

How to Make the Right Project Decisions

Date:	Apr 9, 2010	
Title:	How to Make the Right Project Decisions	
Code:	WDCA	
Abstract:	This article lays a solid foundation for a project team to make key decisions that can enhance or derail a project getting completed in a timely fashion. How decisions are made can also have impact of motivating or demoralizing your project team.	
Version:	1.0 (2009-12-23)	Original (excerpt from Leading Software Maniac's <i>Deliver Projects On Time, Every Time</i> seminar)
	1.1 (2010-04-09)	Added real life corporate examples
	1.2 (2010-04-16)	Posted on LSM Web site (along with movie clip)

Overview

Every software project, agile or not, brings with it many decisions and requests that must be handled quickly and decisively. Some say, “*Any* decision is more important than making *no* decision.” Not true— the *wrong* decision may haunt you, your team, and your project forever. When key decisions need to be made, there is usually some *influencer* that stands behind it. For example, if an unplanned, urgent feature is required to be implemented, there is usually a customer behind that request. If a project delivery schedule needs to be made (even if the product being developed hasn't been completely tested), there is usually a company financial motive behind it.

You get the idea ...

Background

If these situations are typical of what you've had to deal with, this article could be important for you.

Situation 1: Let's Do Everything!

There is a project that has started with a new set of features identified. This is going to be a major product update. Collectively, there is a high probability of failure, which the team has estimated to be around 75%, if the team attempts to build everything requested. (One notable reason is that every module would have to be modified!) On top of these feature requests, management has requested performance improvements. You've estimated that there is far less probability of failure (25% to be

exact) if two of the key features were implemented with no attempt at performance improvement made.

The team decided to take on the “all features with lightning—fast performance” route and like any great agile project, the two most important key features were designed, prototyped, engineered, and tested first. With an agile approach, *anything is possible!* No doubt, the customer, product management, and the executives were excited.

The team kicked off the work ...

Although the two key implemented features functioned as expected, the performance was not blazingly fast and management was concerned that unless time was spent on improving performance there would be significant risk trying to get all of the features to operate at the expected level of performance.

The team, however, was convinced that they should “do the right thing” and fix the performance issues and complete the rest of the features even if there was schedule impact. “The customers would want this,” was the overwhelming team war cry. The customer only knew about the key features—they hadn’t expressed the need or commitment for anything more than that. (But if they got more features than expected, they’d certainly be pleased!)

Finance, of course, wanted the product released as soon as possible.

Talk about competing influencers!

As a result, the project dragged on and on. The number of expected iterations (Sprints) were extended until, out of exhaustion, the team eventually completed the “must have” features. Ill-fated attempts at resolving underlying performance issues were scrubbed.

Situation 2: How Are We Making Decisions Around Here?

You, as the overall leader, enjoy motivating teams to deliver projects on time. But what really gets you excited is to provide solutions that customers want to use. You could say that your mantra, not unlike the mantra of your company, is “customer first.”

With a pending release slated for this quarter, the project is a little bit behind. Unfortunately, there were too many quality issues remaining with a handful of key features that needed to be implemented. As the team edged closer and closer to delivery, you received a telephone call from a key customer, who demanded a necessary feature to be enhanced. To make matters worse, his call was followed up with a call from your VP of Sales stating that “you better do what this particular customer asked.”

At the next Daily Scrum meeting, you explained the situation to the team. They needed to change what they were going to work on during the next Sprint and focus on this new requirement. “Definitely, we’re a customer-centric company,” is the thought that crossed your mind. Some of the engineers were upset with the change of direction, but they understood that the customer paid the bills.

The following week, one of your best engineers says that she uncovered an issue in the code that if resolved could actually significantly improve the product’s reporting performance. This hadn’t exactly been a customer’s request, but there was no doubt that not making this fix could be a problem the more that customers ran reports. This task really interested her and, in her mind, was more important than what she was working on.

“This is the beauty of Scrum—we can adjust as we go!” Yeehah!

In this case, you approved the change and tell the team of the decision. The software engineer was ecstatic. “Awesome,” proclaimed a QA tester on the team, “but we need to run *before* and *after* benchmarks. We don’t have those tests automated and we’re focusing our testing efforts on the agreed-upon features.” Knowing that you don’t want to upset the software engineer’s desire to improve the

product, you still agree to make the improvement anyway—the reward will be better than the risk. *The team is really getting nervous about another untimely change.*

Both of these situations have unfortunately set the team back and with the last day of the quarter looming, the CFO made a rare visit to your office proclaiming that “this project better ship, otherwise we’ll have a disaster reporting another quarterly loss to our investors.” Sweat poured out of your head. “But, our testing may not be completed.” “I don’t care,” states the CFO, “I’ve talked this over with the rest of the executive team. We can always come out with a ‘.1’ update release after we ship in the early half of the next quarter. I’m counting on you to live up to your original commitment for on-time delivery.”

