

## **Determining Whether to Fund Your Next Agile Project**

Date: Jul 17, 2011

Title: *Determining Whether to Fund Your Next Agile Project*

Code: AR-FND

Abstract: Too many projects and not enough money or resources to do them all! All of us need to make prioritized decisions to determine which projects to fund. Chances are you, in a software leadership role, can't make the final determination alone, but your expertise will be certainly called upon to help make that determination. This document presents tips that can assist you in making those "fateful project decisions."

Version: 1.0 (2011-07-17) Original

## **Help! I'm In Danger of Making Things Too Complicated**

A friend of mine teaches an advanced project management class as part of a masters degree program at a major university. He was asked this question from one of his students:

"I am working on setting up a more streamlined project management process for a brand new IT department. We have corporate processes and tools that we can use, but I need to right size them for our department and our typical project size. I was wondering if you had any examples of scorecards you use to select which projects to pursue that you would be able to share with me? I am looking for something beyond NPV and IRR – something that compares risk, timing, alignment with strategy, etc."

In my humble opinion, the key word used in her request is "streamlined."

*Let's help this student out!*

# Project Justification Approaches

There's probably three approaches to determining whether to pursue a project or not:

- Take a chance—management wants it and it makes overall sense. So, without much debate the decision is simply to just plain do it.
- Put the team (and yourself) through “analysis hell” and come up with all sorts of statistics and measurements that show if the project is even worth pursuing. This process may take weeks, but everyone will at least have all of the facts in order to make a decision.
- Take a balanced approach to make a time-limited decision based on a number of factors including market need, development effort, and business opportunity. (Based on my experience, this can take as little as two to three hours!)

Although the first approach takes the least amount of effort, you really need a little bit more analysis to make an informed (and not a gut) software development project funding decision. More established organizations may decide that the second approach appeals to everyone's analytical nature while the third approach is probably the most realistic approach of determining whether or not to pursue a project. As the title of this document implies, this may be a great way to plan for agile projects where customer focus and flexibility are ingrained in your company's culture).

# Project Cost Calculation Techniques

You're likely to be asked by upper management for some sort of financial analysis that will justify the value a proposed project will bring to your company. You want to definitely keep calculations simple and you work closely with Product Management in a unified analysis. The following table is a summary of the most common project viability calculation techniques:

Calculation Technique	Acronym	What This Is	How Will It Help Justify?
Return on Investment	ROI	The amount of income, not revenue, divided by the investment.	Biggest percentage wins. If a project is projected to earn \$300K of revenue and it costs \$200K to development, the ROI is 50% based on $(\$300K - \$200K) / \$200K$
Internal Rate of Return	IRR	The amount of money a project will return versus the cost of development	This is a rather complicated formula, but most companies place high value on the project

Calculation Technique	Acronym	What This Is	How Will It Help Justify?
		and ongoing support and maintenance.	that offers the highest percentage (even if the revenue isn't the highest compared to other proposed projects).
Present Value	PV	Assuming that a dollar today is worth more than a dollar tomorrow, evaluate the value of a proposed project based on today's dollars.	Assuming diminishing value of the future dollar, how will the proposed project's long-term revenue look? The project with the higher PV wins.
Net Present Value	NPV	Using PV, calculate the net profit in today's dollars.	Due to expected rising labor costs to counter revenue degradation, just how profitable is this proposed project? The bigger value wins.
Benefit Cost Ratio	BCR	Subtract the total cost of doing the project from how much revenue the project will take in.	Largest revenue to cost ratio wins. For example, 2:1 beats 3:2.
Opportunity Cost		The money you would have made because you didn't choose the project.	Smaller number definitely wins (but, you know, this is very subjective and reeks of "bad energy").
Payback Period		A favorite with financial guys, this represents the amount of time it takes to recoup the cost of a project's development.	Shortest time wins. Cute.
Sunk Costs		Costs that have already been spent on a project.	You could say that this technique is "water under the bridge," but the general practice is any pre-existing costs

Calculation Technique	Acronym	What This Is	How Will It Help Justify?
			should <i>not</i> be considered as part of the projected costs.

