

How to Successfully Deploy Productivity Software Across Teams

Part 1 of 4: Software, Change and Missionaries

In twenty-five words or less, what would you define as the critical success factors for rolling out project and performance [management software](#) across a workgroup or team? The results can be truly gigantic. Teams and organizations do it everyday you were to start today, what would be on your list for doing what works, and avoiding what doesn't?

In this white paper we would like to cover the basics for what you need to know to be a success at deploying software across teams. By the way, if you aren't sure how to answer the questions above, maybe even slightly uncomfortable at the prospect, you are not alone. Rolling out a [project management](#) or groupware solution is enough to make all of us in the position of management cringe at some time or another. It starts the voices in the back of the head. They are quite persistent in their warning that there is trouble ahead. The voices say that to launch on such a venture is to:

- Enlist direct reports and peers in learning "one more" software package amidst their protests that they are already swamped, and frankly don't need or want another tool,
- Incur high frustration, something akin to "herding cats" or "pushing a wet noodle up a hill,"
- Expose oneself to public challenge, defiance and/or defeat,
- Fail to generate the outcomes expected, promised, or hoped for,
- Engage in risks such as spending money and taking on resistance, and for what exactly?

In fact introducing new software to be applied across a team or group looks a lot like engaging in a change management effort amidst numerous risks. It is. Employing new software tools, especially as part of a step forward in performance and productivity, requires a change in the way people work. Let me say it again more directly, **it changes the way people work**. For many people in management, the software rollout (change) process means embarking with the expectation that "things are going to be different." And as we will see, often without having clearly defined and worked through with the team what exactly is going to be different and why.

At Performance Solutions Technology we have accumulated several years of research on what does and does not work in the practice of rolling out high performance software tools, such as our flagship product [ManagePro](#). Our results consistently point to three knowledge areas or best practices that highly affect the outcome and are key to your success:

1. Shed common illusions about the process of performance improvement and replace with four key reality concepts,
2. Use "Pull – Push" leadership actions,
3. Proactively help team members climb the "Use of Tools" ladder.

This article is a four-part review of our findings and recommendations in this area. But before we continue, let's be clear about what to avoid in implementing software, or as we like to describe it,

The course of the hapless missionary

The typical process of implementation often starts with a technology adopter or visionary who becomes a "missionary" (for use of a software tool). By that we mean someone who decides that a software package is a valuable solution for others. Sometimes they arrive at this decision point out of personal experience; sometimes they arrive at this conclusion based upon what they think someone else on their team needs, e.g. "You really need to be more organized."

Having uncovered value in the software, they determine that it would be beneficial to have others, if not everyone, share in their discovery, and experience the benefits of using this software tool. They begin the process of attempting to convert others. They may or may not encounter interest, but they always encounter some form of opposition, passively if not actively. Everyone is not convinced they should convert to the new "betterment" tool.

Part of the missionary approach to tools, is the belief that others would share in their appreciation or value or benefit from this tool...if they would "just try it." However sharing their perspective and tools, even if done enthusiastically, is usually not enough to convert the team or group to a new tool. Typical results yield only a couple of converts within a workgroup of six to twelve, with regular usage by others limited to a cursory review and/or try it approach (drive it around the block and then back to the lot). This is further anchored or underscored by the belief position of other team members that:

1. "I'm too busy to learn a new tool," and
2. "I'm doing basically OK with the (preferred) tools I presently know how to use," e.g. so why do I need to change.

Encountering this type of reaction, an effort that started out with such zeal easily becomes thwarted. Our findings indicate that "missionaries" are eaten by the "cannibals" (give up on the software conversion due to resistant staff) within six months, if not supported by insistence from a person or position of power. What do we mean by that? Despite the lofty commitment to values of increased performance, productivity or simply reduced work effort, the average implementation across a group drifts into what might be called an aspiration. An aspiration to improve that is not realized, and is only moderately enhanced by a command from above. From inspiration to aspiration... without a successful deliverable turns out to be bad business. It's not just an unsuccessful attempt that generates little return on investment; it can leave individuals frustrated and the workgroup less collaborative.

Purchasing and installing productivity software from this purview parallels what one might expect to be the life cycle of the average piece of home workout equipment. The average consumer starts out with gusto in the inspired phase. Shifts to an aspiration as usage declines, and then concludes in storage or a garage sale. From inspiration to aspiration to storage on the sidelines as one more goal unachieved, due to the requirements of more work, more effort, more discipline... more of something than was unavailable to resource the change in behavior.

