From MBWA to LBWA

21st Century People Solutions for Software Problems

Jonathan Powell, CACI

Abstract: People solutions to software problems is a misnomer. In the final analysis, people are the only solution to software problems. A qualified team built on trust and engagement optimizes organizational productivity and can solve any issue with software.

Introduction

If the DoD is to remain the wellspring of U.S. technological innovation in the 21st century, it must develop and institutionalize new and different approaches to people management.

The changes required run the gamut, from recruiting to day-to-day working conditions. Some of these areas have started to be addressed widely (examples include streamlining the recruiting process and expanding telecommuting). This article includes advice on some of the areas already examined, as well as an area that hasn't received as much attention, but is as vital (if not more so) to maximizing the organization's productivity and opportunities for innovation.

We have not evolved to the point where software is capable of designing, building and deploying itself. Nor have we reached the point where, once deployed, the software can configure, maintain, and enhance itself. Sure, certain advancements have been made. Automation is now found throughout the software development lifecycle, from requirements through operations and maintenance. This includes requirements gathering and modeling software through "self healing" systems and artificial intelligence.

However, as much as software and software development have evolved, it still comes down to people who design and build the software, and then intercede and fix the software when it does not work or behave the way it is supposed to.

Today when the Navy needs next generation software for its submarine sonar systems, defense contractors are not deploying hordes of automatons to the Pentagon to gather requirements, design the software, build the prototype, make it ready for production, and then support it through operations and maintenance. It still comes down to people, and I submit it will come down to people for a long time to come. Even in the far out future, as parts of this chain are automated, people will be needed to intercede, because software is not perfect, and problems always arise.

So if one's ability to overcome software challenges fundamentally comes down to people, the question becomes, "How do I get the most out of my people?" This will be the focus of this article.

Hiring

Since people are the key to overcoming software challenges, this essential topic needs to be addressed on the front end. Everyone understands the basics of hiring–what skill sets do I need and for what level of personnel (junior, mid-level, senior). What so many teams fail to do is institute rigor with respect to a person's "fit." All corporations have a culture, and within that umbrella one can find variations on different software development teams. As program manager for a large, complex, custom software development effort for the U.S. Army, I have personally interviewed all developers brought on to the team. The program peaked at 60 people, so maintaining this policy was certainly challenging. However, I viewed it as vital, and I'm not alone. For decades Admiral Rickover personally interviewed all Junior Officers who entered the nuclear Navy. Jim Collins in the book *Good to Great* cautioned on the need to be slow to hire. Robert Townsend, former CEO of Avis wrote, "The important thing about hiring is the chemistry or vibrations between boss and candidate: good, bad, or not there at all" [1].

Fit is important because it helps minimize transition costs. Hiring is costly. These costs are exacerbated if someone is brought in who requires extra training, or doesn't follow the team's behavioral norms, therefore negatively impacting productivity. Worst case, the person cannot adjust and has to be removed.

Best case, the person is able to adjust to function seamlessly within the team. And this best case is exceedingly rare, because for experienced professionals, change does not come easily as one's work patterns have become set over a number of years.

Let me give you an example. We had a developer who did not fit. One of his major issues was not being able to collaborate effectively with his fellow developers, which severely impeded his progress because it is a complex object oriented system. Since he failed to collaborate with his peers, his code was frequently rejected by the Test Team. Here he failed in another major way–instead of working with the Test Team to understand their reasons for rejection and establishing a way forward, he would simply adjust the software to his liking, and toss it back over the wall to Test, where it was invariably rejected again. By the time the issues with this developer were escalated to my level, it was clear he was set in his behavior and was not going to change. We put him on a performance improvement plan in accordance with company policy and had to dismiss him when he did not improve as the plan stipulated.

We have implemented some best practices to help avoid this situation. First is utilizing an interview panel, to include multiple developers. A variety of perspectives are important to help ensure we're bringing on someone with the right fit. Second, we include members from multiple teams where possible. So, for a developer interview, we would include interviewers not just from the team he or she is interviewing for, but also a developer from another development team on the program. If possible, we would also include a tester in the interview process. Communication and collaboration intra-team and across teams is vital for our program's success, and we want to bring in people who can function well in this environment.