All of these calculations have merit and basically provide a different view into the financial viability of a project. I'll come back to a recommendation, once a few more topics are examined. The most important thing to keep in mind is that financials are only one piece of the pie. Revenue would be estimated by the Product Owner (or, the Product Manager) and I'll discuss development costs later.

**Note:** Often overlooked is the concept of *timing*. Not taking timing into consideration the calculations mentioned in the previous table could skew the results. Waiting six months for a project that has a short window of opportunity can be project suicide. (And, if your company is considering embarking on a project that satisfies a specific market need, chances are high that your competitors are considering the same thing!)

## Does the Project Align to Strategic Objectives?

I'm a big believer in lists and priorities; I take every opportunity to breakdown any decision criteria with tables. The table below indicates you can lead a team to visualize how projects fit into the strategic vision of the organization. Although largely subjective, filling out this information can be enlightening and I've always found it to be a wonderful team building activity for the cross-functional team of stakeholders.

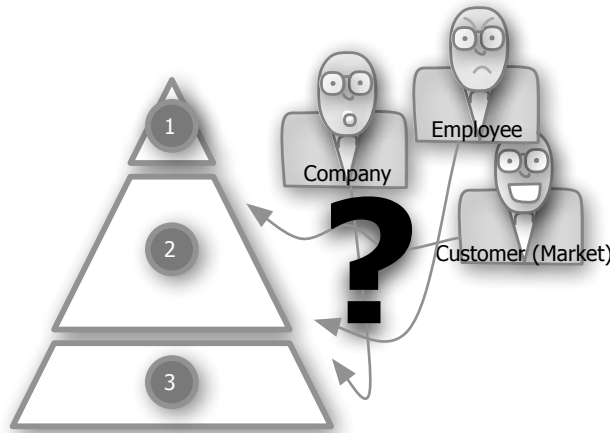
The project ...	Project A			Project B		
	Not Really	Sorta Does	Yes	Not Really	Sorta Does	Yes
Supports the company vision (the "alignment")		▪				▪
Will cause competitor discomfort (this is good!)	▪				▪	
Will delight the customer (the market)			▪			▪
Will put the company "on the map" (a real game changer)?		▪				▪

In the above example, Project A basically satisfies customer and market needs, but isn't much of a bold move. In fact, it sounds like an upgrade focused on some non-core set of features, doesn't it? and Project B benefits the customer while providing innovative value supporting your company's core strengths. (If I had the choice, Project B would surely earn funding consideration over Project A.)

In addition, your team should consider what is the principle motivation for ranking one project higher than another?

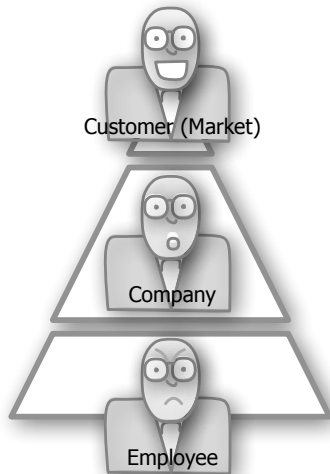
I use a tool called the **Decision Pyramid** to illustrate the "pecking order" associated with the hierarchy of stakeholders representing the needs of the customer (the market), company, and employee. The combination of "energy" it takes to try to please everyone along with the lack of consistent decision rules (as to which group is most important to satisfy) can result in a "team meltdown" and ultimately lack of

confidence in the leadership of your company to make a decision. So the trick becomes, “which of these groups is the most important decider?”



Based on what employees want to work on or what the marketplace (in other words, the customer) needs? Or, heaven forbid, you make funding decisions sometimes for the good of the employee because if you don't, the employee might walk!

For this reason, I'd recommend a single hierarchy (hence, the pyramid) to ensure that the most important stakeholder group, #1 in the figure, is satisfied first. And if there's a tie between two projects, drop down to compare the #2 priority. Based on experience and training hundreds on the concept of the Decision Pyramid, one ordering stands out:



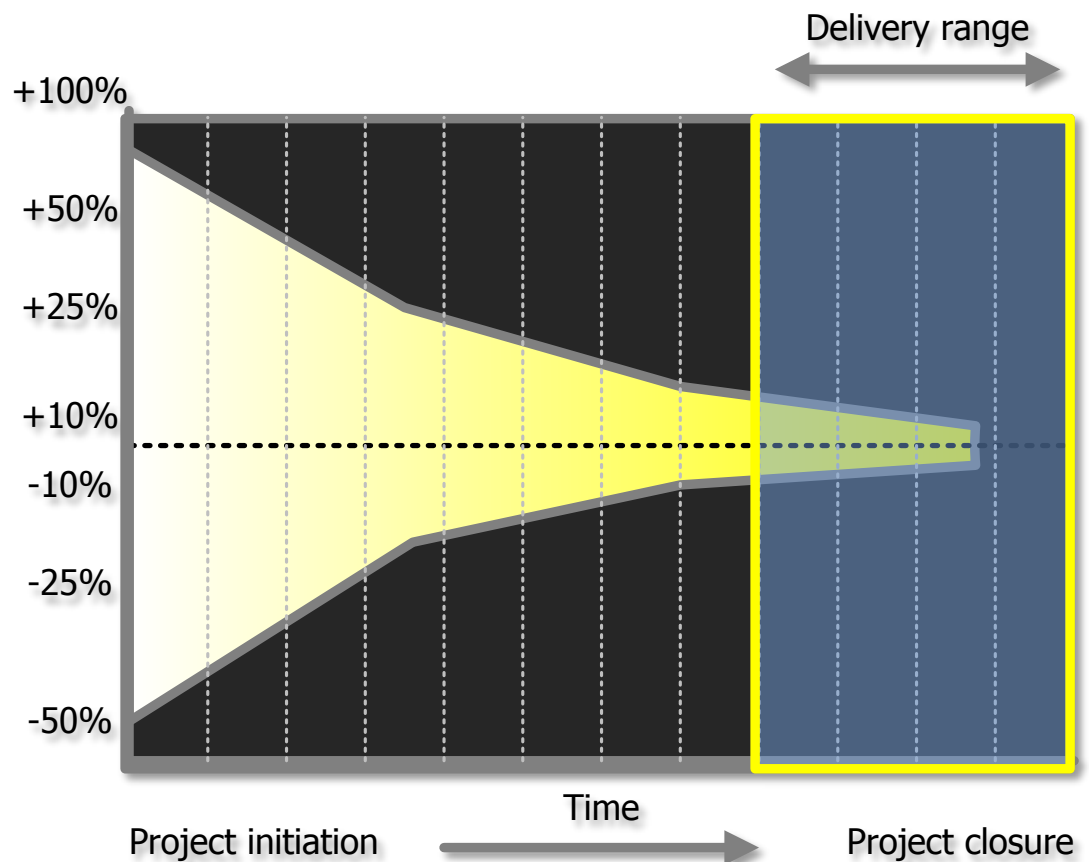
You guessed it! Justifying a project by how well you satisfy the customer (market) will most likely result in your company doing well. And if your customers are happy and the company is thriving, employees should be thrilled. How many among us,

*especially* in these difficult economic times, take customer satisfaction or a profitable company for granted? Nobody that I know!

## Bullet-Proof Way to Estimate Project Costs

Most ScrumMasters that I meet are fairly confident with their ability to predict realistic project schedules assuming that the assigned team are highly-qualified developers that have a history of project success.

Estimating a complex project with a lot of unknowns can be close to impossible, yet executive management really needs to know what the cost is going to be in order to pluck in the cost calculations already mentioned earlier. A simple technique is to get the team together and produce a **Rough Order of Magnitude (ROM)** level of effort estimates. As long as you have a set of resources identified a great team effort in the early Scrum Planning phase is to get the team together with the Product Owner. Get team buy in of ROM level of effort and as you go through the **Must Have** list of features, plan out how many Sprints (iterations) you'll need to complete the project.



