



DEVELOPER

Lorna Mitchell & Ivo Jansch

N Ways To Be A Better Developer

Lorna Mitchell and Ivo Jansch

This book is for sale at <http://leanpub.com/nways>

This version was published on 2016-12-30



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2013 - 2016 Lorna Mitchell and Ivo Jansch

Tweet This Book!

Please help Lorna Mitchell and Ivo Jansch by spreading the word about this book on [Twitter](#)!

The suggested hashtag for this book is [#nwaysbook](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

<https://twitter.com/search?q=#nwaysbook>

Also By **Lorna Mitchell**

[Git Workbook](#)

[Zend Certification Preparation Pack](#)

Contents

Introduction	1
From Keynote to Book - Ivo	1
Sharing Ideas - Lorna	1
Learn to Speak Manager	3
Simple words - Lorna	3
Teach!	4
Educate yourself by educating others - Ivo	4
Can I explain this to you? - Lorna	4

Introduction

This book is a work in progress; perhaps it always will be as we certainly have no end goal in mind. It's an idea that began, and we don't know yet what it will turn out that we have made. The book will continue to grow and change as we edit, update and add many more tips to it in the future. If there's anything you'd like to see us include, tweet it with the #nwaysbook hashtag and we'll give it our full consideration!

From Keynote to Book - Ivo

At the end of 2010 the organizers of a developer conference approached us to do the opening keynote at their spring 2011 conference. They were looking for a keynote that would inspire developers and help them become better at their jobs. This request was well timed. Both Lorna and I had left our jobs at a web development agency a couple of months before that to work on new and exciting things, and we both had an interesting career progression to look back on. Lorna over the years had grown from a developer to a technical team lead and eventually community manager. I had grown from a developer to software architect to CTO of a web development company. What we have in common is that we're both developers, conference speakers and book authors. This allowed us to reflect on the various lessons we had learned along our career paths, so we started the keynote preparation by listing our lessons learned.

Since the conference required a title and abstract way before we even had a list of lessons learned, we somewhat randomly decided we would cover 27 topics in the talk, so the keynote became '27 Ways To Be A Better Developer'. I can't remember how we arrived at that seemingly arbitrary number but in hindsight I'm glad we didn't pick 42!

The keynote in 2011 was well received, and we've frequently been approached by people who told us they had taken away something from our keynote. In 2013 we were asked to do the keynote at another conference. With 3 years of additional experience there was so much we'd like to share on top of that original keynote, but we also realized that cramming 27 topics in a keynote was a challenge, so we decided to do less, not more, and came up with '0x0F ways', trimming the keynote down to the 15 topics we had received most positive feedback on.

We think all of the other lessons from the original 27, as well as the new ones we planned to add, would still be valuable to developers, so the idea of converting the keynote to a book was born. Because we have no idea how many topics we'll end up with over time, we decided the book needs to be flexible, hence the title 'N Ways To Be A Better Developer'.

Sharing Ideas - Lorna

I first met Ivo when he interviewed me for a job that I was hoping to land with a PHP consultancy. I thought I was out of my depth, but I got the job ... years later Ivo told me that he knew that he'd

given me a mountain to climb; my skills weren't really ready for it at that stage but I survived somehow. Those first years in that job were easily the most rapid career progression of my life, I learned and grew so much, prodded along at intervals by Ivo, and adding plenty of good practices of my own as I went.

Fast forward a few years and we're now running our own businesses and have a great deal of experience to draw on - not just as developers, but also as leaders, evangelists, project managers, business owners, authors, speakers ... the list goes on. What started life as sharing a stage with a friend has grown into this opportunity to share all our ideas and advice with anyone who wants to read the book. Some of the tips are very short, some take a bit longer to explain, but together we have written the book that I needed back when I took a job I wasn't quite ready for. I hope that for you, the book has found you at the time that you need it.