The good news is that we have also identified not only the painful average lifecycle of (marginally successful) groupware implementation; we have also identified the patterns of what works in successful software introduction. Patterns that emerge in the three best practice areas mentioned above. Let's start with the first practice, "Shed illusions about performance improvement and replace with four key reality concepts" in the part two.

Part 2 of 4: Shed illusions about performance management and replace with four key reality concepts

There are several important things to "know" about the process of implementing software across a group. In plain English, there are ways to be smart about this, and then there are the other options. The interesting thing is that getting to the "knowing" part often means letting go of something currently "known" or expected. It seems that software roll-out parallels transitions in general, and William Bridge's work on the topic consistently finds that the first step is one of "letting go." The image comes to mind of not being able to grab something new until you can let go of what's currently in your hand. The great thing about what we are going to discuss is that letting go of these beliefs actually lightens the load, frees you up, and increases your chances at success. What follows are four key concepts and some of the opposing perspectives or partially true assumptions that are at the heart of most unsuccessful software implementations.

1. Myth: Most people use software because it improves their productivity.

Reality: No, most people aren't driven by a need to be more productive. Most people use software because of one or more of the following three reasons:

1. It's a required tool where they work, or
 2. Because it makes something easier, or
 3. Because it helps them avoid something unpleasant
- If you still aren't convinced, why do you think so many people still work from a paper and pencil and list mentality (whether it is an actual word product or the electronic version)?

Bottom line: aspiring to high performance is not a universal motivation, whereas having things be easier or avoiding something unpleasant is. This is a difficult concept for many software missionaries to get, because their personal motivators are often quite different from the norm or at least a number of their workgroup members. Proceeding ahead as if others will be motivated as you are... is a position fraught with difficulty and usually erroneous assumptions. Don't do it! Be kind to yourself; plan on constructing your software rollout based upon the three real reasons for adopting new software, not the productivity myth. You will be happier and so will your team.

2. Myth: People will use new software valuably if you purchase it for them.

Reality: No, most people will use software if it solves a problem for them, relatively directly. In fact, if it solves their problems very directly and relatively easily, they may use the software even if you don't purchase it for them. They may even purchase it themselves – can you hear the concern in your IT department yet?

Bottom line: purchasing software, no matter how good it is, is not a necessary precursor for your staff using the software. It is a guaranteed cost factor however.

3. Myth: People will use new software that changes the way they work if you purchase it and train them. In fact, in no time they will figure out how to configure and apply it to solve the business problems it best addresses.

Reality: First of all, most of us derive only the most apparent solutions out of software, and configuring software to solve business problems is usually at least one or more abstractions beyond what normal usage looks like. So don't expect that to happen spontaneously.

Bottom Line: Rolling out new software is a change management process. People don't change very quickly most of the time, unless the timing is incredible... so be prepared for a process. That means plan this process out. That means make sure you have enough resources (time, momentum, meaning and power) to complete the journey.

John Kenneth Galbraith described some of the mental gymnastics we all go through in facing change and said, "Faced with the choice between changing one's mind and proving that there is no need to do so, almost everybody gets busy on the proof." If John had any insight in this matter, while you are preparing for a process, be prepared for people to push back in passive or active ways. This means accept the fact that 2/3's of the population doesn't particularly enjoy change (using new software), and will either not comply and/or certainly test you to see if you really mean it. Do you?

By the way, the software assimilation process we (at PST) expect is for:

1. Some to get inspired (1st step),
2. Many to labor or work at letting go of old tools and expending the energy and discipline to adopt the new ones (2nd step),
- 3) If escorted, most to reorganize their work efforts with significantly improved deliverables (3rd step).

4. Myth: Once people have started using a productivity-enhancing tool, they will continue. "Get 'em over the initial hump and its down-hill from there."

Reality: Higher performance, improved productivity, “raising the level of the game”... however you phrase it, is counter to the natural laws of entropy if you care to review history, recent or ancient. It is as “**produced**” as the ascent of hot air balloons, no pun intended. That means that higher performance is not self-sustaining most of the time. Great sports teams don’t occur without a coach and an adversary (motivation). Furthermore they have a difficult time repeating a really successful year. It certainly doesn’t look easier the year following.

