignments that may last one to five years, always ready to jump to the next company as soon as they gain sufficient experience to merit a better offer. Many employers have corresponding expectations, i.e., employers do not expect to retain employees for extended periods.

As a result of the generally short-term duration of developers, project managers are confronted with the problem of the "Plug and Play Staffing Syndrome" (PPSS), the phenomenon of viewing developers as interchangeable components on a project team. Project Managers are painfully aware that replacing developers on an active project can be difficult and costly.

You may or may not have experienced PPSS on your project team or in your environment. However, in any case, the concept of planning for the possibility of the loss of your most experienced and valuable developer is good insurance for any PM on any project team. The purpose of this white paper is to discuss this Plug and Play Staffing Syndrome, and to suggest some ways to deal with it.

Where did PPSS come from?

We seem to live in a plug and play world these days, especially when it comes to technology. If you want to upgrade your PC, you simply plug in the new component. If your car breaks down, the faulty mechanism is rarely repaired – it is simply replaced with a new one. Although this works well for mechanical or electronic devices, it becomes problematic and costly when it comes to replacing developers on a project.

PPSS is the problematic concept that a developer on a project team can be readily replaced at any point by another with similar technical skills - with no impact on project schedules or costs. This stems from the thinking that technical skills are all that are required to get the job done, much like any plumber can fix any sink. Project teams are often comprised of developers on loan from other departments,

developers committed to more than one project simultaneously and consultants from external companies. PM's often do not totally control the assignment, duration or priorities of resources. Selections and assignments are often handled by upper management, the human resources department, recruiters or even managers in other departments. Developers may be moved from and to the project based on the thinking that, as long as a Java developer replaces a Java developer, the change will be invisible to the project. This is the essence of PPSS.

Project managers know well that this is rarely the case and are very aware that replacing a developer is risky and often costly. It is critically important for PM's to recognize these risks and costs, anticipate staff changes and plan accordingly in order to ensure both short term and long-term project successes.

The impact of PPSS

What happens when a seasoned developer leaves and is replaced before the project is finished or, following completion of the project, does not stay with the team to apply his skills to the next project? What is the cost to the project team, to the company and to the customer? The answer depends on many factors, including the size of the team, the skills of the remaining developers and the current status of the project.

In the case of a developer leaving before a project is completed, the following can be expected to occur, with varying costs:

- **A replacement must be recruited.** The time spent reviewing resumes, interviewing candidates, negotiating hiring terms and arranging a start date can be extensive and is rarely figured into the project cost equation.
- **The new developer will face a learning curve.** A considerable amount of time must be invested researching and understanding the components of the system, digesting the requirements, getting comfortable with the team and learning new standards. Additionally, there are new tools to be mastered and the new developer must spend time getting acquainted with the office environment in general.
- **Team peers and the PM will invest time and effort orienting the new developer, answering questions, reviewing work and evaluating performance.** These are often the hidden costs of bringing on a new developer, but they can still be significant. Team peers, as well as the PM, will invest time orienting the new developer, which means less time to spend on their own assignments. The capabilities of the new developer are an uncertainty until proven, so the PM will spend time reviewing and evaluating the new developer as they get up to speed. There is also a risk that the new developer will not be able to perform adequately, which usually is not discovered until the new developer has been on the job for a number of weeks.
- **The business analyst will spend time communicating the requirements to the replacement.** The business analyst is generally the one who knows the customer's business, interacts with the customer and serves as the link between the customer and the project. Most importantly, the business analyst is usually the author of the requirements, which means the

business analyst will necessarily invest time working with the new developer to ensure that the requirements are well understood.

- **The project schedule may change.** Unless it is a very large project, it is inevitable that the impact of bringing on a new developer will change the project schedule to some degree. If the requirements are highly detailed and locked in, the new developer may become productive quickly. However, this is not always the case, so any time spent getting up to speed will be added to the schedule.
- **Project costs may increase.** Many costs, such as the learning curve, are obvious. Other costs, such as the time spent by peers orienting the new developer and the impact on quality, are not.
- **Team morale may be affected.** A new developer always brings a new attitude and personality to the project. Peers who find that they are spending time away from their own assignments to explain things to the new developer may not be pleased with the situation. Styles and attitudes may conflict.
- **The customer may have concerns.** In many smaller projects, customers have a degree of direct interaction with developers on the project team and over time develop a sense of confidence and trust as they get to know the individuals on the team and their capabilities. Seeing the make-up of the team change frequently may be unsettling for the customer, particularly when these changes impact project schedules and costs. The customer may also be concerned that subsequent maintenance of the system once it is completed may be impacted if the original developers are not available to handle subsequent repairs efficiently.