This process is not perfect and certainly is more costly, so the reader needs to weigh the costs-benefits as it pertains to his or her program, but for large complex programs like ours, the up-front investment is more than outweighed by the costs and risks introduced by bringing in the wrong type of person. If you take shortcuts with your hiring, and do not ensure as a first step you are getting people who will fit in with your group and productively work with the team you have in place, you will fundamentally undercut your ability to respond to the challenges you will invariably encounter during the software development lifecycle.

Work Environment

If people are the solution to software problems, and we have selected candidates who we believe can contribute to the program's success and succeed, the question then becomes how do we get the most from our people? Here the answer starts and ends with environment. How your team performs today and responds to the myriad software development challenges that emerge along the way all depends on the work climate you establish and foster in your organization. "Provide the climate and proper nourishment and let the people grow themselves. They'll amaze you" [2].

Ask yourself some questions. Is everyone made to perform the same way in your program? Or are roles tailored to the particular strengths of each person? Are you fitting people into your organization or are you building team cohesion around the individual strengths of its members? Townsend wrote, "Why spend all that money and time on selection of people when the people you have got are breaking down from underuse. Get to know your people–what they do well, what they enjoy doing, what their weaknesses and strengths are, and what they want and need to get from their job. Then try to create an organization around your people, not jam your people into those organization-chart rectangles … You cannot motivate people. That door is locked from the inside. You *can* create a climate in which most of your people will motivate themselves to help the company reach its objectives" [3].

What Silicon Valley has and continues to accomplish is a testament to the soundness of this thinking. In the '80s and '90s it achieved notoriety for its unconventional work practices–casual clothing, foosball tables and other amenities, flextime according to personal needs, etc. Of course this notoriety was eclipsed by the operating results and history-changing evolutions that came from the Valley, innovation that continues apace to this day.

You might be thinking, "All of that sounds great Jonathan, but how do I make this work in the world of DoD software development?" The answer is remarkably simple and it starts with walking around. That is right–getting out of your office or cubicle and personally interacting with your developers, engineers, analysts, and other team members. While, as Watts S. Humphrey stated, one can no longer use Management By Walking Around for knowledge work, he or she can and must use what I call Leadership By Walking Around (LBWA).

LBWA and Other Keys to Success

LBWA is the perfect way to get to know your people and observe how they are functioning. It also shows as a manager you are interested and care, which will inspire and motivate those around you, as well as foster an environment of open communications, critical to any project's success. Walking around is one of the simplest, yet most vital tools in providing effective leadership, not just because of all the things you learn about your teammates and the open communications you inspire, but most important of all, because of the example you set. J. Paul Getty put it best when he wrote, "No psychological weapon is more potent than example. An executive who seeks to achieve results through the people who work under his direction must himself demonstrate at least as high a standard of performance as he hopes to get from his subordinates" [4].

Furthermore, LBWA helps build trust, and trust is the oil that makes an organization truly hum as a well-oiled machine. "Trust is what motivates people to follow our leadership, whether at work or home. And trust must be earned" [5]. And what better way to earn trust than by visiting people in their space and truly listening and interacting with them on a personal level? If it works for world leaders and diplomats for affairs of state, cer-

tainly it should work at the project level with day-to-day software matters. The trust you will build will help unlock individual motivation for the mission and increase engagement by the team. And as the CEO and President for the Partnership for Public Service stated "Effective Leadership is the No. 1 most important issue for employee engagement."

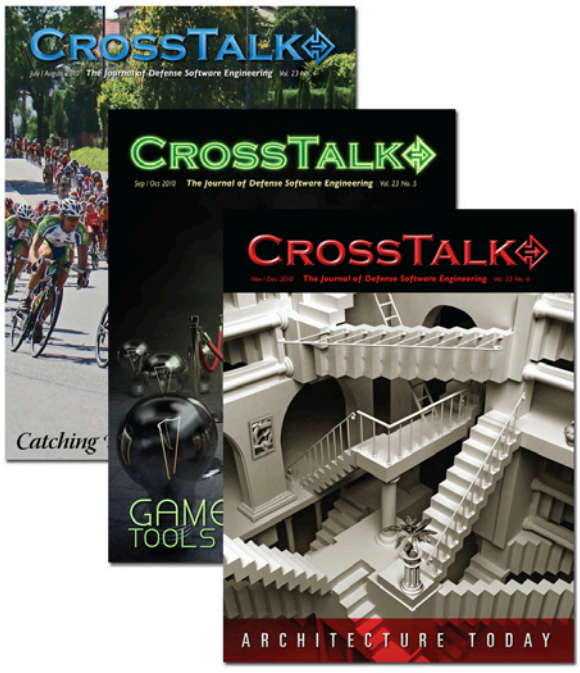
LBWA is not the only tool in your toolkit. However, as the leader, it is incumbent upon you to start with this fundamental–not aimless wandering around mind you, but prudent, targeted engagement.

