

The Product Manager role

Of all the roles that play a part when software is created only one is essential: Coders. You can get by without Testers, either because the Coders are so good you don't need to test or through sheer bloody mindedness. And if you are very lucky the Coders will be able to work out what is required directly from the people who want the software. But generally speaking it helps to have someone decide what is needed from software before coding begins.

This is not to make a case for big requirements document – or indeed any documentation. Documentation may help, but it is only a medium for communication. The spoken word is another, and requirements are often better communicated by a conversation. The advantage of conversation is that it is a two way process. The listener can ask questions during a conversation and they can resume the conversation at a later date if they need clarification - both difficult to do with a document.

However the requirements are communicated, somebody needs to decide what they are. Scrum calls this person the Product Owner. It is the person (or persons) who decides what is needed, what is to be created, and what the priorities are. However Scrum has nothing to say about how the Product Owner finds or decides what is required.

Similarly XP has a role called the Customer. However, again XP has nothing to say about how this role is performed. On the original XP team the customers' role was to translate what the current system did so the C3 team could make the new system do something similar.

For any system of size, knowing what needs to be created and what the priorities are is a process of discovery. Unless a team is building a replacement system with no added functionality requirements, then requirements discovery is complicated and involved process.

Fortunately there are people with these skills and experience. In most organizations these people are either known as Business Analysts or Product Managers. Their role is to decide what needs to be built and communicate it to the developers.

Unfortunately the roles of Business Analyst and Product Manager are not always clear. More unfortunately still the role Project Manager is often mistaken for that of a Business Analyst or Product Manager.

Lesson 1: The term Product Owner is an alias for Product Manager or Business Analysts. In some organizations the Product Owner will be a Product Manager and in others they will be a Business Analyst. For the development team the net result is the same: a Product Owner with the authority and legitimacy in the organization to decide what needs to be done.

In this article and the next article I would like to try and unravel these roles and examine them in a little more depth.

Product Managers and Business Analysts are different

Basically, Product Managers work at companies that create software that sells in a market. They are outward facing. They know they need to seek out customers and find out what they need to make their lives better. Product Managers are concerned with competitor products and change outside the company.

In contrast, Business Analysts work at companies which develop software for their own internal use, or on specific developments for other companies which will be used internally. Thus they are normally found in corporate IT departments and external service provider (ESP) companies.

Business Analysts look inwards, they look at the operations and needs inside a company. They know exactly who their users are, indeed, in some cases there may only be one user. When Business Analysts look outside the company they are looking at suppliers as alternatives to development not as competitors in the market.

Lesson 2: The Product Manager and the Business Analyst roles are different. Not enough people appreciate the difference.

What does the Product Manager do?

The Product Manager role is summarised in Figure 1. The most obvious activity for the Product Manager is talking to the development team. This is a two way conversation during which the Product Manager tells the team what is needed.

Or rather, the Product Manager specifies what the goal is: they should not propose solutions, and neither should they get involved in technical design. However, they may review several proposed solutions and express a preference for one when the solution makes a difference to how customers relate to the product. For example, a Product Manager would not review alternative database schema designs, but they would review and comment on different user interface designs. One is visible to the final customer and the other invisible.