In the broader perspective, replacing a developer, particularly a key developer on a small to medium size project after the project is completed can prove to be even more costly. The primary reason is that a great deal of knowledge is amassed on any project, which can be subsequently applied to reduce costs, increase profits and improve quality on similar projects.

As an example, if the first \$200,000 fixed-price project is completed in 2000 hours at \$100 per hour, completing the next similar \$200,000 fixed price project in half the time (1000 hours) would result in an additional profit of \$100,000. Additionally, the subsequent project would inevitably benefit from improvements in the techniques and approaches used in the first. The improved efficiency from project to project also has secondary benefits, in that the company will be in a better competitive position to seek contracts by virtue of being able to offer reduced prices to customers, minimizing risks and ensuring profits.

What can a PM do to minimize the impact of PPSS?

In an ideal world, a project team is comprised of talented and experienced developers who share a commitment to quality, have proven track records, have worked on many projects as a team and can be depended on to be thorough and efficient. In my world, and probably yours, this is not always the norm. The reality is that developers come and go for a variety of reasons, and, according to Murphy's Law, usually leave at the most inopportune time. Replacements, despite great resumes and references, are always an unknown quantity until they have had

an opportunity to get their feet wet, overcome the learning curve, blend with the team and demonstrate their capabilities. Inevitably, a new developer brings a different style, attitude, work ethic and skill set to the team, which requires adjustments on the part of peers as well as the PM. Given the inevitability of changing resources, PM's can anticipate and minimize the impact of PPSS in a variety of ways, depending on the environment, level of management support, schedules and budget:

- **Require comprehensive documentation.** All PM's know the value of documentation and it is always a challenge to ensure that it is sufficiently detailed and always current. Detailed documentation can minimize the learning curve and, just as importantly, reduce the amount of time that team peers spend helping the new developer get up to speed.
- **Require adherence to standards.** How often, after a developer has departed, have you spent time trying to figure out where the developer stored various software modules and how they work? Adhering to standards greatly enhances the ability of a new team member to step in and be productive quickly.
- **Conduct frequent knowledge-sharing sessions with team.** Small teams tend to interact easily. Large teams often do not. Formal sessions with the team focusing on key design and coding issues can prove invaluable when it comes time for a new developer, or even a team peer, to step in. Knowledge that resides only with one person is always a high risk.
- **Pair team members.** An ideal approach is to pair a junior with a senior team member whenever possible on assignments. This is good insurance in that it provides a built-in backup, in the absence of a team member. Additionally, it provides an opportunity for the junior member to gain the experience and knowledge to become a senior member on the next project.
- **Require comments in code and conduct code reviews.** Some developers write profuse comments in their code. Some write none. To the programmer who has authored the code and thoroughly understands it, comments may not seem important. However, to subsequent developers new to the project, comments can be invaluable in minimizing time and effort.
- **Proactively communicate with and educate decision-makers.** In many cases, staffing decisions are made by upper management who often have a perspective on the business and priorities that differs from that of the PM. The costs of PPSS may not be immediately apparent to upper management as they contemplate their staffing decisions, so proactively communicating concerns and issues to upper management regarding staffing issues can be of great benefit, especially if PPSS impacts are presented in the form of costs to the project and to the company. Ask for upper management for staffing commitments for the duration of the project.
- **Keep communication channels with team members flowing.** Although not always possible, knowing that a member of the project team is seriously considering leaving the team or the company can allow the PM time to make arrangements to minimize the impact, such as varying assignments and pairing team members to ensure knowledge transition.

- **Allow for the impact of potential staff changes in the Project Plan.** How many project plans have a line item allowing for the cost of staff changes? My guess is very few. However, building in a management reserve can provide insurance in the event a replacement occurs, as well as provide an additional profit margin should no replacement happen.

Conclusion

If I could have just one wish as a PM, it would be to have an experienced, proven and dedicated team stay together long enough to complete a dozen significant projects. Is that a lot to ask for? Maybe so. In any event, I know that it never gets any easier for PM's as technology, requirements, customers, attitudes and values change. Whether it's PPSS or any other potential project hazard, the trick is to always be alert, try to anticipate what's next and always have a 'Plan B' ready.