Beyond LBWA, there are a number of widely known tools that will help improve one's ability to lead effectively. These include open-door management policies, conducting all hands meetings, e-mail messages directly from senior leaders, brown bag lunches, and hosting off-site sessions. These items are all very much relevant and useful within the DoD software development context, even more so because these are cost effective and simple to employ, and well within established military cultural norms. Recently I've successfully employed all of these tactics with a large Army customer comprised of many varied and dispersed stakeholders. Moreover, I've been able to leverage the military's Integrated Program Team framework to increase the effectiveness of these tools. For example, on the eve of entering a particularly intense high operational tempo period, a senior Army proponent was kind enough to help me lead an all hands session where we jointly presented to the 55-person contractor team. He and his staff have also written e-mail messages of commendation to program team members. These sorts of efforts tend to have an exponential reinforcing impact on morale and engagement when appropriately used to supplement company-internal communications.

Other ways exist to further leadership effectiveness and thus increase employee engagement, which in turn increases a program's ability to tackle and overcome software challenges. Some of these methods may require more creativity than others, but they are all still relevant and readily applicable within the DoD space. For example, tailoring work hours to the time of day when an employee is personally most effective. It is widely accepted that people have different biorhythms and perform better and are therefore more productive during certain parts of the day. Your program should try to support these individual needs to the extent practicable. This notion is a lot more accepted today in the DoD than it used to be. Work practices have evolved to include considering how we cater to the differing needs of three workforce generations–Baby Boomers and Generations X and Y, respectively. In particular, when it comes to attracting and retaining younger talent, implementing more progressive policies than those seen in decades past are imperative if the DoD is going to thrive in the 21st century. Certainly DoD cannot hope to be the wellspring of creativity in this era if the environment is a disincentive, especially with young men and women now having career options to choose from around the world.

Increasingly the DoD has moved away from contractually mandated work hours to performance-based work. Still, corporations catering to this sector too frequently default to standard working hours instead of examining how they can best maximize the productivity of each individual in a way that optimizes the whole organization. Same thing with work location, and the increasing traction telecommuting is achieving. Historical boxes and walls of work location, hours, etc. are being shunned in favor of management theory seeking to provide the employee with the best fit possible, instead of vice versa, thus enabling him or her to provide the organization with his or her best work.

The DoD is a large organization you say, so what can I do then if I work in a part of this sector where flexibility around work practices has not caught on? I would encourage you to look again. Even within the strictest confines of traditional DoD, I believe as a leader you have both an opportunity and an obligation to be creative for your people. Take a look–you will be surprised by what you find.



CALL FOR ARTICLES

If your experience or research has produced information that could be useful to others, CrossTalk can get the word out. We are specifically looking for articles on software-related topics to supplement upcoming theme issues. Below is the submittal schedule for three areas of emphasis we are looking for:

Software's Greatest Hits and Misses
November/December 2011
Submission Deadline: June 10, 2011

High Maturity - The Payoff
January/February 2012
Submission Deadline: Aug 10, 2011

Securing a Wireless World
March/April 2012
Submission Deadline: Oct 10, 2011

Please follow the Author Guidelines for CrossTalk, available on the Internet at <www.crosstalkonline.org/submission-guidelines>. We accept article submissions on software-related topics at any time, along with Letters to the Editor and BackTalk. To see a list of themes for upcoming issues or to learn more about the types of articles we're looking for visit <www.crosstalkonline.org/theme-calendar>.

