(Chapter ONE)

Introduction #

(PROOF COPY -- Not for distribution.)

Why

Chapter 1 - Introduction

My goal in this book is to give you a basic working template - data and process - for managing product management that you can implement quickly and easily and start using immediately. I have been a product manager for nearly 20 years, and for seven of those years my customers were product managers. I've had a lot of opportunity to think about product management as an art and science, to gain a theoretical view of product management, you might say. I wish I'd had a book like this in my first years as a product manager, and since I'm now in a position to write it myself, I hope others will find it valuable.

This book is primarily a cookbook. It has information and guidance you can use out of the box, but with enough background that you know why it works, and how you might go about altering it to your own needs.

One thing I've learned as a product manager, and you can consider this a rule of thumb:

People don't buy your product for its capabilities, but for *your* knowledge. They consider you an expert in the domain, and they want to buy that expertise, not just the capabilities that support the expertise.

With that rule in mind, I don't want to give you a list of "some things you can do." I'm the expert here, so I'm giving you knowledge - this is a fleshed-out template that works as-is. You can use the template as it's presented here, and you will be better off and more successful versus not using a template.

I explain why the template is formulated the way it is, because that's how you learn, and how you get power to make the template even better suited to you. But if you don't want to learn anything right now, just get going, then simply apply this template (go to Chapter 2) and your product management process will start working better.

There are other templates or processes you could use - this one is not the be-all and end-all. It has no intrinsic superiority to other good quality product management templates out there, aside from the fact that it exists in this book. (Of course, there are *worse* templates out there. For example, scrum templates are not good for the product management team, although they are basically workable for the development

team.)

"Your Obvious Is Your Art"

A lot of what I say in this little book may seem obvious to you. If so, that's fantastic! If something seems obvious to you, but other people seem to think it's a big deal, that's a sign that you are in your area of expertise. Your obvious is your art, and you should take advantage of it.

# # #

Goals of Product Managers

Practical goals

As a product manager you have a set of *practical goals*, which are defined by your organization and management, and to some degree by the job itself. If someone asks, you probably describe your responsibilities along these lines:

- Get the product out on time

- Make sure the product has the most important features in it

- Ensure the product launch is successful

- Give the sales team the correct tools to sell the product

Those are part of your official job description. They are public, and not controversial, and everyone will nod their

heads sagely when you list them. And if you use the template in this book, it will be easier to achieve these goals.

Personal Goals

But you also have a set of *personal goals*, which you probably *don't* share publicly. They drive a lot of the work you do. These goals might include (and I'm just listing some of my personal goals here):

- I don't want to look like an idiot because I don't know the status of my product's development, or its launch plan
- I want to kick some ass with this product and change the world
- I need to be able to answer any question about my product
- I need to be able to justify any decision I've made
- I want to have insights into my customers needs and how they combine to create value
- I can't be blindsided by any bad news from other organizations like engineering or marketing
- I want to be a superstar, a product ninja

Research has shown that for job satisfaction it's much more important that your personal goals be achieved than your practical goals - indeed, your practical goals are only possible if your personal goals are met.

The good news is that this template will enable you to achieve your personal goals as well as your practical goals.

# # #

Goals of Having A Product Management Process and Tooling

The Benefits of A Process and Tooling

A good product management process and associated tooling provides many benefits for an organization:

- You will have better collaboration between all stakeholders
- There is a lot less "duplicate" (but slightly different) information floating around in emails and files
- You have a way to capture enterprise knowledge
- You can "mistake-proof" the product definition process
- You will be able to make much better product decisions and tradeoffs
- Decision-making will be easier and more transparent

But a process and tooling also enable many personal benefits for you. I already mentioned your personal and practical goals. But there's much more, and it has to do with how a good system can augment your capacity *and* your intellect.

Augmenting Yourself

Like all knowledge workers, as product managers our most
precious resource is cognitive capacity, and we lose it
constantly throughout the day. If you're like me, you probably
end up handling 15-20 different activities and interruptions
every day. Each of those requires a mental context switch, and
context switches are expensive. Anything that makes the
context switch easier will make your life as a PM easier. A
good process, and the associated tooling, is one of those
things.

There are three main ways process and tools help product
managers out with the cognitive load problem (and help you
seem smarter - thereby achieving a personal goal!):

- Offloading memory

- Enabling deeper analysis and investigation

- Offloading process steps/status

Offloading memory

The less I have to remember - why a particular requirement is
important, who suggested it to me, who made the most recent
comment about it, even the reason we're scheduling the next
release - the more cognitive capacity I have left for doing

the more interesting parts of the job, like designing new features, creating marketing positioning, or talking to customers.

As Amy Hoy (http://unicornfree.com) says:

- Almost all productive people are far too busy to remember everything they do each day because they're Getting Shit Done.
- Almost all people are numb to their own pain.
- Their most dangerous problems aren't the minor irritations that sting, but the dark shadows that lurk below the surface, unsaid, unnoticed, unmanaged

Enabling deeper analysis and investigation

And it's not just your memory that benefits, but your intellect as well. The information in the system is easy to retrieve, so it eases the load on my memory. But it's also available for analysis and investigation that I *can't* do if the information is all in my head, or distributed around lots of spreadsheets and emails and Word documents. That means I'm *smarter*. Even a simple and obviously valuable query like "what features were requested by both customer X and customer Y?" is typically beyond the capacity of a normal person's - even a PM's - memory.

Offloading process steps/status

We each have particular aptitudes and inclinations. Those
enable us to do certain product management activities
automatically, but they may also lead us to forget about other
product management activities. For example, I'm very good at
thinking about how a feature might be used, but I often forget
to consider the impact on upgrades when working on a new
requirement. A good process and tooling can help me remember
about upgrade impacts (or help you remember about usage
scenarios). I implement this using what I call "Impact Areas"
in various tools I've used. I'll cover that technique later.

Another example of offloading process steps is the launch
process. If you are developing your prodcut in a more or less
traditional manner, you spend months building it, and then a
few weeks launching it. Every few months you have to remember
how to do a launch, remember what went wrong (or well) in the
last launch, and decide how you're going to do it differently
this time. Obviously, a system that not only keeps a list of
the steps required for launch, along with who is responsible
for each, *and* a history of what happened last time, means you
don't have to invent this every time, and you can learn from
history.

Enough Theory

The rest of this book is divided into three main parts - the data, the process, and then some items that didn't fit into the other two sections.

Let's be done with all this talk, though, and get into the details of the template.

< < < < < > > > > >