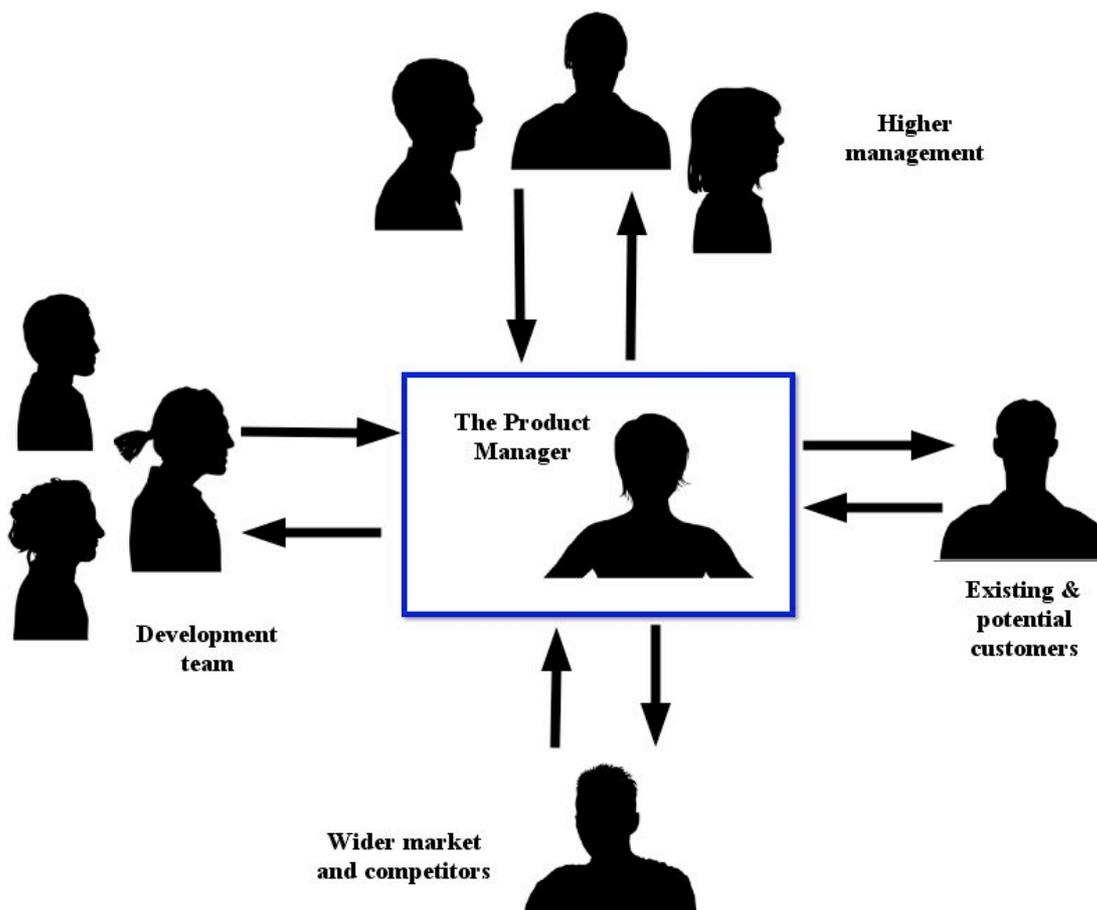


Figure 1 - Product Manager sit in the centre of many conversations

Lesson 3: Product Managers work with the development team, customers, senior managers, sales and keep watch on the wider market, including competitors.

Product Managers continue the dialogue about what is needed as the product is built. Developers come and ask for clarification, “Do you mean this? or that?” Whenever there is a decision that makes a difference to how the product functions to the customer Product Managers need to be consulted.

On products with a non-trivial user interface development teams should include a user interface designer. In matters of UI design and operation they will deputise for the Product Manager on UI decisions. However in the absence of a UI designer these kind of decisions should be made by a Product Manager rather than a developer.

In addition, Product Managers talk to the development team about what is technologically possible. Both about the development in hand at the moment, and about for future work. Product Managers need to reconcile what the technology can do with what customers need. To do this they need to stay abreast of technology developments.

It also means that Product Managers need to be in an ongoing dialogue with customers. Both customers who have already bought the product and are using it and potential customers, the kind of customers the company wants to have.

Product Managers need to visit, observe and research the customer market as a whole. They need to identify the problems customers have for which the software product is a solution. They need to understand how this relates to the customers tasks, problems and daily routines. Importantly they need to understand what will make the customer part with money.

There are various ways a Product Manager can do this. When customers are not known the first task is to find them. Once the customers are known there needs to be a dialogue. The Product Manager may ring them up and talk to them directly. Better still is a face-to-face visit.

Lesson 4: Product Managers need to be in regular contact with customers.

Traditional market research methods like surveys and focus groups are part of the Product Managers' toolkit. So too is *win-loss analysis*. Product Managers visit (without a salesman) customers who have bought the product, and potential customers who have not bought the product. The objective is not to make a sale, or turn around a failing one but to understand why one customer bought and another one didn't.

Product Managers need to look at the wider market and at competitors. They need to attend trade shows and read trade journals, watch competitors' websites and talk to customers of competitors.