That afternoon, you address the team and tell them the news ...

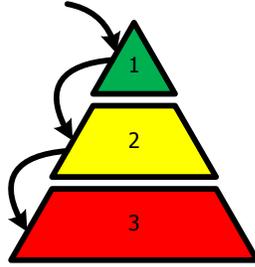
How Should We Make Key Project Decisions?

Both of these situations, although contrived for this whitepaper, are all too real for most of us that have been in the software industry for a long period of time. Competing pressures that impact decisions that can either enhance or derail a project are a part of the daily role played by every software manager, project manager, and ScrumMaster.

Is a mechanism really needed to provide structure for making key project decisions? You bet! Being in a leadership role, you’ll be pressured from influencers wanting decisions made for the best interests of the company, the employee, or the customer (and they are typically conflicting needs). Let’s look at the three categories of influencers (or as I like to call them, **Decisionakers**, short for decision makers) for any typical software project:

Decisionaker	Who They Are
Company	Most concerned about the business (read stakeholders, quarterly earnings, and so on)
Employee	The team that produce the products or services being developed (engineers, testers, and so on)
Customer	Those individuals that represent the needs of the market, users, and resellers/partners

We'll use a pyramid as the basis of how decisions should be made:

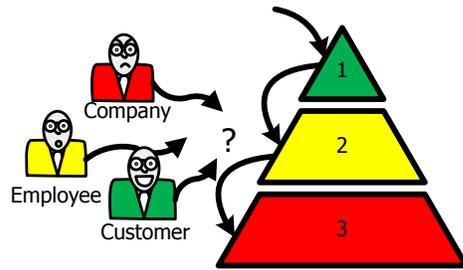


The way to read this diagram is as follows (starting from the top):

With every decision, you'll start at the top and decide the outcome (example: which features to prioritize). If the decision criteria you use results in a bought-in outcome, then great—you are done! However, it is never quite that easy (especially if you have to trim the decisions to fit a budget, resource constraints, or even schedule). At that point, you'd drop down to a second-level of decision criteria to further help in a final outcome.

Of course, if you still need further refinement, you can use yet a third criteria to decide the final outcome. (We don't recommend more than three levels, by the way!)

How does this relate to the Decisionmakers discussed earlier? I'd propose that you need to place one of these in each of the decision groups:



#1 Decisionmaker – the King of the Mountain, who generally makes the decision?

#2 -- Who is next most important if the #1 group can't decide?

#3 – Last but definitely not least, Who provides the foundation that everyone relies on (when #1 and #2 can't decide)?

If your company places stockholders as the most important decision criteria, then the Company should be placed in the number 1 spot. What that means is that key project decisions with the greatest benefit to the company wins. As an example, if the team is deciding between a feature to include that has great customer benefit versus another feature that has much bigger financial benefit, the latter feature will most likely be chosen (over the customer requested feature).

If, as another example, you make decisions based on benefit to the employee (“if you build great technology features, then customers will come!”) then you may decide on something that is whiz-bang technically innovative and hope that customers will want it and that it will bring benefit to the customer.

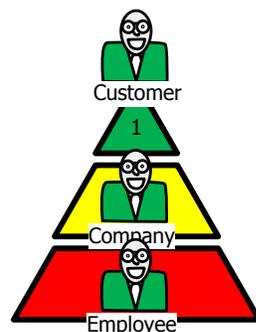
You get the idea ...

The exercise to come up with your company's Decision Pyramid may be a particular good thing for start ups and for organizations that are being transitioned to a new and better structure.

If you aren't sure what the priority should be, project decisions may be more “ad hoc” than you think and it may be a good idea to brainstorm with your management to come up with your company's Decision Pyramid. Once you have defined it, you need to consistently follow it. Otherwise, the very “bones” of the company (in other words, what you stand for) will be under scrutiny by representatives of the three classes of Decisionmakers: the customer, the company, and the employee.

... Not a good thing!

What Decision Pyramid do I recommend? Working over the years with companies that have had different approaches to consistent decision making criteria, there is only one that I have found to be long lasting:



Yep, you guessed it: Customer first!