Bottom Line: reaching higher performance through use of new software tools will not likely be self-sustaining, and will instead need ongoing SELLING, support, recognition and incentives. People might aspire or be interested in the change brought about by adopting new software and the associated work patterns, but change usually requires a serious dose of motivation and someone exercising leadership to keep pushing towards the *new destination* and selling it as a worthwhile objective. There’s actually a component of both push and pull to effective leadership. But that’s what we are going to look at in the next section.

So let’s summarize, in other words, “So what do you do with all of this content?”

1. Well first of all you avoid starting the process with denial or myths in place. That’s a very good thing, as they incur frustration and pain. And you don’t need reminding that everyone would like to avoid pain. Knowledge is awareness of the playing field. It is seeking to operate with a minimum of surprises and to make plans and decisions based upon the facts of reality, not preference, history or habit. Why? – because it’s easier that way.

2. Secondly realize you need to plan this process out and work a plan, given that there are obstacles in place that founder many eager but unaware software missionaries. Remember this is essentially invoking a change process. The planning process starts with an important set of leadership activities. That’s where we will pick up in the next section.

Part 3 of 4: Use “Pull - Push” Leadership Actions (A Requirement – not an option)

The kind of leadership needed to excel at implementing [project planning](#) and performance management software across a workgroup or team (at the top or the front line) is contained by a set of critical actions that fit into what we have identified as a “pull – push” leadership model.

The Pull

In our nomenclature, pulling represents creating an environment where your workgroup or team is drawn up to the next level. It is engaging in leadership behaviors that create a sense of motivation, an incentive, and a want to. It is essential to have this. Without it, you are left to depending upon command and control (or hope) to prompt tool usage - which incurs another whole set of problems.

Back to our basic motivation discussed earlier, one of the key motivators to using software is the wish to avoid the unpleasant and the anticipation of something better. Put another way the introduction and implementation of performance tools for most people needs to be clearly viewed as a solution. It needs to be a solution to a discomforting problem. Not a solution to a denied problem, a minimized problem, a not-me problem, someone else's problem, e.g. a solution for which there is a felt need.

Perhaps you sense where we are going with this, but we are describing a process of providing a solution which mandates that the workgroup has "felt" or identified the problem first! Pull – Push leadership starts with doing a very effective job of prompting and responding to this key question:

"Why do we have to use this software?"

We think of software implementation as climbing a long grade in a vehicle and that the process of uncovering problems is like picking up speed prior to starting the grade. It's the development of tension that makes a solution all the more of a release (welcomed and embraced)... not a burden.

Selling a solution for which there is no perceived problem or need is a tough sell. So the work in selling a solution starts with uncovering problems, to make them and their consequences more visible, palatable, and actuarial. It is consciousness raising for many groups. Sometimes it means rediscovering actual costs of current work practices, totaling costs of miscues, re-dos, poor hand-offs, limited coordination and resulting slipped start and due dates. Denial and the past both easily cover the costs of problems, and both need to be minimized in order to build momentum towards a solution. The world is a different place (the need for solutions is not perceived as compelling) when there aren't "barbarians at the gate."

One way to provide more motivation for the pull side of the equation is to make quite clear the types of problems that are inherent to the current tools or practices, tools and practices that make work regularly frustrating and more difficult than need be. Here are some common places to look:

1. The limits of improved self organization; working harder, solo efforts, and improved fire-fighting, e.g. managing via lists and email.
2. The limits of applying a scheduling/resource software tool to the management of team efforts, recognition of other's contribution and management in general.
3. The limits of utilizing tools that don't reinforce coordination or collaboration or documenting information for increased knowledge improvement.

The Push

High performance gained through technology adoption requires one or more people to lend their support, the power of their position, and ultimately their insistence when being tested, that "Yes, this is the tool we are going to use." This is a management push, by the way, and should not be presented as an IT or technology driven directive.

In a broad sense, all efforts to raise performance, productivity, effectiveness, efficiency, or however it is described - requires a push. It is all supported by Newton's Laws of Physics if you think about it. Here's a brief overview:

1st Law – An object maintains its state of rest or uniform motion unless it is acted upon by an external unbalanced force.

Implication: People will tend to keep on working the way they are used to unless impacted from the outside (ex. someone using pull-push behaviors).

2nd Law – The acceleration of an object varies directly as the external unbalanced force applied to it and inversely as its mass.

Implication: If you intend to create change, you had better bring a sense of force and mass (conviction, authority, power of reason, persuasion, enthusiasm) and a sense of speed or urgency.

3rd Law – For every action force there is an opposite and equal reaction force in the opposite direction on the object exerting the action force.

