



# Joel on Software

*Joel Spolsky*

Download now

Read Online ➔

# Joel on Software

*Joel Spolsky*

## Joel on Software Joel Spolsky

Someone once said that the task of a writer is to "make the familiar new and the new familiar". For years, Joel Spolsky has done exactly this at [www.joelonsoftware.com](http://www.joelonsoftware.com). Now, for the first time, you can own a collection of the most important essays from his site in one book, with exclusive commentary and new insights from joel.

## Joel on Software Details

Date : Published October 28th 2004 by Apress (first published January 1st 2004)

ISBN : 9781590593899

Author : Joel Spolsky

Format : Paperback 384 pages

Genre : Computer Science, Programming, Software, Nonfiction, Science, Technology, Business, Management



[Download Joel on Software ...pdf](#)



[Read Online Joel on Software ...pdf](#)

**Download and Read Free Online Joel on Software Joel Spolsky**

---

# From Reader Review Joel on Software for online ebook

## Alexander Borshak says

Not bad, worth to read. Many interesting ideas.

---

## David says

If Jerry Seinfeld had decided to become a software professional, he might have written something like this. He observes the things that software developers and their colleagues do, skewers those practices with humor, and then says how it really *ought* to be done. Sometimes I agree with him, sometimes I didn't, but I like his writing well enough that it seemed worth buying a copy of the book. Most of the articles in the book are also available online on Spolsky's blog, but as I seem to recall from his introductory observations, at least a book won't electrocute you if you have a mishap while reading in the bathroom.

---

## Andreea Lucau says

I really enjoyed reading this book, even if Joel was kind of obsessed with Microsoft and Netscape. I like the little stories about problems he faces building his own company and learning about mistakes that compromise a business - not software bugs, but strategy bugs. My things to remember from this book are: know who your customers are and pay attention to what they need/want.

---

## Tim Poston says

In one sense, I haven't read the book, as the book.  
In another sense, I've read it, as the essays on line.  
Now, I'll buy it.

Very few people in IT can think so clearly or so deeply,  
and *very* few can write so clearly.  
Beautiful.

---

## Luís Soares says

Joel does not play well with mumbo-jumbo. Illustrative software development real stories told with a fine sense of humor. Joel seems to be a great developer and software manager, which is hard to find. Every developer and manager should read it.

---

## Tiago says

More than a collection of software engineering thoughts, this book describes a collection of experiences from several years of working in software engineering companies, starting as intern at Microsoft Excel's team, to his own company Fog Creek.

Reading it in 2018, 14 years after it was first published, it's still valuable as it was back then, as most of the lessons still apply (some of them I can relate with my own experience as software engineering). There's a lot to learn from these lessons and Joel provides lots of interesting links and book references, making it a must read for young software engineers and a great read for others.

Some of the lessons I personally relate to are:

- \* his thoughts about using .net as new shiny technology around (in my case wasn't .net but other), how upgrade was being planned, and particularly, how he felt he was not able to write good code even though he had many years of software development under his belt.
- \* painless functional specs (the importance of having clear specs that other developers can use to implement their code and how that can save countless of hours - specially in endless code reviews, which are too late to make changes anyways)
- \* fixing bugs vs. delivering features/value.

Notes to self: it's noticeable that Joel has read many many books about software engineering (technical and non-technical). Evidence was his statement that he read all self-published books from MS employees. It would be interesting to learn his thoughts now about some of the topics he wrote (e.g. Microsoft strategy and how it stands, web applications and where are we heading now, how to cope with even bigger set o technologies/libraries/tools that are appearing everyday).

---

## Reid says

A fair bit out of date at this point, but that's one thing that makes it interesting - he made predictions about technology that are coming true (or not) right around now ("We'll never use [cloud-based] document storage!"; ("When we all have 100mbps internet...").

In any case, a lot of it is not really out of date or style - the software project management aspects in particular. He also drinks the Microsoft kool-aid, so prepare to hear a lot about .NET and Excel. I'm gonna go catch up on his blog.

---

## Fayçal says

Most tech books get old and out-of-date with time. However I learned plenty of stuff from *Joel on Software* and enjoyed the author's humour and approach on many topics.

Here's a non-exhaustive list of things I learned (more) about:

- Always have a bug tracking system.
- Fix bugs first before you move on to working on new features. Saves lots of time.

- The Joel Test. I need to score better at that.
- Character encodings. I always got away with not knowing much about that, now I do. Thanks Joel :D.
- Do paper prototyping.
- Incentive pay. Don't do that.
- Avoid multi-tasking. Humans aren't really good at it.
- Not-Invented-Here Syndrome. It's not necessarily bad.
- Chicken-And-Egg problems in the software world.

There's a sequel: More Joel on Software, I might read it sometime.

Quotes:

I always wanted to start a blog, this is somewhat of a motivation.

[...] Philip Greenspun, who taught me that if you know something, you need to publish it on the web for others to learn from.

Rewarding good coders by promoting them to a different position, one that involves writing English, not C++, is a classic case of the Peter Principle: people tend to be promoted to their level of incompetence.

About the Not-Invented-Here Syndrome:

If it's a core business function, do it yourself, no matter what.

---

## **Narayana says**

One of the best contemporary books on software engineering. Though a bit outdated (articles from early 2000s), the concepts and ideas are extremely valid. Got a lot of new perspectives from this book. Must read.

---

## **Otis Chandler says**

Joel's a windows coder - but a smart one. Its actually interesting to hear about that side of things, as usually I'm just turning my nose up at it :) A good read for any programmer.

---

## **Alasdair Craig says**

I really wanted to like this book more than I did because I'm a big fan of Spolsky. A lot of the content I'd read before from his blog, which I'm OK with and was fully expecting. However it really jumps all over the place. (I suppose the expanded title does allude to that.) The second half is decidedly random, doesn't follow on neatly from the first, and seems to have been added as an afterthought. The book stops so abruptly with no conclusion but a Q&A chapter where both the questions and answers I feel haven't really stood the test of time. I'm glad I bought it, his writing is very entertaining, and I'm still a big fan - but I give it only 3 stars.

---

## **Ramanan Balakrishnan says**

Essential reading if you write/read/make/design/touch/use software. Interesting to see how assertions from almost 20 years ago have turned out - most are still valid. Even more valuable are the cases where the predictions have turned out differently.

---

## **Adnan Ali says**

Treat it like the Bible. Take only the good stuff out of it.

---

## **Ant says**

Great book, a bit dated now but interesting to see how some of his ideas went with regards to Microsoft's business directions.

Totally love his style and he's obviously a smart cookie who knows his stuff both as a programmer and a project manager. His wit is very engaging and turns even the driest discussions about API's into something of a comedy.

Spolsky talks from first hand experience, he's no theoretician which ironically allows him room to theorize quite accurately given the retrospective view this book now gives on the whole software industry.

One of the best parts about this book is that it makes one, as a developer, feel that what project managers are constantly telling you (It's late, you're slow, your estimates are unrealistically large) is somewhat of a universal pain felt by even the best of us. i.e. Joel.

Despite his plethora of knowledge, wisdom on the subjects he speaks about and intellect, he always comes across as humble and one who perhaps learnt the hard way, like us.

It's no longer 2003, but Joel on Software is still as relevant as it was back then.

Encouraging and realistic at the same time. A friend to read.

---

## **Erkin Unlu says**

It is surprisingly refreshing to find out that Joel's thoughts about how software should be developed nearly

fifteen years ago have all been accomplished and further advanced. It is also funny to read about how Microsoft will lose the API wars (they lost it to the Web and Mobile) but still stay strong (which is still true).

---