Leading SoftwAre MaNiACs™

Seven Successful Habits of Effective Software Leaders

Date:	Mar 25, 2011	
Title:	Seven Successful Habits of Effective Software Leaders	
Code:	W7SH	
Abstract:	It is easy to find a million ways that software managers can fail with their teams and their projects. We've prioritized seven practical leadership tips and techniques that can help build great teams that consistently deliver great projects. These habits are so simple you can put them into practice immediately.	
Version:	1.0 (2011-03-24)	Original (the follow-on to "Seven Deadly Habits of Ineffective Software Managers")

Is There "Trouble in River City?"

If your team is constantly motivated and focused: wonderful! If your projects are always on time and on budget: congratulations! This article will be probably be a waste of your time. Otherwise, please read on …

The wonderfully-titled book, *Troubled IT Projects*, gives us a "heart-warming" synopsis of what the author believes represents the key root causes of troubled projects based on a study by KPMG:

- Lack of senior management involvement and commitment
- Failure to focus on key business and customer needs
- Failure to break complex projects into manageable "chunks"
- Poor (or unimaginative) project management
- Poor risk management and contingency planning
- Unclear contracts and poor contract management
- Insufficient focus on user training needs

With that background, I've amassed seven habits that you may wish to consider using on a *daily* basis with your teams and on your projects. And that's actually the secret: to provide a handful of useful best practices that can really make a profound difference with your ability to lead resulting in successful projects and teams.

Habit 1: Don't Use Project Management Jargon or Impose Useless Process

"I've spent all of this time and expense to learn project management fundamentals and now that I'm PMP certified, I have lots of flashy terminology I'd like to use." Why not confuse everyone with a vocabulary composed of terminology specific to the *PMBOK® Guide* (example: BCWS) that nobody quite understands?

Don't do it!

You will only confuse, alienate, and diffuse the real benefit of project management best practices.

Keep the buzzwords out of your conversation and messaging by putting yourself in your stakeholder's shoes. Ask yourself, "How would I want to know about the project in ways that would make sense?" I've seen too many project managers surround themselves with sophisticated jargon that if constantly regurgitated verbally and in writing, alienates them from the very team they're trying to bond with. (This approach is similar to agile project teams translating complex technical features into *stories*.)

If you can't fit pertinent project status on a single page of paper (or on a couple of PowerPoint slides), stakeholders lose interest in the details that you think are so important. It all comes down to the following highlights, doesn't it?

1. Is the team on track (and how do you know)?
2. What are the major risks and what can we do to help mitigate them?
3. What work is left to complete?

That's all there is to it. If mounds of project reports have no real value to stakeholders, stop spending time creating them.

Habit 2: Your Job Is to Make Everyone Else's Job Easier

Your role as a manager, leader, and facilitator is to remove obstacles so that the team can concentrate on delivering quality results. It is the agile way, right? Let's look instead at what you can do to make your life *easier* and your team's life *worse*:

1. Delegate mundane administrative duties to your team even though you could easily do them yourself.
2. Demand endless status reports (in fact, the more the better—let's pretend that stakeholders actually read them).
3. Make sure you occupy half of your team's day in meetings—besides the more communications the better.
4. Expect everyone to be absorbed in e-mail and to respond to all e-mails regardless of importance instantly. Wouldn't you rather have your team "e-mail yapping" than actually writing code.

By the way, don't do *any* of these recommendations—instead do exactly the opposite! Make every morning a *positive morning* by thinking like this:

> "Now what can I do to make my team's life easier today so that they can be as productive as possible?"

Habit 3: Project Leadership Is Better Than Project Management

This book's intention is to hopefully demonstrate best practices to enhance your role as a leader in guiding project teams through overall integration and risk planning. Taking the best thoughts from the *Project Management Practitioner's Handbook*, keep these key leadership attributes in mind:

- Leaders take a holistic perspective by including the people side in project management
- Leaders embrace change
- Leaders look for creative ways to align people and other resources to focus on achieving results by increasing project efficiency and effectiveness (in other words, continuous improvement)
- Leaders emphasis is on what and why rather than how.
- Leaders demonstrate the ability to motivate (even when times are tough).
- Leaders constantly ask themselves "what can I do to achieve the vision?".
- Leaders take smart, calculated risks.
- Finally, leaders always put the customer first (remember the Decision Pyramid?).