For example, many services-based contracts adhere to Labor Categories (LABCATs), including formal job descriptions. Robert Townsend called job descriptions, "Strait jackets … insane for jobs that pay $750 a week or more. Judgment jobs are constantly changing in nature and the good people should be allowed to use their jobs and see how good they are" [6].

Practically, we all understand this, as folks are constantly asked to assume expanded roles and/or assume collateral duties. Since we cannot completely rely on a job description as a gauge, as a first step, make sure you have your folks playing the right positions and if they are in the right position, ensure they are in the correct role. This means doing work beyond a crosswalk of a person's skills/experience/education to LABCAT requirements, but an in-depth qualitative check for real organizational fit. For example, some developers prefer bug fixes over operations and maintenance.

Others prefer building new capabilities. And you can take this analysis down to finer levels of detail, to ensure a fit is made that resonates and is optimal for the individual and organization.

How about attire? The Army program I have cited permits personnel working outside of the Pentagon to dress "business comfortable." That does not mean folks are running around in tank tops and flip flops. In fact, there has not been a single incident of someone going over the line. If the mindset is to treat individuals as adults and with trust, the employees are more engaged, and you also increase your odds of attracting the best talent.

Unleashing the Organization

A good work environment fitted to support individuals and not the other way around, coupled with effective leadership, leads to engaged employees. And engaged employees are the key to unleashing the power of your team to solve problems.

Robert Zawaki observed in *Transforming the Mature Information Technology Organization* the following:

ED = RD X CD [7]

An Effective Decision (ED) is equal to the Right Decision (RD) multiplied by the Commitment to the Decision (CD). Employee engagement is central to both variables in the equation. Developing the right decision will require team involvement. Only by leveraging their collective wisdom, across functions, will you arrive at the RD. While some projects get this aspect right, often overlooked is how vital CD is to success. By involving the team in deriving RD, much of the heavy lifting in increasing CD is done. In executing a transparent collaborative decision-making process, you will be far ahead in your ability to get team members to commit to the best course of action, the RD. This is because they will have seen the logic used to arrive at the RD and how various perspectives were raised and used to shape the RD, including their own. Since both RD and CD are needed for a software solution to succeed, the effective leader will not omit the time and attention needed to maximize these.

Summary

In today's hyper competitive global environment, all organizations are focused on maximizing productivity. What is too often overlooked is the best way to achieve this. Bottom-up is the way to go–determining how the organization can be molded to unleash productivity on an individual basis, while maintaining a system to optimize the whole. Effective leadership is the key to shaping the organization to maximize employee productivity, because it spurs employee engagement. By maximizing the output of the entire team, and harnessing it for mission accomplishment, the effective leader can overcome any software problem through people. "The capacity of people to find answers, if they know it is worth the trouble, has never been tested to its practical limits" [8].

ABOUT THE AUTHOR

Jonathan W. Powell, CGFM, PMP is a Senior Program Manager with CACI. A former submarine officer, Mr. Powell has led complex engagements for military, federal, and intelligence agencies. His articles have been published in PM Network and Contract Management, respectively. Mr. Powell serves on the Board of the Montgomery County Revenue Authority, where he is a member of its finance committee. He holds a B.S. from USNA and an MBA from the University of Maryland.

Jonathan Powell
CACI
14370 Newbrook Drive
Chantilly, VA 20151
Phone: 703.258.4735
E-mail: jopowell@caci.com

REFERENCES

1. Townsend, Robert. Up the Organization. San Francisco, CA: Jossey-Bass, 2007. Page 96.
2. Townsend, Robert. Up the Organization. San Francisco, CA: Jossey-Bass, 2007. Page 96.
3. Townsend, Robert. Up the Organization. San Francisco, CA: Jossey-Bass, 2007. Page 95.
4. Getty, J. Paul. How to be Rich. New York, NY: Penguin Group, 1965. Page 80.
5. Farrar, Steve. Point Man. Frisco, TX: Multnomah Books, 1990. Page 153
6. Townsend, Robert. Up the Organization. San Francisco, CA: Jossey-Bass, 2007. Page 59
7. See <http://www.leadstrat.com/resources-strategic-equation-for-success.html>
8. Townsend, Robert. Up the Organization. San Francisco, CA: Jossey-Bass, 2007. Page 150