Once information is gathered, Product Managers combine all it into product roadmaps and strategies. These they present to senior managers, but before they can do so they need to understand what senior managers are trying to achieve with the company. *What is the company strategy? And how does this product play a part?*

Lesson 5: Product Managers both follow corporate strategy and influence it.

There are even more tasks that naturally fall to Product Managers but are not so core. They may be asked to visit customers with sales staff to talk about the product, present product roadmaps. and listen to customer issues.

Product Managers are never very far from Product Marketing and are often the public face of the product. (Product Marketing is described below). They may be asked to speak at conferences or to the press, they may need to advise on how the product is presented in marketing literature.

Product Management in the UK

I spent the first ten years of my professional career working in and around London. I worked for a variety of companies but I never met a *Product Manager*.

Then I went to work in Silicon Valley for a bit. Here it seemed I met Product Managers all the time. Not just in the office but socially. There are a lot of Product Managers in Silicon Valley.

Since then I've made a point of understanding what Product Managers do, I've worked as Product Manager myself, I've been on Product Manager training and I've read a lot about what they do and how they do it. And I've become convinced that good Product Management is one of the key differentiators between successful software product companies and the unsuccessful ones.

Without someone who knows what customers value in a product and why they use it then all attempts to improve and enhance the product are just a shot in the dark. Success and failure come down to luck.

This person, whether they actually have the title Product Manager or not, needs the authority and legitimacy inside the organization to direct the product and guide its development.

This is vital for software products – software which people are expected to pay for directly or indirectly. Companies that create software for their own use, or develop bespoke software for customers who pay for the software to be written have a very different user base. When this software is delivered customers have little choice but to use it. The software has already been paid for.

There are lots of successful Silicon Valley software companies using Product Managers and acting as role models for new software companies. In the UK there are fewer successful software companies and fewer role models. While things have improved in the UK the Product Manager role is still not as widely known as it needs to be.

Time

What should be obvious by now is that there is a lot for a Product Manager to do. When the work is done well it really can make the difference between a big success and an also-ran product. What isn't so clear is that it's too easy for the role to be squeezed and become ineffective.

Squeezing happens for two reasons. Firstly, everyone in the company from CEO to Receptionist has an opinion on what the product could do, should do and what will make money. The Product Manager will have these feelings too but they need to find the facts to support their decisions. Only when armed with these facts can they make rational decisions and deny others their requests.

But getting facts brings us to the second reason for the squeeze: time. With so much to do Product Manager time is at a premium. Visiting a customer may involve two days of travel for a two hour meeting. This might not seem like an effective use of time but without these meetings the Product Manager will be blind to customer needs.

Lesson 6: It costs to acquire information; but without this information and facts money will be wasted elsewhere chasing guesses and opinions.

One solution is simply to have more Product Managers working on a Product. How many you need depends on the nature of your product, the number and type of customers, how new the product is and many other factors.

As a rough guide I recommend one Product Manager for every three to seven developers. If a product has been around for a while, the market is stable, and no big new innovation is planned then one Product Manager can probably keep seven developers busy.

If however the product is new, the product is innovative, the market is developing rapidly, and there are many needs to be addressed then one Product Manager for three developers is more realistic.

Lesson 7: Appoint one Product Manager for every 3 to 7 developers. Without a Product Manager to guide them a team will be guessing, success will be based on luck.

When a development team is larger and multiple Product Managers are needed then things become more complicated because different Product Managers must co-ordinate their work. One answer is to have a Tactical and a Strategic Product Manager – TPM and SPM respectively.

The SPM does most of the customer visits, most of the conversations with management and does the long term roadmaps. The TPM spends more of their time working with the developers, helping sales and the near term roadmaps. Importantly the SPM and TPM talk regularly, they should sit next to each other. This arrangement also makes it easier to arrange visits to customers and debrief afterwards. (By now you may have realised that Product Managers need to travel a lot.)

MRD-PRD model

Within product management circles there is an accepted way of creating requirements documents. Of course organizations and even individuals differ in exactly what they expect from each document, and what they say in each document but as a general rule it goes like this.