Learn to Speak Manager

Simple words - Lorna

As developers, we don't typically think of ourselves as being "a people person" (I'm generalising of course, but I include myself in that generalisation), but this next recommendation needs few skills and will make a major difference to your career and ability to work with others. You need to learn to communicate with that particular breed of colleague: the manager.

Adapting your communication style as appropriate for the setting and your audience is always important. Managers in particular need to be spoken to in words that they understand (i.e. short ones). The main difference between a manager and someone who is focussed on making a specific technical thing happen is priorities. Developers care about making a thing happen, because things are important and they like to execute well. Managers care about budgets, because that is what they are responsible for. In general, you can go a very long way by adding "revenue" as every 7th word in whatever you say to a manager (yes, that's my official professional advice!)

Managers will have no idea why you would want to know more technical things (they will also try to offer more money to a developer leaving an organisation because her job is boring, which makes complete sense to the manager and no sense at all to the developer). However if you explain that this additional knowledge will allow features to be delivered more quickly (increasing revenue), improve the quality of the code (reducing bugs and therefore increasing revenue) and allow the team to take on more complex projects (that will command more revenue), then the manager is much more likely to see your point of view and pay your expenses for attending a conference. Both the manager and the developer in this story have valid points of view, and both are important to have present in an organisation, but being able to translate between the two is a skill.

Developers can seem strange to anyone who has never been one, so being able to explain what you are doing and why is very important. If you move into a client-facing role and come into contact with customers when doing meetings or presales, this becomes even more important. Developers care about technology, about solid solutions, and about getting things done. Managers care about getting things done on time and on budget. We need both in every organisation.

Teach!

Educate yourself by educating others - Ivo

You might think that ‘to learn’ is the best way to become better at something but I find that ‘to teach’ is often even better. It’s one thing to dive into a topic and know everything about it, it’s an entirely different thing to pass on that knowledge to others. Teaching others requires a more in-depth knowledge of a subject. Making a topic understandable to others will help you structure your thoughts.

Teaching can be done in a number of ways. You can of course give someone a training course, but that might not always be practical. You can however also teach by writing about a subject, for example a blog post, or you can teach by presenting a topic at a conference. Having to present something to an audience has a number of effects on you: first, you’ll have to research the subject in order to create a compelling presentation. Second, you want to anticipate questions that the audience may have, causing you to dive into the subject even more. Stage fright will no doubt add a little pressure to know a subject intimately, but even without that fear - just having to break down a topic into something that you’re able to teach someone else will be incredibly helpful in your own understanding of a subject.

Can I explain this to you? - Lorna

Being able to express ideas is a step beyond knowing what to type. To know if you truly understand something, try explaining it to someone else. This someone else might be an interested colleague, someone you meet at a user group, or even a very patient bystander (my non-technical sister once spent a long motorway drive getting OOP explained to her, for which I’m still eternally grateful). That moment of trying to get all the ideas lined up so that you can explain them in a logical fashion will always further your understanding of any subject. I often think that this technique is linked quite closely with the “confessional debugging” idea that I mentioned in another chapter - say it out loud and your ideas begin to crystallise.

Whether you’re teaching in a formal sense, or writing a blog post about a new tool, being able to support someone else in their curiosity is a skill. You need to give enough information to be interesting, but not enough to drown. Some concrete examples will help, but only if the person you’re aiming at can understand and relate to those examples. If you’re teaching someone in person, then you may also need to observe and gently correct some elements of their own practice in this area.

To be able to do all of this requires a depth of understanding that is beyond that of a mere user. You may not feel like a teacher when you email your colleagues telling them about a new tool you found, but that’s essentially what’s happening here. And if you do teach in a formal setting, then you stand a good chance of getting questions that will probe the boundaries of your knowledge on any topic (pro tip: never be afraid to say “I don’t know, but I’d like to. Shall we investigate

after the session?” - you’ll learn so much!). Whether you speak, write, or help out a buddy at a user group, try to look out for opportunities to teach, and grab them with both hands. As the saying goes:

“If you become a teacher, by your pupils you’ll be taught.” — Oscar Hammerstein II