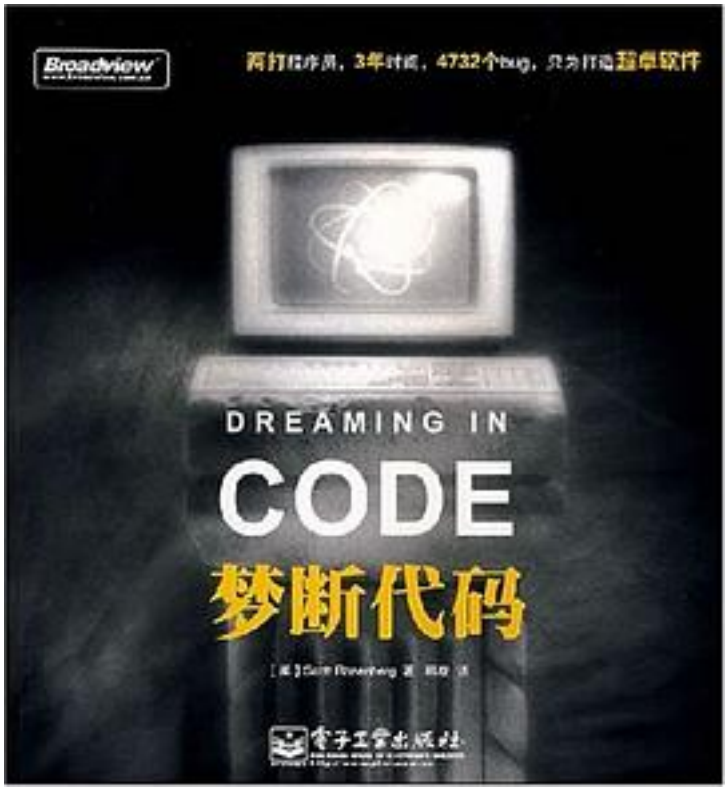
梦断代码



梦断代码_下载链接1_

著者:罗森伯格

出版者:电子工业出版社

出版时间:2011-6

装帧:平装

isbn:9787121135699

《梦断代码》内容简介：软件乃是人类自以为最有把握，实则最难掌控的技术。《梦断代码》作者罗森伯格对OSAF主持的Chandler项目进行田野调查，跟踪经年，试图借由Chandler项目的开发过程揭示软件开发中的一些根本性大问题。

《梦断代码》是讲一事，也是讲百千事；是写一软件，也是写百千软件；是写一群人，也是写百千万人。任何一个在软件领域稍有经验的技术人员看完《梦断代码》，必掩卷长叹：做软件难。

作者介绍:

Scott Rosenberg：作家，编辑，1981年毕业于哈佛大学，1995年与他人共同创办了Salon网站，此后担任其首席技术编辑达数年之久，并负责技术工作。从1986到1995年，一直为San Francisco Examiner写作，最初写剧评，后来又写影评和“数字文化”专栏。所写的剧评曾于1989年获George Jean Nathan奖。在进入Examiner之前，一直为Boston Phoenix写剧评、影评和书评。个人博客地址为www.wordyard.com。

韩磊，技术管理者和作译者。2001年刨办CoDelphi.com中文开发在线网站，2003年加入CSDN，历任网站总监、副总经理、《程序员》杂志和CSDN网站总编辑等职，2010年加入21世纪报系，现任21世纪新媒体副总经理、CTO。拥有美国Borland公司“Delphi产品专家”及“微软最有价值专家”称号。普领导开发多个网站和移动应用项目。译有《C#编程风格(Elements of C# Style)》、《梦断代码(DreaminginCode)》、《代码整洁之道(Clean Code)》，与刘韧合著有《网络媒体教程》，与戴飞合译有《Beginning c# Objects：对象到代码》。微博地址：http：//weibo.com／grhunter。

目录: 第0章 软件时间

・・・・・・(收起)

梦断代码_下载链接1_

标签

软件工程

软件开发

项目管理

计算机

编程

程序人生

梦断代码

计算机科学

评论

比起《DOOM启世录》《观止》《苹果往事》等其他某公司或项目经历的纪实类作品而言，此书在项目管理、开发等技术上的描写更“硬”，而且Chandler项目实际上是作为记述的起始点，由它引出了很多软件工程和技术的思考。在自己周围，其实很多烂事或者正在变烂的事和想法都在这本书里看到了。当然，也有一些没有被验证是否烂的想法，我想试试看。

