Books to Read

- Clean Code: This is one of the first books I ever read. It's a very good guide to what good code looks like. You guys don't really NEED to read this book, but I'd ready it anyway. You can get some good ideas from here.
- Clean Coder: This is an extension to the previous book. This book should be taught at Universities since it talks about how to be a professional and not just a developer. This is the book I'd start with.
- The Software Craftsman: Kind of a continuation from the Clean Coder. The previous one is more about principles while this one is a bit more specific with examples.
- The Phoenix Project: This book is AWESOME! It's not really a technical book, more of a novel based on a company having troubles with their delivery, and how they use DevOps principles to fix their problems. I**f you had to read only one book from this list, this should be it, it really opens your eyes.**
- The DevOps Handbook: This is the more technical side of the previous book. It's written by the same author.
- Building Evolutionary Architectures: This one is quite new and it describes ways in which you can structure your architecture to enable change easier. Very good book.
- The Five Dysfunctions of a team: Very good book which is also kind of a novel describing the story of a CEO that has just been hired to fix some wrong behaviours in the company. It's really good describing the different personalities in a team and how to approach them.
- Coaching Agile Teams: Describes how to go from being the "Teacher" of the team to the "Coach" of the team, so basically from telling people how to do things to getting them to do it themselves sharing the same views you may have. Great book if you don't know where to start when trying to become the leader in the team.
- Production Ready Microservices: Describes how to define what a Microservice should look like. The author goes through some examples of things they've been added to their services for better reporting, etc. I got some great ideas from this book.
- Microservices: Flexible Software Architecture: Great book that describes Microservices in an easy way with some Java/Spring code to go along the way. I really recommend it.
- Agile Testing: What it should be like to be a tester in an Agile Team. You should really keep in mind their view on things since it will help you work with them. It talks a lot about how you should automate repetitive things, etc.
- More Agile Testing: Extension on the previous one, but taking a look more from a Team perspective, and how the team should have a testing mindset. Very good!
- Soft Skills: The Software Developer's Life Manual: This one is more of a personal development/lifestyle point of view. It talks about A LOT of different things like how to become a good developer, how to save money being a contractor...very entertaining.
- Peopleware: All about building teams from some people who worked at Google.
- Implementing Domain Driven Design: Some examples of how to write code using Domain Driven Design. I think it's a good overview, though I wouldn't focus too much on the specifics.
- Building Microservices: This is the first book that described what Microservices where (he came up with the term). It's an awesome book and you guys should watch some of his presentations, he's got some very interesting ideas.
- Growing Object-Oriented Software, Guided by Tests: Awesome book that describes how to do TDD at different levels. I read it a bit too early in my career and I think that was a mistake, but you guys are in a position where you'll understand it much better. It's considered the best TDD book out there.
- Success in Programming: Talks about how to build your career around programming. A little bit of everything, but mostly about how to build your brand.
- The Passionate Programmer: Good tips about day to day things.
- How Google Tests Software: A very interesting read looking at how Google structures their teams and how they make sure they are able to test their massive codebase.