Making decisions first and foremost for the customer brings business opportunity. With a satisfied customer and a successful company, what software employee wouldn't want to work for a company that practices that priority?

A Final Word

Although not a software/IT example, if your company was a leading financial institution, how would you like your company to be ranked on a least-trusted list?

The Least-Trusted Banks in America

by Jennifer Saranow Schultz
Tuesday, February 9, 2010

provided by

The New York Times

Forrester's annual Customer Advocacy survey ranks nearly 50 financial services firms in the United States by the percentage of each firm's customers who agree with the statement: "My financial provider does what's best for me, not just its own bottom line."

About 40% of Wells Fargo customers agreed (the highest of the large banks). Contrast that with HSBC which scored about 16% of customers surveyed agreed with that statement. Why? "Part of it is that the banks are preoccupied with their bottom line. They are public institutions who are in business to make money for their shareholder and inevitably, that shows to customers," says Bill Doyle, VP at Forrester.

What's the impact?

Customer advocacy rankings are a predictor of customer retention and attrition, and customers who rate their firms high are more likely to consider their firm for additional products. In contrast, customers who give a low ranking are most likely to switch in the next year and are "going to be reluctant to put any more money and open new accounts at those institutions."

This could be a disaster for software/IT companies in an unforgiving industry where there are few second chances. One of the few industries where COGS (Cost of Goods) is basically pennies with potential for huge profit margins, there are still tons of companies who haven't lasted due to corporate arrogance (SCO Systems, Data General, Lotus, Ashton Tate, and so on).

Recently, we all witnessed the demise of the nationwide electronics chain, Circuit City:

Inc.com

Why Circuit City Failed, and Why B&H Thrives

Many companies that have gone bust didn't die because of the recession. They failed for one reason: They treated customers poorly
By Joel Spolsky | May 1, 2009

According to Joel Spolsky's article in Inc, the tale of a major electronics retailer, Circuit City, is a lesson for all of us on how to do things wrong. The chain's CEO, in an e-mail, blamed Circuit City's demise on "poor macroeconomic conditions and general economic slowdown." True, the economy has been horrible, but have you walked into an Apple Store lately or visited a purchased computer hardware or software from B&H Photo Video (www.bhphotovideo.com)?

What economic slowdown?

Spolsky cites the fact that B&H culture is that sales people are very knowledgeable, frequently try to save the customer money by suggesting lower-priced options, and are empowered to accept returned merchandise if a purchase didn't suit a customer's needs. This is in sharp contrast to Circuit City especially after they let go 3,400 of their most experienced sales people in 2007 and replaced them with untrained, low-cost staff as a last ditch effort to trim expenses. Similar products, similar low prices, and yet the attention to the customer was light year's apart. And even now, when the economy has still not recovered, the main B&H Photo Video store in Manhattan is bustling with activity. Their Internet mail order business is also booming!

How does *your* company truly prioritize the importance of serving the customer?

Bibliography

DeMarco, Tom and Timothy Lister. *Peopleware: Productive Projects and Teams, 2nd Edition*. New York: Dorset House Publishing, 1999.

Kawasaki, Guy. "Make Meaning, Not Money." *Anderson's Business Briefs*. 7 January, 2010 (<http://www.andersonsbusinessbriefs.com/entrepreneurship/2010/01/guy-kawasaki-make-meaning-not-money.html>).

Larman, Craig. *Agile and Iterative Development: A Manager's Guide*. Boston: Pearson Education, 2004.

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

Schultz, Jennifer Saranow. "The Least-Trusted Banks in America." *The New York Times*. 9 Feb 2010. (http://finance.yahoo.com/banking-budgeting/article/108801/the-least-trusted-banks-in-america?mod=bb-checking_savings).

Spolsky, Joel. "Why Circuit City Failed, and Why B&H Thrives." *Inc.com*. 1 May 2009 (<http://www.inc.com/magazine/20090501/why-circuit-city-failed-and-why-bh-thrives.html>).

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:



Applying Project Management Principles to Software Development Leadership, Principles of Software Development Leadership, 4Ps, Leading Software Maniacs, Soft-Audit, jus' e'nuff, Nerd Herd Game, the 4Ps logo, the Leading Software Maniacs logo, and the Nerd Herd Game logo are marks of Leading Software Maniacs, LLC.

The Lewis Institute logo is a mark of Lewis Institute, Inc.

PMI, PMP, PMBOK, the PMI logo, and the PMI Registered Education Provider logo are registered marks of the Project Management Institute, Inc.