不以物理学为基础的软件工程不可称其为工程，应称其为管理学。再有就是，做软件真是难。

人类要面对计算机世界的复杂性，软件工程很大程度上只是个人经验。书里的轶事很多。

翻译还是不是特别满意，也许是原作者旁征博引，让没有相关背景知识的人比较难懂吧。

一本提出问题并深入探究了成因和解决方案的书~

真的一点不参水成功的软件，好像俺还没有做过。一般的所谓的成功，是可以打折扣的。

一部软件开发的血泪史 TwT 当我对计算机编程和软件开发失去信心的时候觉得自己无能的时候 请看此书

代码没有价值，但往往能告诉你怎么走下去。

一个历时多年的软件项目惨败的故事，其实类似的案例很多很多，不过大家都不愿意承认失败，或者将失败轻描淡写而已。

3年时间，打造不出一个成熟的PIM软件，所有程序员都训练有素，也有资金支持，简直是黑洞。这本书是关于人类高级智力产品的墓志铭，记录了一场惨痛的失败

这本书展示了硅谷遗留的软件开发者是如何进行产品开发的，把人、事、技术以及产品的发展过程结合在一起，每个有志于开发畅销产品的程序员都值得耐心品味每个故事。

看得仓促，可能不适合我这种还在读的学生看，但软件项目管理绝对是软件设计过程中不可或缺的。想想在学校呢，做个小的project，最烦的就是各种报告，各种说明书，根本没有任何的项目进展的概念而言，相比实际开发，尤其是做一个规模大一点的应用，有效的管理，优良的文档，其实更重要不过。

作为失败的案例是比较成功的。

结构性不好

精彩的软件工程故事书，讲的不仅仅是chandler开发的故事，还引经据典，讲述和总结了很多其他软件的开发过程/不同软件开发理念等等。
初期看这本书，其实是想了解基金会+开源这种软件开发模式。
读完这本书，发现看到的不仅仅是chandler的开发，也是很多其他软件的开发过程。内容很丰富，很适合作为软工入门学生的科普书籍。如果我是老师的话，我会推荐这本书给本科生。

尹咪咪。这名字也太。。。。。

翻译的语言差 当看看故事吧

这人是国外的开源软件斗士。。。。跟中国那个姓袁的各有千秋。。

看的我心力交瘁，还有好多不懂的。

需要在读一遍

梦断代码_下载链接1_

书评

花了一周的时间，看完了《Dreaming in Code》（梦断代码），看得我心潮起伏。对里面那帮家伙的评价也起起落落。最终的结论是：外国大牛也不过如此。
别看他们名头那么响，做了那么多超有名的项目，实际的能力（软件开发能力与项目管理能力）看来相当有限。感想很多，想到一点...

在图书馆的阅览室看了这本书，花了我两个小时的时间，午后的阳光透过图书馆的玻璃照进来，很温暖，可是我的心却一点点的凉了下来。
再过半年我，一个计算机系的学生，就要投身到软件开发这个行业中去了，可没有任何经验，仅凭着那些薄弱的理论知识。边看书边记下自己的想法...

前两天与同学交流时，同学让我谈谈软件开发与测试的区别，我说撇开技术和工作细节不谈（除了时常辅助地写些Unit
Test外，实在对测试不甚了解），二者对待同一问题的心态上本该不同——测试人员大抵悲观而怀疑，开发人员往往乐观而执着。作出这样的推断，一方面作为一名...

看完《梦断代码》以后，我一直在思索这本书的主题是什么，想了半天也没有头绪。翻到书一开始的《作者的话》，看到这么一句：「它提出问题，讲述故事」，啊哈，确实如此，作者回避给任何问题一个标准答案，甚至一点倾向都没有，以一个中立的观察者来讲述故事。好的书不仅在于传...