What distinguishes leadership from management? Keep practicing the above in everyday life, and you'll be recognized as a project leader (instead of a project manager). According to the *Practitioner*'s authors, Kleim and Ludin:

> *Project management* uses the tools, knowledge, and techniques needed for defining, planning, organizing, controlling, leading, and closing a project.
>
> *Project leadership* is the only function that occurs throughout the project cycle. It is, in many ways, the glue that holds the other functions together.

Habit 4: Assume You Are At Risk of Losing Your Staff

Hopefully, you've read Andy Grove's excellent book *Only the Paranoid Survive*? Intel built its business assuming that competitors were going to "bury them" (even if not true). This constructive paranoia helped the management team succeed, become more efficient, and listen to customers. This enabled Intel to retain a clear ownership in CPUs for PCs and then in the mid-2000s grab Apple's business.

Your biggest asset is your team—your staff. And they walk out of the office every evening to go home. What if they don't come back? What impact would that have on your business.

Money initially attracts software developers but motivation and inspiration keeps employees. The *PMBOK® Guide*'s human resources knowledge area defines processes on methods to attract, motivate, and retain staff members.

Some basic employee retention tools and techniques can go a long way:

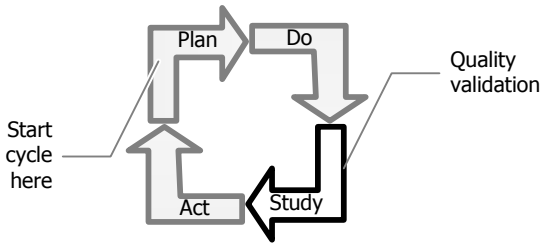
1. Communicate effectively (especially two-way) but do not become overly bureacratic or meeting bound.
2. Believe in and communicate the company vision and how your project goals support that mission.

3. Find out what motivates your staff and do your best to make sure you help deliver on that (for example, training or a specific career path role).
4. Last, but not least, negotiate project priorities with each team member and then deliver frequent performance feedback (monthly works out great).

Habit 5: Never Sacrifice Quality, Design in Quality Instead

The traditional waterfall method of software delivery incurs the greatest risk where the amount of testing is delayed until the total functionality of a project reaches completion. This impacts both cost and time (remember the *Triple Constraint*?). It isn't unusual in waterfall projects where there is tremendous pressure to cut down (or cut out) testing at the tail end of a project in order to make a schedule commitment.

Alternatively, breaking up any project into subprojects and leading the team through mini-releases (plan, do, study, and act) is a better way to deliver projects to market, as shown below.



PMBOK® Guide reinforces the importance of executing through a series of PDSA (Plan-Do-Study-Act) cycles to create a product (or services). This approach has numerous benefits:

1. It dramatically reduces overall project risk.
2. Gives the team a sense of accomplishment early in the project.
3. You always have a buildable, tested product (at the end of every mini-cycle).
4. Encourages customer interaction along the way to avoid releasing something that may miss the mark resulting in possible project failure or non-acceptance (by the customer your project should ultimately satisfy).

Change impacting quality shouldn't be feared, instead you adapt to change as you go.

Habit 6: Be Realistic in Planning Project Schedules

So you're in the middle of a difficult situation. So, your team is telling you that they "can't predict a schedule—it will take as long as it takes." Alternatively, you're probably getting pressure to commit to a schedule for obvious business reasons.

Without a schedule …

- The Finance department won't know what to budget.

- Marketing can't properly launch all of the salvos at the competition and media to generate interest (so that Sales can sell)!
- Sales can't plan when the software (or service) will be ready to generate revenue.