You may wish to consider estimating work in term of “shirt sizes” like small (less than a day or two of work, medium (about a week’s work), or large (more than one week of work). In the figure above, each vertical line represents a sequence of Sprints. The cone shape represents the level of uncertainty where at the beginning of the project, Sprint iteration 1, the accuracy of when the release should occur is about +/- 50% (in other words, pretty wide) and as the project proceeds, the release schedule becomes much more certain (to about +/- 10%). No surprise, but this graph is known as the **cone of uncertainty**.

Regardless, it would be best to indicate the projected cost as a **range** (which, in this example, is between Sprints 8 and 12). The cost would be calculated as the number of people resources on the project would simply be multiplied by the number of Sprints.

$$\begin{aligned} \text{CostPerSprint} &= (\text{resources} * (\text{avg annual salary}/26)) \\ \text{Minimum cost} &= \text{MinSprintsUntilRelease} * \text{CostPerSprint} \\ \text{Maximum cost} &= \text{MaxSprintsUntilRelease} * \text{CostPerSprint} \end{aligned}$$

If the average annual salary of an engineer is \$90K and there are five resources dedicated throughout the entire project:

$$\begin{aligned} \text{CostPerSprint} &= (5 * (\$90,000/26)) = \$17,310 \\ \text{MinimumCost} &= 8 * \$17,310 = \$138,480 \\ \text{MaximumCost} &= 12 * \$17,310 = \$207,720 \end{aligned}$$

Any practical CFO “worth their salt” is going to budget for worse case, and, in this example, the project will take at most 24 calendar weeks and not exceed \$210K (rounded up) of people resource costs. (By the way, people aren’t the *only* resources you’ll need to consider in the overall projected cost structure.)

## What About Qualifying Risk?

In the case of agile projects, the team attacks (I mean “removes”) obstacles on a daily basis. **Risks** (both negative and positive ones) have the ability to throw a project into mayhem thus impacting the team’s ability to deliver with quality and on time. For any executive team to make a project funding decision, some indication of serious, schedule-impacting risks is necessary. It is easy to forget that the **Risk Management knowledge area** in the *PMBOK® Guide* includes processes that can assist you in risk planning. I choose to include an abbreviated **Risk Register** for each project (that way, if the project gets funded, you already have the basis of a risk register already to go).

The Risk Register below is a simple way to show qualitative analysis of risks in terms of threat/opportunity (T/O), probability (P), impact (I), and urgency (u) as either low (L), medium (M), and high (H).

#	Risk	T/O	P	I	U	Priority
1	New “core engine” improvements	T	M	H	H	H
2	Team attrition that would impact delivery	T	L	H	L	L



A simple calculation can be used to create the overall risk (Priority) as:

$$\text{Priority} = \text{Average}(P, I, U)$$

An overall risk priority for a projected project can sway the executive team to not fund a project especially if the combined risk priority is high and the company culture is risk adverse.

## A Final Word

A summary of a project's information used to get funding should be able to fit on a single page (or at most four to five slides). It should include the following items discussed in this article:

1. Project goal statement and what its main benefits are.
2. Pro Forma justification (ROI works best and have the other calculations ready as background material if asked).
3. How the project aligns to company strategy (use the Decision Pyramid!).
4. Schedule, level of effort, and overall expense (presented in terms of range).
5. Assessment of risk that would have an impact for project success (or failure).
6. Your recommendation (fund the project or not).

I can't imagine a single chart or magic formula that encapsulates all of these important funding considerations, but the combination of the above six elements should provide plenty of information that will help your executive management team make the right go (or no go) project funding decision.

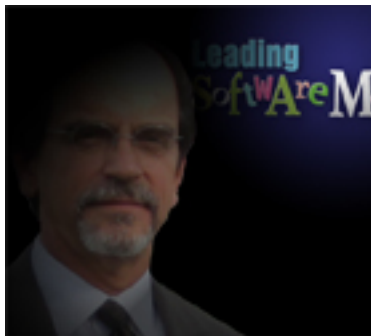
## Bibliography

Cohn, Mike. *Agile Estimating and Planning*. Upper Saddle River, NJ: Prentice Hall PTR, 2006.

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, 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 leadership books: *Managing Software Maniacs*, *Principles of Software Development Leadership*, and *I'm Not God, I'm Just a Project Manager*.

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.

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