



The Product Marketing & Product Management Experts™

Is Agile development a threat to the product management discipline?

by David Howard

As I read more articles about Agile development, many of which attempt to redefine the role of the product manager, I ask myself – is Agile development a threat to the discipline of product management? I've read forums and blogs, and talked to colleagues and friends about this issue. Some people, engineers, product managers and business managers among them, have been outright derisive of the Agile development model. Yet the movement continues to gain momentum and new adherents. I wonder how it can sustain this momentum – and the threat – given the proven value of the product management function within high-tech companies.

The reason is that many people mistakenly believe that “Agile development” – a software development methodology – is a product management methodology. While it may have a role to play in the overall product development philosophy of a firm, it simply is not. Indeed, it is routinely defined as a family of software development processes (Wikipedia) or a collection of ways of developing software (Agile Manifesto); it is contrasted with the waterfall software development model and the Agile Manifesto was written by authors who were first and foremost software developers and programmers and not product managers.

If you believe that product management is really a business management function, as I do, and not exclusively a software development or engineering function, then Agile development should be viewed as only one component of the organization's product management model. Product management often resides within the engineering department, or, in a firm that recognizes the function as the nexus of customer needs,

market assessment, financial justification for development activities, and clearly articulated engineering requirements, within Marketing. I would go a step further and suggest that, especially for young and growing technology firms where initial product success equates directly to enterprise success, product management, as a business management function, warrants it's own department with board level visibility.

What makes product management a business function? And why is Agile development not the same as managing product development? As I look through my product management toolbox assembled over the years, I can identify from checklists and charts several product management tasks and deliverables that originate from outside of the software development group and are therefore un-related to the implementation of an Agile development model:

- **Requirements development and business case.** The financial justification for the development of a new product or new features is a business function, not a software development function, and is often performed by product managers or marketing analysts. It may make sense to communicate the output of this exercise to engineering in an Agile development model, but this is clearly component of a broader product management methodology.
- **Field and beta test programs.** As with the requirements and business case development task, some outputs of beta test programs are inputs to the software development team, within an Agile or any other software development model. However, management of the programs is an exercise aside from the development of the software. Components of the product aside from the software, such as technical support, manufacturing, pack-out and ship requirements, and out-of-the-box experience are routinely tested, and the product manager must assure accurate collection of data and reporting on all components of the beta product.
- **Pricing plan, gross margin analysis.** Clearly, the deliverables from this exercise are not an output of the software team's development model, but a function of product management working with senior management, finance, and depending on the product, with manufacturing and engineering for cost inputs.

- **Sales order entry and product initialization.** It is the product manager's job to make sure the product is orderable and orders can be fulfilled. Part numbers need to be entered into the order management system and enabled, and service people need to be trained. Customers must be invoiced and payments collected.
- **Product technical documents and user manuals.** While Agile development attempts to minimize technical documentation as part of the development process, it cannot eliminate the need for product manuals and other support documents for the end user. And the production and delivery of these documents needs to be ensured by the product manager.

We could go on to review many other product management deliverables outside the scope of Agile development, such as positioning statements, marketing collateral, training plans, regulatory or export certifications, but it should be clear that Agile development occupies only part of the product manager's mind.

Let's shift now from the perspective of the product manager to the perspective of the product itself. Ted Levitt's concept of "whole product" marketing, popularized by Geoff Moore and Bill Davidow has long been a standard framework around which technology companies have developed products.

If you adhere to the "whole product" philosophy of technology product management, you may view the final output of the Agile development process to be the *core* or *generic* product, around which the product manager must wrap the *expected*, *augmented* and *potential* product. The whole product, beyond the core product may consist, for example, of customer service and warranty provisions, system integration, support, partnerships and interoperability, company and product brand equity, training and so on, most if not all are fruits not of the software development model but business management efforts by the product manager.

Trading metaphorical tornadoes for sandstorms, Joe Bentzel, in his recent book [Asymmetric Marketing](#), rejects the classic whole product model and chasm theory but

still advises marketers to wrap their core technology in “**compound sets of services and intangibles.**” Bentzel writes of “wrapping one’s core offering in a set of related services,” to create a compound product which “integrates code, services, content, communities, commerce, knowledge-bases, intangibles and more.” Once again, these intangibles that surround the core product must be delivered with the product by the product manager, and except for the code, they are not a deliverable from the software team, regardless of the development model. With apologies to Bill Davidow: great technology is invented in the laboratory; great products are invented by product management.

Additionally, the concept of product must be put into the context of the type of firm performing the software development. For small information technology consultants delivering custom software development projects, perhaps the software deliverable often *is* the whole product, and perhaps in this case product management does equate directly to software development management. But larger firms trying to build scalable, repeatable business models based on selling many times over the core software deliverable will have, *should have*, if they don’t already, a much broader conception of product management of which the software development model is just one component. (And I would encourage those small I.T. consultants to go back and closely examine their product to look for ways to add additional value outside of the core software deliverable vis-à-vis of Levitt’s, Davidow’s, Moore’s and Bentzel’s work.)

We can also contrast the marketing needs of these two types of organizations, and the software they produce. One-off software projects or website roll-outs may not need launches and launch planning, but other products such as Enterprise software, packaged applications and networking gear typically rely on months of pre-planning leading up to a public announcement to be successful. There are key customers, analysts and journalists to brief, editorial deadlines to meet for contributed articles and product reviews, collateral to be printed for trade shows and so on. All of this depends on a “stake in the ground” set of key features by a date far in advance of the launch that sales, marketing and customers can count on, something you’re not going to see in an

Agile environment. Strategic customers expect a credible vision and roadmap for your products because changes impact their budget cycles and implementation plans – don't expect them to be as agile as your software team.

As the Agile trend grows, product management professionals, who have a moral and fiduciary responsibility to their products, need to be watchful for and resist the erosion of their discipline as their organizations consider adopting Agile development. For it should complement, but not replace, structured product management in order to deliver winning products.

Agile development promises many benefits, and organizations must continue to innovate not only in product but in process, and apply those innovations that deliver. I'm all for better code delivered faster, on time and with fewer bugs. But mis-application of Agile development from a misunderstanding of what it can, and cannot do, is bound to generate incomplete products, or worse, product failures. If Agile development is to continue to flourish, it's important that firms adopt it when, and where it fits, and ask no more of it than it can deliver.

David Howard is a technology marketing consultant based in Alameda, California. Find him online at www.up-periscope.net. Copyright 2008, David Howard. All Rights Reserved. Used by the 280 Group (www.280group.com) by permission.