The obvious benefit of adopting an agile process makes it a lot less risky to deliver a product when:

1. Every few weeks you'll have a new, tested product ready to review (and possibly ship).
2. Early adopters can look at the software and give feedback (and potentially even give it a go and test it).

Agile frameworks provide many innovative ways to schedule projects and once you have that schedule agreed upon by the team, you may wish to utilize a *Rough Order of Magnitude* (ROM) scheduling technique for all outward communication. This technique approximates schedule major milestones in a way that gives the team the leeway it needs and gives other organizations in your company enough information so that they can do some basic planning.

As an example, consider an agile project that believes that it can deliver a full project in five to six Sprints. This is how a project key milestones might look as it is being reported over a five month period:

Month Reporting	**Early Prototype**	**Technical Feasibility**	**Release to Marketing**
Feb 2009	Mar 2009	Q2 2009	Q3 2009
Mar 2009	Mar 10-14, 2009	Jun-Jul 2009	Q3 2009
Apr 2009	Mar 10, 2009d	Jul 2009	Q3 2009
May 2009	Mar 10, 2009d	Jul 10-24, 2009	Aug-Sep 2009
Jun 2009	Mar 10, 2009d	Jul 7, 2009d	Sep 2009

The way you read this is as follows.

In February, the schedule for Release to Marketing is sometime in the third quarter. Three months later (May) the early prototype was completed (signified by the "d" following the actual completion date), the technical feasibility milestone has narrowed to July 10-24, and the release is now assumed to be between August and September. The remaining milestones firm up as time goes on.

As a result, the schedule solidifies and "narrows" (you could call this a form of *progressive elaboration*) over time. This technique keeps the mechanics of the agile process in use away from sales and finance (which probably is a good idea anyway) and keeps everyone informed in a manner that makes perfect non-technical sense.

(Now all you have to do is train your peers to accept this technique!)

Habit 7: Dramatically Simplify Your Communication

Have you ever attempted to communicate what you do to your mother (or someone not associated with the software world)? Does the blank stare tell you something?

What about communicating with technical people with everyone "littering" their dialog with monster big words and terms (like "ubiquitous" and "synergistically") that you'd never use in normal conversation? Impressive, heh?

Those are both two extremes, for sure, but it goes to show that the simpler you communicate, the more successful you will be leading and facilitating direction with the team and stakeholders.

Regardless of the audience, your ability to effectively communicate is critical.

Seems rather obvious—but is it?

Communications things not to do:

1. Lay blame or avoid ("stretching") the truth about your project's status.
2. Become bureaucratic and meeting-bound—you must use the team's time productively.
3. Assume that one-way communication is as effective as two-way communication that emphasizes a "feedback loop."

Communications things to do:

1. Identify and classify stakeholders and communicate appropriately.
2. Communicate urgent status change in a timely manner.
3. Use project objective measurements (Planned Value, Actual Value, and Earned Value) to consistently report project status.
4. Develop a culture of listening, respect, and trust.

The *PMBOK® Guide* makes the statement that about 90% of a project manager's time should be focused on some form of communication. And, you haven't *really* communicated unless the receiver of the information understands it and is able to act upon it.

A Final Word

There's one more parting habit that can make a difference. In fact, I believe I've saved the best to last. Consider waking up every day with the attitude of "who can I make happy today?" rather than "oh man, my project is behind and how am I going to tell my boss?"

You'd be surprised how a positive attitude renews the soul, inspires others, and focuses energy on achieving positive results. (I'm serious.)

Bibliography

Grove, Andy. *Only the Paranoid Survive: How to Exploit the Crisis Points That Challenge Every Company*. New York: Doubleday Business, 1999.

Hamel, Gary. *The Future of Management*. Boston: Harvard Business School Publishing, 2007.

Kliem, Ralph L. and Irwin S. Ludin. *Project Management Practitioner's Handbook*. New York: AMACON, 1998.

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

Smith, John L. *Troubled IT Projects*. London: The Institute of Electrical Engineers, 2001.

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:

