



Notes to a software team leader

Roy Osherove

Download now

Read Online ➞

Notes to a software team leader

Roy Osherove

Notes to a software team leader Roy Osherove

Why?

Many software team leaders today face a weird problem: they believe in a bunch of great things that software developers should be doing but somehow can't seem to make this work at their own team.

For example they feel that shorter iterations can be good for the team and customer, but no one seems to believe them, or no one cares when they mention it.

They believe that unit testing can save the product from its poor quality, but their team members would rather do anything but write those tests.

Team leaders everywhere seem to be hitting a wall everywhere they try to make a difference – so they give up, or they get things done half-way to the point of failure.

People.

Driving “the new stuff” into the team is usually the least of the worries of today's software team leaders. Team leads usually have little to no idea how to handle people related issues – issues that affect how the morale, quality of work, and overall performance of the team, and of course impacts how easy or hard it is to implement “the new stuff”.

Most team leaders are clueless as to how to handle their manager giving them an impossible due date, a team member reluctant to try anything new, or another team member teaching all the other members practices from 25 years ago that today only hurt the team.

Why?

No one teaches that to software team leads. Team leads today, in the overwhelming majority of places, are just developers who worked hard and stayed with the company long enough to be promoted. But they have no people or management skills - and those are very painfully needed when you are trying to drive the things you believe in inside an organization that has very little interest in changing.

Team leadership is the next big thing that software developers need to conquer, or none of this unit testing, TDD, Agile or Lean thing is going to catch on, except in very small circles, that, by chance, happen to have the right people leading their teams.

Notes to a software team leader Details

Date : Published March 1st 2012 by Roy Osherove


ISBN :

Author : Roy Osherove

Format : ebook 100 pages

Genre : Business, Management, Leadership, Nonfiction, Computer Science, Programming, Science,
Technology, Software, Self Help, Engineering

 [Download Notes to a software team leader ...pdf](#)

 [Read Online Notes to a software team leader ...pdf](#)

Download and Read Free Online Notes to a software team leader Roy Osherove

From Reader Review Notes to a software team leader for online ebook

Dave Bolton says

I read this last year and then made all my team leads read it too. It's a really good overview of what it means to be a team leader, and how to do it. It is succinct, which is important when you're trying to persuade other people to read it, and it works from first principles, so you don't need to refer to a lot of other books to get value from it (again, important when you're looking to have others read it).

I also liked that it gave an outline of a simple "Agile" way of working without ever mentioning Agile - bonus marks for that. Specifically getting into Agile would have been opening a can of worms that would've been unnecessary for readers to get value from the main messages of the book.

Abdul Khaled says

3 stars as i found the book too short, found it helpful nonetheless

Ashok Sahu says

This book is just awesome. Had read Roy Osherove's Unit testing book earlier and once I saw this book mention became very curious to explore. Spending time with this book was really worth.

It's a must read for any software technocrat or leaders (new-bee TLs, experienced TLs, architects, managers and of course the developers) who works with an organisation or team. Everybody would have something the other very important to gain from it. It shows how and why we should be pushing our bars, need of it, purpose of it and the benefit of it...

Roy speaks about what is the importance of a learning plan for every team or the individual team member(s) and how important is this to be realised by their TL/Managers and to provide the required support and encouragement. The whole book is divided into **Elastic leadership, survival phase , Learning phase, Self organising phase**. The 3 phases of a good team which is very much required for the survival. On top of that there are around 25+ advices with some of well known leaders/technocrats/gurus like Uncle Bob Martin, Gary Reynolds, Patrick Kua....etc. which covers up the above bold styled values and explains them still better.

Giustino Borzacchiello says

I really liked this book: the idea behind the "elastic leadership" is something that I've never thought of and it will be something to think of going forward.

With hindsight, I recognize all the phases referenced in the book, and how I failed to deal with them.

Jascha says

I was pretty much excited to read this book, so much that I've played some tricks and got it on top of the Want to Read queue. Now that I have finished it, I must admit that I have felt distant from the author and its stories. It's not that I disagree with his ideas or thoughts. It simply did not give me the emotions *Becoming a Better Programmer*, a book that I have recently finished, was able to give me. I'm talking about emotions, not content.

Notes to a Software Team Leader is about stepping up and leading your own team. Leading, not bossing. It's about understanding its strengths and weaknesses and getting its members to grow. It's about realizing what phase the team is in and taking the proper decisions in the short term, aiming, in the mid-long term, to grant it time to learn and get better.

The book is divided into two parts: in the first, the author shares with the readers his experiences and thoughts; the second is a collection of short advices written by other people, each providing a guideline, an insight.

Again, it's not that I do not agree with the author. I didn't get entertained. I didn't write down a single note (well, this is a lie. I've taken down one. See below). I didn't add it to my favorite books. I don't feel that reading this book resulted in adding value.

So, it was not completely true to say I did not write down anything from this title. This is the only sentence that got to my soul:

People who come across my path, even for a short time, will get the same amount of respect, expectations, and challenge from me as if they had been there forever. I try to always leave people whom I've led better off than they were... It's my personal integrity to do so.