First it starts with a Business Case, or Business Requirements Document - a BRD for short. This outlines the business opportunity and how it might be exploited, and what the return might be. For example, it might say: “Telesales staff (who have the problem) waste a lot of time calling to customers who do not wish to receive telesales calls (the problem to be solved); a product which automatically prevents numbers on the national “do not call” list being called (solution) would increase productivity (benefit).”

The BRD might also give some indication of market size, number of potential customers and so on. Sometimes there is no need to write a BRD, perhaps because the product is a development of an earlier one, or because the business case is implicit in the companies purpose.

Next comes the Market Requirements Document, the MRD. This is the document that examines what the market needs. The MRD will take any BRD as a starting point and develop the ideas further. It discusses the problems a product would need to address, who would buy the product, makes some suggestions on the functionality needed, what the performance characteristics would need to be and examine the potential competition.

In some cases the BRD may be merged into the MRD forming the first few pages of the MRD. When it exists as a stand-alone document it should be a short document.

Neither the BRD or MRD addresses the features of the product in depth. The MRD might say “User access needs to be controlled” and might discuss current market standards but it would not go into detail. The MRD would also lay out the constraints on the product: “Needs to sell for less than \$100 to be competitive.”

Next comes the Product Requirements Document. The PRD translates the requirements in functionality into features with essential details. The PRD may also refine statements on the performance and constraints.

Again, sometimes the MRD and the PRD get merged together. In my view this is a mistake because the MRD should focus on the need, and the PRD is the start of the solution. While the BRD (if it exists) and MRD should be created by a the business side (Product Manager or higher executive) the PRD is the start of the engineering process. As such the engineering group should contribute to, or even write the PRD, with the Product Manager.

What happens after the PRD is less well defined. Sometimes the PRD is enough to get started on. Other times the PRD may be further refined as an functional specification – sticking with the convention this would be called a Functional Requirements Document or FRD. Software developers might like to respond to the PRD with a design document.

If this all sounds very waterfall-ish that's because it is. One document leads to another and eventually some code gets written. However the idea of separating the market need and business case from the product requirements and the functionality to meet those needs holds in iterative and evolutionary models. These documents need to become living documents. Market needs change and so the MRD can be expected to change over time.

Document creation needs to overlap. The BRD kick-starts the work with a small team, Product Managers continue to develop the MRD and engineers respond with PRD changes or directly in code. As the scale of the work becomes apparent the team may increase in size.

In-bound versus out-bound marketing

Another way to refactor the Product Manager role is to ensure it does not involve any outbound marketing. Strictly speaking this is the role of Product Marketing.

In the purest form marketing is about both discovering customer needs and communicating solutions to the customer. Discovering needs is in-bound marketing, its about finding out what is needed. Once there is a solution available the focus is then on communicating about the product. This second form of marketing is what most people think of as marketing but strictly speaking this is out-bound marketing and is known as Product Marketing.

Product Management is at heart an in-bound marketing role. However the role is often caught up with out-bound marketing, communicating about the product. This is particularly true at small companies who might not be able to afford an additional person. As a company grows these responsibilities should be passed to a dedicated Product Marketing Manager.

The Product Marketing function, often filled by Product Marketing Managers, is concerned with communicating to customers that the product exists, the benefits of the product, and changes to the product. This is done through advertising, public relations, press releases, online websites and other media.

Lesson 8: Outbound Product Marketing is a different and distinct activity from inbound Product Management.

(Just to complicate things, Product Manager as described in this article and as practiced by successful software companies is different to the Product Manager found in many non-technology companies. The role of Product Manager at a company like Proctor & Gamble is an out-bound marketing role, one usually involved with brand management.)

A Product Manager cannot be a Developer

A Product Manager needs to be technically knowledgeable, they need to understand what technology can and cannot do and they need to understand their products. Thus it is not uncommon to find Developers moving into a Product Manager role. However anyone taking this route must accept that their coding days are behind them for several reasons.

Firstly, perhaps unsurprisingly: time. As already outlined the Product Manager role is a full time role. It is wrong to think a Product Manager will have time to talk to customers, decide what is needed, survey the competition and so on, then change hats and write the code.

That said I have come across developers who, in the absence of a Product Manager – take on many of the duties. Often this happens unofficially, and often the developer doesn't do the complete role. While this is understandable it is a sign that there is a role to be filled.

