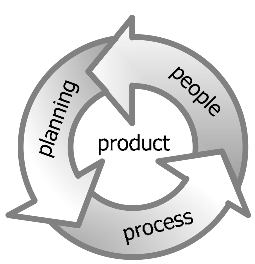
Meeting the Needs of Software Developers

Date:	Oct 7, 2010	
Title:	Meeting the Needs of Software Developers	
Code:	WMHN	
Abstract:	Using extreme project management tips and techniques will only be successful if you know how to effectively and consistently manage the human element. This is particularly difficult when dealing with creative types: graphics animators, web designers, and software developers. Although some of the best practices in this document can apply to anyone, I'll focus on software developers.	
Version:	1.0 (2009-08-01)	Original (excerpt from Leading Software Maniac's *Deliver Projects On Time, Every Time* seminar)
	1.1 (2010-10-03)	Completed after revision
	1.2 (2010-10-07)	Posted on LSM Web site (along with movie clip)

Overview

I believe that the key ingredient of software development leadership success can be represented as a balance of *planning*, *process*, and *people* leadership in order to produce quality *products*. The following diagram is a feeble attempt to show this relationship between the three leadership roles you must perform:



Successful projects require balancing planning, process, and people

When it comes to people leadership, there are a few simple things that you can do to reduce attrition or demotivation — both key risks that are under your control to mitigate.

Background

Most of us in leadership positions want to be sure that our teams are satisfied with the operation of their department and have a clear focus of what is expected of them to do. Since there are only so many hours in a workday, you may find that planning and process activities take a back seat to the constant stream of project and people−related activities including:

- Making sure that developers get the best assignments and that they are satisfied with their positions, work, and future prospects.
- Efficiently use creative techniques to find and hire the best talent.
- Frequently communicate both company and department information without burdening everyone's time with endless meetings and documentation.
- Ensure that decisions have the buy−in of the team. (And this doesn't necessarily mean *consensus*!)

It isn't enough just to lead and facilitate the software "nerd herd," is it? We may need to dig into the inner depths of what makes individuals and teams work and look at two simple mechanisms that you can put into practice immediately:

1. Addressing staff needs
2. Motivating through performance reviews

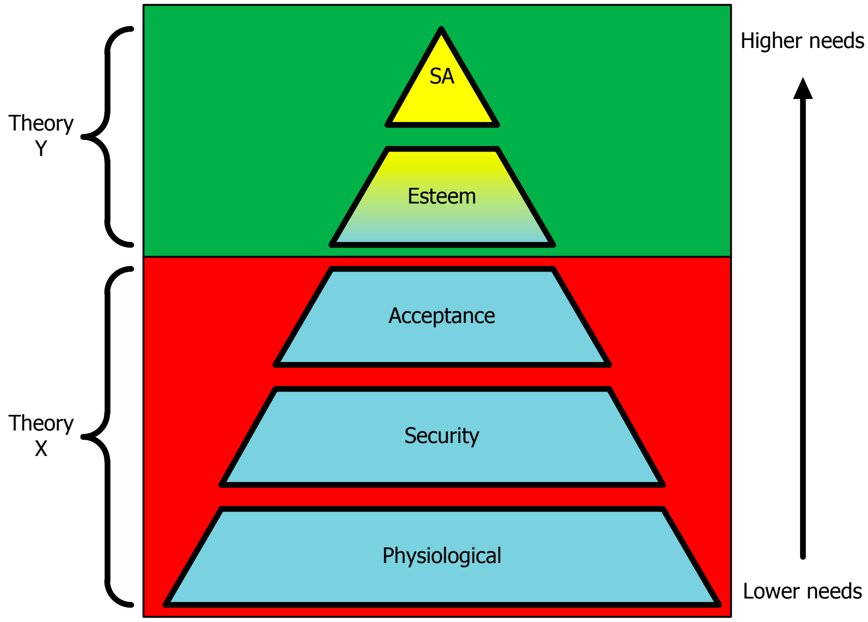
Understanding Staff's Needs

Abraham Maslow defined human needs into five basic categories called ***Maslow's Hierarchy of Needs***. These categories are defined as a ranked hierarchy of basic human and organizational needs. Maslow concluded that the needs at the bottom must be satisfied *before* the needs at the top:

Need	Overview
Physiological	Basic needs (software tools, PC, indoor plumbing, electricity)
Security	Freedom from fear (regular pay, accepted rules of engagement)
Acceptance	Part of the team (accepted, participant)
Esteem	Feeling of importance (recognized, achieves, confident)
Self−Actualization (SA)	Working to full potential (over achieves, intuitive, a true master)

Maslow's Hierarchy of Needs (bottom to top)

Maslow's Hierarchy of Needs is usually shown as a pyramid with the base being the lowest level (its foundation) and the top pinnacle (where you want your team to achieve):



How Hierarchy of Needs overlays with Theory X and Theory Y

Douglas McGregor's work, an alternative to Maslow's Hierarchy of Needs, shows that the lower three "needs" (physiological, security, and acceptance) are what motivates the Theory X workforce and the upper two (esteem and self-actualization—SA in the figure above) are key traits of a Theory Y workforce.

