

CIO Tools: 'Technical Debt' a framework for proactive business discussions

by Jim Vaselopoulos - Partner, PSC Group

February 2009

Every once and a while something new crops up on the business landscape that truly makes you stand back and think. In most cases I have found that the lasting ideas are so simple and pragmatic that you wonder how they escaped unnoticed.

Nearly a year ago we were embarking upon a campaign to formalize our Agile project management offering and the topic of Technical Debt was introduced to me for the first time. And although it was not a major part of our discussion, it was the part of our conversation that most piqued my curiosity for a very simple reason; the concept of Technical Debt translates a complex technical cause-effect problem into ordinary business terms that our end-customers can readily understand.

Improved Project Communication

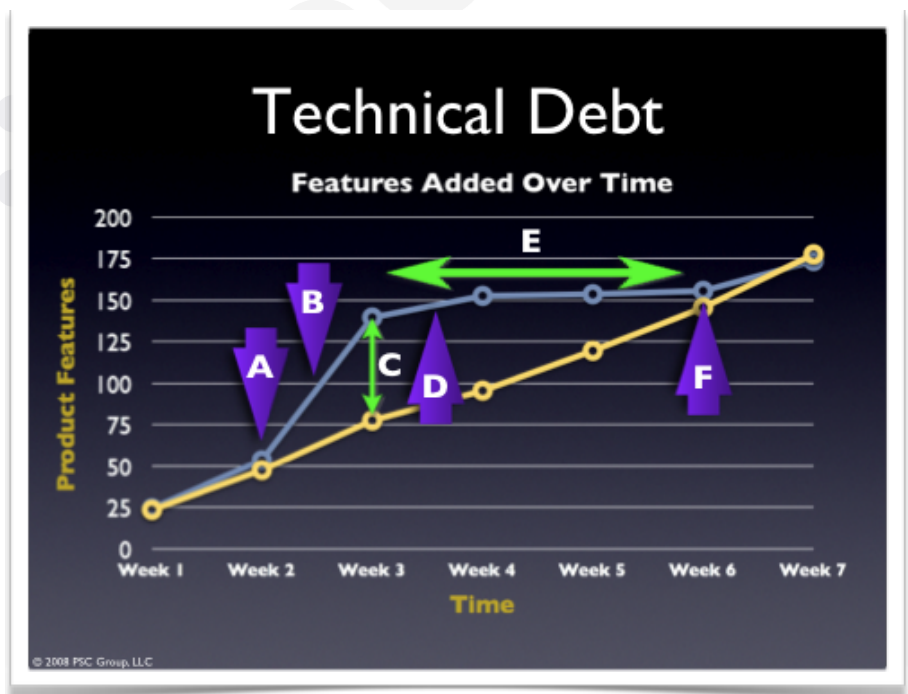
As Agile practices improve the working relationship between the business and technical teams, gaps still remain in how we deal with difficult issues such as accelerated schedules, overtime, and code re-factoring. Technical Debt is an exceptionally powerful concept that provides a framework for building more equitable relationships between the business and technical project teams. When applied as a communication framework, Technical Debt fosters proactive dialogue through the use of up-front contracts.

Up-Front Contracts

The up-front contract is a pretty simple thing. It is a verbal agreement between you and someone else that outlines exactly what will happen next based on a sequence of events. It is good business and good 'neighborly' common sense. The end result is that it is a great way to manage expectations and discuss the outcomes of decisions proactively.

In the context of a complex and lengthy development project, however, it is understandable that the consequences of decisions made now are hard to imagine later. That is where Technical Debt comes into play. The idea of Technical debt provides a foundation for a complex discussion that can be simplified because:

1. it can be visually represented, and
2. it uses the customers language - business terms.



Leveraging this concept enables an IT department that is using a few basic Agile principles to maintain long-term success through a thoughtful dialogue with the business. The outcome is the creation of a forum for challenging situations that occurs when it is most valuable - when it is proactive.

Technical Debt Explained

Although it is not necessary, some basic understanding of Agile PM methods is clearly helpful to understand the concept of productivity graphs, sprints, etc. If you do not have any background in Agile, suspend any criticisms about the scales and measures of the graph and trust that you can find a steady-state expectation for how many ‘things’ your development team can deliver in a common time-frame.

A (see figure) – a business decision is made to go beyond our normal operational expectation of 25 features – we decide to incur debt. It is important to note that this is done in conjunction with the business and the technology implementation team.

B – Debt is accrued in the form of less elegant code that may need to get re-factored later on, stressed out developers, technical shortcuts, etc.

C – Full Debt Load achieved

D – Inefficiency (interest) is incurred – the team is burned out. Notice how the slope is not as steep (the team is not as productive) as it was before the debt was incurred.

E – The debt needs to be retired. Now this does not need to be done right away, but it does need to get done. What does this mean? Shortcuts

that were made need to be resolved. Code needs to be re-factored. Hard-coding needs to be eliminated. User Interface sacrifices need to be resolved. And ultimately, the business needs to provide the time the team negotiated (in the up-front contract) to get back to a point where we can sustain a predictable development effort.

F – The debt is eventually retired.

Better Relationships Via Arbitration

In the end, the Technical Debt framework creates a good business discussion in terminology that is appealing to executives. Everyone understands that debt can not be ignored indefinitely.

By recognizing that we often have to meet business needs that are unexpected we can be better partners with the business. And by having a way to accommodate and manage the effects of these unexpected needs we can make the discussion more of a give-and-take arbitration rather than an all-or-nothing confrontation.

Another great benefit of this communication framework is that it helps us avoid having to sell or justify the zero value-add technical projects (re-factoring, re-architecting, etc.) we have all had to do at one time or another.

It also illustrates how we do not have an endless supply of good will and Mountain Dew with our development team. If pushed too hard we do incur debt through inefficiency, mistakes and shortcuts.

The end result is that we consider the full impact of our decisions before we make them - allowing for more equitable relationships between the business and project teams.

It’s all in the way we listen.™



Jim Vaselopulos

office: 847.517.7200
 mobile: 847.274.9637
 jimv@psclistens.com

Jim Vaselopulos is a seasoned business executive with domain expertise in Financial Services, Marketing, Manufacturing and Service Industries. Jim works closely with many firms to help align business needs and technology for competitive advantage.

His many roles include Partner at PSC Group, LLC, interim CIO at several organizations and strategic business consultant to many others. His speaking engagements include regional executive events, various podcasts, industry organizations and technology-centric educational institutions such as the University of Illinois.

Jim holds an Engineering degree from the University of Illinois and an MBA from Marquette University.