从任何角度，Chandler项目开始时都是值得羡慕的，充足的资金，舒适的环境，激动人心的愿景，高手云集的团队，专业睿智的老板，然而如同大多数项目一样，Chandler在历时多年之后，依然泥足深陷，至今没有一个能真正工作的版本。
是什么让项目陷于困顿？ 1、目标大得遮蔽了他们...

[part 2] 完整评论见：http://yishan.cc/blogs/xin/archive/2008/11/17/ii.aspx
时间和交流：时间对每一个人都是公平的，对每一个软件项目也是这样。 nearly all software projects require only 1/6 of their time for the writing of code and fully 1/2 of their sche...

这本书看了已经一半多了，就看完的这些部分说点自己想说的。开始看的时候，还是很轻松很调侃的在看老外大牛们的囧事。可是越看越发现这个项目里的很多扯淡的事情其实每天都发生在自己的身边。冷汗啊，一身一身的出，想想以前的很多事情，那真是不停的后怕。 做技术的人，...

引子
昨天晚上，我终于把《梦断代码》这本书看完了。之前，我开展”Chandler中文化”这

个项目，在内心深处的原因是
“我啟動這個項目，并不是從所謂的用戶角度，或者是為了推廣
只是覺得這樣一個優秀的、能與GTD這么緊密結合的軟件 實在罕見，也難以找到 雖...

我手头上有两个人，其中一个对企业流程非常的熟悉，另外一个就CODING来说简直就是上帝，为什么他们搞不定一个小小的自动化过程?
废话，他们两个，一个是火星人，一个是金星人，语言不通。
看不明白的话，不解释了。 ---------- ...

前几日偶然遇到科学松鼠会开出的一份书单，列出了各专业领域经典书籍，跟计算机相关的，第一本就是梦断代码。这才想起遥远而又不太遥远的几个下午，我在图书馆楼梯口的某个位置看完了此书。
对我来说这本书跟经典无关。如果你想学程序设计，你应该去看XX编程艺术、thinking in...

刚看完《梦断代码》，虽然是讲一个软件项目是如何失败的，不过里面有几点让我觉得很有意思。
第一，从计算机诞生那一天开始，人们希望用计算机能够为自己提供足够的信息管理，人们有太多的东西，日记、帐本、音频、心得等等，而不仅仅限于管理，在管理的基础上还要在一定程度...

What I've learned: * No revolution starts big. Any grand vision has to start from smaller things like scratching your own itch and dogfooding. * Do not overinvest infrastructure. Do an agile project and remember the vision on the same time. * Start with...

---
*
读这本书真得是消耗了太多的精力，倒不是书不好，而是自己太浮躁，很多书读不进去。 * 第一次借这本书应该是2012年，没看几页就满30天，还。 *
第二次是2013年初，毕业前，还是读不进去。 *
这一次是2014年2月，在独墅湖图书馆借的，超期60天，终于字啊5月5日看完。 * ...

本书的可贵之处，在于真实。在软件开发中，这样真实的历史总在重复。我们总在找寻

“天才”，期望“天才”的组合能开发出“伟大的软件”。其实不然。就像生命体一样，对于一个软件组织而言，要有骨有肉，有器官有皮肤，不同类型的人才组合，方能实现一个根本的目标。而软件开发...

压抑，真的很压抑。
我不是学计算机的，工作三年来一直在想一个问题，软件到底应该怎么开发才好，有标准吗？那些大公司都是怎么做到牛×的？每个项目我都在改进编程的方式，在改进软件设计的流程，修正架构。项目开始时，认为现在的方式很适合，比上一个项目的好，但项目结束...

软件开发是件不靠谱的事。
不管是大教堂还是集市，不管是开源还是自由，都不能改变这个根本属性。
这么说，软件工程的爱好者们以及管理者们肯定要向我扔鸡蛋了。
不过，我心目中的软件开发，大抵就是这么回事。
为什么？因为，真正创新性的软件开发总是带有一点艺术与作坊的意...

---

1. 它只跳票了5年。（《永远的毁灭公爵》跳票了13年） 2.
它只消耗了数百万美元。（《永远的毁灭公爵》消耗了两千万美元） 3.
它毕竟推出了1.0版本，虽然反响很差。（《永远的毁灭公爵》只推出了一点图片和视频，还有无休止的谣言） 4. 它是开源项目，未来也许会有人在它这里找...