Implication: The effort to move people out of their customary work habits through deployment of new software tools will generate resistance to doing so. Bring enough "push" to work through, persist through and beyond, the reactionary resistance.

Pushing is just as necessary and just as important as pulling. You will never do such a good job of pulling, that you won't also have to do some pushing sooner or later, and usually sooner. Pushing at times will be reduced to confrontation and insistence, which besides being a necessary push back when encountering resistance to adoption, is also most effective when supported by an objective, and at least a philosophical paradigm. In other words don't make the (parental sounding) mistake of stating, "Just do it because I told you to," but instead as an example state, "We are going to use it because we cannot afford to continue working with our old tools, or we can't afford to generate the level of production, quality of work, level of profit... that we have without existing tools set and associated work practices.

If this all leaves you still confused or unconvinced, think of deploying productivity software as similar to hot air ballooning – no pun intended. You can have the nicest basket and induce people to get into the basket (buy-in), but at some point you will need to turn on the burner (and episodically after that), because lift requires input of energy.

Part 4 of 4:

Helping Others climb the "Use of Tools" Ladder

Implementing productivity or any type of [management software](#) across a group brings to bear a very interesting, powerful and ancient dynamic. The dynamic is the human reaction to new tools. History suggests that humans have had a repeatedly ambivalent reaction to adopting new tools unless at a point of great threat (loss of

life, loss of land, loss of income or loss of status). One could argue that we continue to work with tools that are easily 5,000 years old. We continue to rely upon and feel comfortable with tools that were around when the pyramids were built - memory, voice and written lists (and control).

We have identified four different steps around which people organize their use of tools – a four step “Use of Tools” ladder if you will. The rungs or steps embody a progressive use of (and implicit valuing) of tools that begin with value to self, and extend to organizational value. From solving personal preference for ease of use, (lack of discomfort) to meeting collaboration and organizational requirements. Let’s identify what each rung looks like and then conclude by discussing how you can use the information to enhance the success of your software deployment.

Four Levels of Managing Information with Tools

1st Rung

People on the first rung use tools for organizing themselves. The relative need for organization is driven by the level of frustration incurred when work is increased in difficulty because of information retrieval or access delays. At this rung people use tools based upon the personal or immediate level of benefit-to-work payoff. This means they use the easiest tool to provide the minimum levels of needed organization. Where easiest is partly a function of skill (writing vs. typing) and partly a function of past learning (it’s easiest to use something I already know how to do). Managing information (with tools) is not necessarily viewed as key to their effectiveness on the job, nor is reporting or collaborating with others viewed as a key prerequisite. Personal and organizational success is viewed as determined by individual competency, not organizational learning or sharing of information. At this stage a person sees success as defined by the ability to deliver the end product by whatever means they deem acceptable.

At the first rung, people would say yes to questions such as:

1. Do you feel organized enough with the tools you currently use?
2. Learning/using new tools would not make me significantly more effective or successful.
3. It’s the skills and experience I have that makes me effective, not how I manage or use information.

2nd Rung

On the second rung, people extend their use of tools beyond a direct personal benefit to include managing information to support processes (ex. Accounting) and one’s superiors. The basis for using tools on the second step is to provide reports or an account of what has transpired – often in the past. Information gathered across teams via tools is not seen as a necessary update to making decisions in the day-to-day experience. Here the emphasis is using tools to track information and generate report based upon their ability to minimize the effort/time to document transactions and create reports. Tools are used to automate and improve efficiency of a verbal/hand written report. Again people seek the minimum pain - maximum payoff. As an example, for many this means using a pre-set format so it doesn't have to be created each time a recurring report is needed. Tools at this level are perceived as a necessary part of

the job, but only secondarily of value to the worker ("If I complete the paperwork/form then I get paid." "Our business processes require completing this information." "Completing reports is part of the job function.") Information retrieval is seen as much more valuable to top level than to the person inputting the data.

At the second rung, people would say yes to questions such as:

1. Are the reports you generate and the time spent tracking and maintaining information more valuable to your boss than yourself?
2. Could you do your job just as well without all the information tracking and maintenance system your organization requires?
3. Do you get most of your key information via phone, email and face-to-face conversations?

3rd Rung

On the 3rd rung, tools are equated with information management, and are seen as key to supporting decision-making and the bottom line. Tools are used for reducing the risk of uncertainty undertaken by the complexity and size of the projects, and the inherent risk in coordinating complex tasks. The need is to leverage the value of information/knowledge... for the purpose of collaborating across a team or work group, ultimately for the purpose of being more successful, more effective in delivering work objectives. Here the tool is engaged as a needed requirement for decision making, where one could not possibly track all the variables and details with simple organization or reporting tools.