I did not read enough books on Team Leadership to make a proper comparison. But I don't feel recommending this title.

As usual, you can find more reviews on my personal blog: <http://books.lostinmalloc.com>. Feel free to pass by and share your thoughts!

Damir Arh says

I'm pretty sure the issues about software development that the author brings up will be very familiar to you if you have any experience as a team leader. But the main value of the book is not in bringing them to you attention; it's in its call to action: inviting you to do something about them instead of passively accepting the state you're not satisfied with. It leads you through the author's vision of steps required to reach the ultimate goal of a growing and learning team striving to always do their best job possible.

The second part of the book consists of contributions by other authors. Instead of being directly related to the first part, their short notes take different, often thought provoking, aspects to the job of a team leader: usually presenting a potential issue and giving a suggestion how to solve it. They are a nice extension to the core

book and give a lot of value on their own as well.

I can definitely recommend reading the book to any team leader, no matter how much experience he already has in his job. Even if you don't agree with everything, I'm pretty sure the book will act as an eye opener and make you more aware of stuff you already take for granted, although there might still be opportunities for improvement. You never know, it might even make you a better leader.

Jake McCrary says

Pretty good book. It is split into two halves. First half it written by Roy and is his advice for leading teams. Second half is a collection of short essays (1 - 3 minute reads) written by others offering their advice.

Roy's advice seems pretty good. I can see fragments of his advice in some of the better leaders I've been around.

The gist of the book is that there are three different modes that a team can live in; survival, learning, and self-organizing. Each phase requires a different leadership style.

There was also some interesting thoughts around commitment language.

One annoying aspect of this book was the numerous typos and formatting issues. If doesn't happen enough to make the book unreadable. When it did happen it was jarring.

Benjamin Reitzammer says

Very good! Gives a lot of of inspiration and a clear framework on how to lead software development teams.

Claudiu Constantin says

Great introductory book for a new teamleader, full of non-nonsense advice.

Igor Afanasov says

This book contains lots of valuable thoughts. Most of them are not rocket science. Some of them you already now. Some of them you'll know just after you'll stop you day to day routine, seat and think about your management and your team. Some one them could've been acknowkedged with pain.

"Notes to a software team leader" helps to understand better things you already now, think about things you don't kn ow well and gain some new knowledge.

For most developers becoming a team lead is nervous thing. This book helps to realize: becoming a team leader is not that dangerous. It's interesting. Really.

That Guy says

Okay book, but not really relevant to my business. Helps me better understand (a bit) how larger software teams (dys)function. Basically, it emphasizes that software leaders scale their teams by teaching, empowering, and leading, not by doing (a pitfall that many newly promoted team leads fall into). I found the first half of the book (which was written exclusively by the author) to be well thought-out and applicable. I particularly liked how the author tied together three team and time phases with appropriate management styles for successfully navigating each phase. On the other hand, I found the second half (a collection of blog posts from various people) to be shallow, unhelpful, and skip-able.

Kirill says

"There are no experts. There is only us." - the motto from one of the first chapters is fascinating! There is a good piece of battle proven wisdom in these few words. Same true for the rest of the book, liked it very much!

Roy writes in the first place about importance of learning and growing people, which are undoubtedly essential principles in engineering teams. Maybe I am lucky, or maybe the industry has changed over the course of the last years - not least due to SW Craftsmen like Roy - but in most of the teams where I happen to work, the power of investment in professional skills was undisputed truth. Sometimes too undisputed, as the learning just for the purpose of learning could be seen now and then.

The last chapter - collection with a dozen of short essays from partially very well known Authors - feels unfitting. Although every single advice is very valuable, I am still missing the structure of first chapters.

Allison says

I liked what he had to say about software culture - it is up to us to create a culture of growth and quality, and team leads bear a large part of that responsibility.

I am not sure how worthwhile it is to spend as much time as he did defining the "three modes" of a software team - in reality, it is all a spectrum.

But all in all, there were enough nuggets in the book to make it worthwhile. I especially liked the part about "commitment language" that encourages people to think carefully about what they commit to (say "I will do..." rather than "I will try to" or "I might"; but also only commit to things that are within your control - say "I will work on this bug for 5 hours today" rather than "I will fix this bug today")

I was a little annoyed by all the typos and formatting problems - perhaps this comes with the territory for self-published books?

Farhodjon says

There are just notes. Helpful notes for a technical team leader.

Ingvald says

i listened to the audio book, and liked it. it had a lot of good advice, both from Roy and the guests in the second part.

why i didn't rate it higher: not much new info or advice for me. and although advice i disagree with isn't necessarily bad if it makes me think, the topic of commitment (as in commitment language) was an example of the opposite for me. even if clearer language often is better, visualization and transparency of work (inspired by kanban, supported by science) is much better to improve what the book argues that increased commitment will improve.

biggest reasons i liked it: a lot of good advice relevant for a team leader gathered in one place. even if not much was entirely new to me, i have only seen it scattered around in a lot of other contexts before. also, Roy's own stories and experiences makes it easy to relate to.