Lesson 9: In the absence of an official Product Manager others are likely to fill the role.

Secondly the priorities of the two roles conflict. Good developers have an empathy for the code base and the product architecture. The code speaks to them. It says things like “refactor me” and “add a database abstraction layer.” Good developers hear these messages and do their best to give the code what it wants.

Product Managers also have a relationship with the product but their empathy needs to be with the customers. The messages they hear are “simplify the UI” and “Give me the product on Oracle”.

Asking one person to keep all this in their mind, and empathise equally with customers and code is too much. It would be like asking one person to present two personalities.

Anyone who tries to fill both roles will inevitably tend towards one side or the other. For a Developer moving into product management they continue to listen to the code when they should be listening to the customer.

Conclusion

Although Product Managers have the word “Manager” in their title they are not managers in some the senses of the word. They manage a thing not people. Their power rests on their legitimacy and knowledge rather than their direct authority.

Without a Product Manager directing the direction of development the success of a commercial product is down to luck and chance. Product Managers are the people who take the luck out of developing software products.

Product Management is not a misunderstood role, it is simply an overlooked role. Too many software companies are either ignorant of what good product management can do for them or they simply believe that developers know best.

Lesson 10: If you work at a software company which sells its products – either shrink wrapped or online via the web - to multiple customers then you need Product Managers working with the development team.

Further reading

Unfortunately there is still a lack of good books on product management. Any aspiring Product Manager should certainly read *The Inmates are Running the Asylum* (Cooper 2004) and *Crossing the Chasm* (Moore 1999) – and probably the sequel, *Inside the Tornado* (Moore 2005) is also worth reading. Clayton Christensen's *Innovator's Dilemma* and *Innovator's Solution* are also to be recommended (Christensen 1997; Christensen and Raynor 2003).

Although I've not read *Tuned In* (Stull et al. 2008) and *Beyond Software Architecture* (Hohmann 2003) I've heard good reports about both. Some background in marketing and business strategy – especially in technology – is also a good idea.

Over the last five years I have been writing a set of patterns about the software business. Many of these patterns relate to the product development process and product management function. For example, the patterns include Single Product Company, Product Roadmap (Kelly 2008) and Same Customer, Different Product (Kelly 2007). These patterns and some more can be found at <http://www.allankelly.net/patterns/business.html>.

Fungible – a correction

Back in the first article of this series I used the word fungible, I described it like this: “Money is, economists like to tell us, fungible. Which is another way of saying it can be exchanged for other things very easily. Money can be exchanged for resources such as a new developer, thereby increasing our resources.”

Actually I got it wrong. My Oxford English Dictionary says: “fungible ... replaceable by another identical item.”

So while a £20 note is fungible – one £20 is the same as another – the exchanging of the note for 15 minutes of developer services is not. Exchanging money for services or good is an example of liquidity.

I was trying to convey the idea that because money can substitute (via liquidity) for many things there is no need to consider different type of resources.

Apologies to all and thanks Edmund Stephen-Smith for point out my mistake.

References

- Christensen, Clayton M. 1997. *The Innovator's Dilemma*. Boston, Mass.: Harvard Business School Press.
- Christensen, Clayton M. and Michael E. Raynor. 2003. *The Innovator's Solution*: Harvard Business School Press.
- Cooper, A. 2004. *The Inmates Are Running the Asylum: Que*.
- Hohmann, Luke. 2003. *Beyond software architecture : creating and sustaining winning solutions*. Boston: Addison-Wesley.
- Kelly, A. 2007. "More patterns for Technology Companies." In EuroPloP, eds. L. Hvatum and T. Schümmer. Irsee, Germany: UVK Universitassverlag Konstanz GmbH.
- Kelly, A. 2008. "Business Strategy Patterns for Product Development." In EuroPloP (European conference on Pattern Languages of Program design). Iresee, Germany.
- Moore, G.A. 1999. *Crossing the Chasm*. Capstone publishing.
- Moore, G.A. 2005. *Inside the Tornado*: Collins.
- Stull, Craig, Phil Myers and David Meerman Scott. 2008. *Tuned in : uncover the extraordinary opportunities that lead to business breakthroughs*. Hoboken, N.J.: J. Wiley & Sons.