At the third rung, people would say yes to statements such as:

1. Managing our information really helps us stay on track and not have details slip through the cracks.
2. If we don't track and manage information, we don't work well as a team.
3. Tracking and managing our information is critical to setting priorities and staying within budget.

4th Rung

At the fourth or top rung, tools are additionally used for the information management required to support innovation and growth or strategic goals. The strategic nature of goals (going where an organization has not gone before) and/or the emphasis upon learning and growth as pre-requisites for success, require information and feedback for accurate decision-making, proactive identification and response to problems/opportunities. This stage embodies a fundamental shift in the organization or within organizational groups, away from working based upon function to working based upon a strategic or project plan.

At the fourth rung, people would say yes to statements such as:

1. Our ability to gather and respond to feedback, whether from employees, customers or vendors, is key to our success.
2. You can't know what you don't know, but learning rapidly and doing the right things in response to what gets revealed goes a long way toward securing our future.
3. Every plan in support of every objective as errors and inaccuracies, our mission is to uncover and correct them as soon as we can.

Given this information and typology, what implication does it have for successful software deployment across teams and work groups? At PST we believe that successful roll-outs:

1. Use awareness of the group member's level on the "Use of Tools" ladder to provide a framework for effective problem discovery and solution selling (Pull – Push) and initial training needs.
2. Only expect people to change or move up one rung at a time.
3. Tie the use of tool features into solving the general problems consistent with the user's rung or level of using tools.

Bottom Line: Beyond covering content essential to basic operation of the software tool, helping others assimilate new productivity software tools needs to be focused upon providing a transition for people up the tool ladder.

Some examples of emphasizing different feature sets at each level include:

Rung 1 - This includes when implementing our program, ManagePro, starting with the use of its robust calendar, a task list that cross references easily to goals and people, the ability to attach documents to tasks and goal... and the ability to synchronize all of it across common organization tools like Outlook and the Palm Pilot. All of this emphasizing the benefits of everything organized in one place.

Rung 2 – At this level connect cooperation with software roll-out into incentives; tying program usage into increased likelihood of success and financial incentives, and into improved tracking of accomplishments, and recognition for achievement.

Rung 3 – Here it means tying the use of the tool into a broader perspective of using project management tools to organize the majority of work projects. To clarify how all projects are enhanced by the development of requirements, identification of the customer and stakeholders, definition of an action plan or breakdown of steps to achieve, and the documentation of progress to-date.

Level 4 – At the top it means tying the use of the tool into a management system that supports high performance through development, communication and collaboration on robust goals, defined action plans and regular feedback loops to determine need for course correction.

In summary, there really are actions enable you to be a success at deploying software across a team, workgroup, department or organization, (and avoid being eaten by the cannibals). Here's our basic list of what you need to know and employ:

1) Successful software deployment doesn't work well with faulty expectations, so plan, resource and ground yourself in the following facts accordingly.

- People use software because it makes life easier, and/or avoids incurring problems.
- Purchasing software, even with training, does not equate to it being adopted.
- Approach deploying software as a change management process.
- Reaching higher performance through use of new software tools will likely need ongoing selling, support, recognition and incentives.

2) Practice an active "Pull - Push" form of leadership with urgency and impact.

- Sell software as a solution to an identified (demonstrated, acknowledged) problem.
- Everyone needs to understand the basis for the answer to the question, "Why do we have to use this software."
- Remember Newton's laws; objects and people don't change without being impacted by an outside force with mass – use your early adopters, conviction, vision and problem-solution connection as your mass. And remember to expect and work through the push back.

3) Understand that any group will have some dispersion across a four rung "use of tools" ladder, and will need assistance to move up one step at a time.

- 1st Rung – Using software for the benefits of personal organization.
- 2nd Rung – Using software to increase ease of complying with reporting requirements.
- 3rd Rung – Using software to manage and coordinate the complicated and complex.
- 4th Rung – Using software to manage growth, creativity and the unknown.

The author of this series, Rodney Brim, is CEO of Performance Solutions Technology (PST). PST develops and assists organizations in deploying performance management software solutions and project management software, and presents these guidelines based upon our work with 1,000's of companies to help ensure your success in the pursuit of performance management. Performance Solutions Technology is found on the web at <http://www.ManagePro.com>