McGregor's Theory X and Theory Y has been used since the 1960s to describe how two very different employee attitudes directly impacts how management should motivate them. Theory X assumes that the staff member *doesn't really want* to work forcing you to micromanage them. There is usually little trust between management and employee. Theory Y organizations, on the other hand, assumes staff members *want to work*, they can be empowered, and there is a high degree of trust between management and employee.

As a case in point, look at how you manage your daily Scrum meetings? Are some team members not taking the lead and just waiting to be given assignments by you? Chances are they are acting in a Theory X way and operating according to acceptance or security level. This may be in contrast to those members who are actively engaged and making every delivery on time while signing up to more than you'd normally expect.

For those Theory X folks, you could keep the interactions the same as before and actually get them operating at a higher (esteem) level by doing more asking (and not telling). Congratulating minor accomplishments of a Theory X developer in front of the team can actually do wonders. That's what you want—to move someone up the pyramid. On the other hand, if there is no real improvement and the developer can't seem to rise to the occasion, you may wish to consider working the individual out of the team (or even the company). The reality is that to succeed you need your team to operate in a Theory Y way with everyone having high esteem and able to self-actualize.

Performance Appraisals Can be Motivating?

Select one of your project team members and honestly answer these questions about them—from *their* point of view. Put a checkmark (✓) or an "x" (✖) next to each question indicating a "yes" or a "no" ("maybes" aren't allowed in this survey):

	I know how I am performing
	I know what my priorities are
	I know how my project relates to the overall strategy
	I know what I need to do to advance my career
	I have regular performance reviews with my boss

"Do you know how your employee is doing?" checklist

If you checked all of them—congratulations! You and your project developer are in absolute synchronization. If you only have one (or two) items checked, you're not doing your job and you're at risk with retaining that employee. According to *Managing for Dummies*, there are benefits why performance appraisals are necessary for employee motivation and retention:

- Chance to summarize past performance and establish new performance objectives.
- Opportunity for two−way, meaningful communications.
- Forum for career planning and development.
- Last, but not least, formal documentation of a person's performance.

Every company has their own performance feedback standards, policies, and procedures but here's a few effective "rule of thumb" approaches:

1. Never wait until the mandatory performance review process to prepare annual performance appraisals.
2. You can ask for self-appraisals until you are "blue in the face" but ultimately you need to retain performance facts throughout the review period (usually a year) for every individual.
3. Perform regular performance reviews (monthly works best) that are designed for two-way discussions between the manager and the employee. Negotiate the top two or three things that need to be completed and how success will be measured. One month later, review with the employee how they did and repeat the planning for the next month.

   And all it takes is a measly 20 minutes a month to manage an employee's performance expectations. Each of these mini−performance reviews should include a summary of the overall goal of the department and prioritized tasks.

A Final Word

Becoming more effective at meeting your staff's needs should free up more of your time for planning and process tasks. Practice these two simple leadership activities and you should see some real motivation and performance improvements. And the best thing of all? The investment is very small!

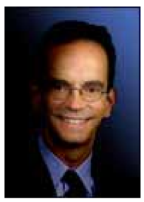
Bibliography

Nelson, Bob and Peter Economy. *Managing for Dummies*. Foster City, CA: IDG Books Worldwide. 1996.

Project Management Institute, Inc. *A Guide to the Project Management Body of Knowledge: PMBOK® Guide, 4th Edition.* Newton Square, PA: Project Management Institute, 2008.

Whitaker, Ken. *Principles of Software Development Leadership: Applying Project Management Principles to Agile Software Development*. Boston: Course Technology PTR, 2009.

Bio



Ken Whitaker of Leading Software Maniacs™ (LSM) has more than twenty-five years of software development executive leadership and training experience in a variety of technology roles and industries. He has led commercial software teams at Software Publishing (remember Harvard Graphics?), Data General, embedded systems software companies, and enterprise software suppliers. Ken is an active PMI® member, Project Management Professional (PMP)® certified, a Lewis Institute instructor, and a Certified ScrumMaster (CSM). Sources for LSM's presentations come from case studies, personal leadership experience, the PMI *Project Management Book of Knowledge* (*PMBOK® Guide*), and Ken's two books: *Managing Software Maniacs* and *Principles of Software Development Leadership*.

Leading Software Maniacs is proudly associated with: