

مهندسی آسونه. این آدمها هستند که سخت هستند.

• بیل کاوگرن، معاون سابق کل شاخه مهندسی در گوگل

زندگی پر از اتفاقات پیش بینی نشده است. هیچ کدوم از ما دو نفر هیچ وقت فکرش رو هم نمی کردیم که یک روز یک کتاب در مورد کار تیمی و همکاری در مهندسی نرم افزار بنویسیم.

مثل خیلی از خوره های کامپیوتر، ما هم علاقه خودمون به ور رفتن با کامپیوتر رو تبدیل کردیم به یه راهی برای کسب درآمد و زندگی کردن بعد از دانشگاه. مثل خیلی از هکرهای دوره و زمانه خودمون توی سال های دهه ۹۰ وقت خودمون رو صرف ساختن کامپیوتر با قطعات استفاده شده، نصب نسخه های لینوکس از روی یه عالمه فلاپی دیسک و یاد گرفتن سیستم های یونیکس کردیم. ابتدا به عنوان System Administrator شروع به کار کردیم و با شروع دوران طلایی ولی حسابی عصر دات کام^۱ برنامه نویسی شرکت های کوچک شدیم. وقتی این حساب ترکید، وارد شرکت هایی شدیم که از سقوط، جان سالم به در بردند و بعد از همه این ها استخدام یک شرکت start-up شدیم تا به صورت تمام وقت روی طراحی و برنامه نویسی یک پروژه کاملا Open Source برای Version Control^۲ به اسم Subversion کار کنیم.

اما بین سال های ۲۰۰۰ و ۲۰۰۵، یه اتفاق غیرمنتظره برای ما رخ داد. زمانی که روی Subversion کار میکردیم، وظایف و مسوولیت های ما به آرامی تغییر کردند. ما دیگر در خلوت خود و به تنهایی مشغول به کد نویسی نبودیم. ما در حال مدیریت و رهبری یک پروژه بزرگ بودیم. این به این معنی بود که باید تمام روز در چندین Chat Room مختلف با گروه های متفاوت برنامه نویس هایی صحبت می کردیم که به صورت داوطلبانه روی پروژه کار می کردن. و این به این معنی بود که تقریبا تمام امکانات و توسعه نرم افزار رو باید از طریق ایمیل هماهنگ می کردیم. در طول این مسیر متوجه شدیم که کلید موفقیت یک پروژه برنامه نویسی فقط در یک کدنویسی فوق العاده خلاصه نمیشه، بلکه نحوه همکاری و ارتباطات افراد دخیل در پروژه هم به همین میزان حائز اهمیتیه.

سال ۲۰۰۵، شعبه دفتر مهندسی گوگل در شیکاگو رو تاسیسی کردیم و زندگی کاری خودمون رو به عنوان دو برنامه نویس ادامه دادیم. در این نقطه از زندگی، ما هر دو غرق در دنیای Open Source نه تنها فقط پروژه Subversion بلکه Apache Software Foundation هم بودیم. ما Subversion رو به زیرساخت های گوگل منتقل کردیم و یک سرویس جدید برای میزبانی پروژه های Open Source (یه سرویس مشابه SourceForge) ارائه دادیم. سپس شروع کردیم که در همایش ها و کنفرانس هایی که برای مهندسين نرم افزار برگزار میشد، مثل OSCON, ApacheCon, PyCon و نهایتا Google I/O سخنرانی کردیم. اینجا بود که متوجه شدیم به دلیل تجربه هایی که هم در محیط های شرکتی بزرگ و هم در کار کردن با پروژه های Open Source کسب کرده بودیم، علم و دانش اینکه تیم های مهندسی نرم افزار چطور کار میکنند رو آموخته بودیم. سخنرانی های طنزآمیز ما در مورد روش های ناکارآمد و غلط توسعه نرم افزار، مثل ارائه "بدترین عادات در Subversion" کم کم تبدیل شدند به سخنرانی در مورد روش های حفاظت کردن تیم ها از چنگال آدم های احمق. به عنوان نمونه: سخنرانی "چطور پروژه های Open Source از دست انسان های سمی نجات پیدا میکنند." به مرور افراد بیشتری به صحبت های ما میومدند، که میشد به این گردهمایی ها یه جورایی گفت: درمان گروهی! همه با مشکلاتی که ما در موردشون صحبت میکردیم یه جورایی احساس نزدیکی می کردند و میخواستن این مسائل رو به صورت گروهی ریشه کن کنند.

حالا ما اینجا هستیم! ۶ سال بعد، با یه مشت سخنرانی در مورد مشکلات کارگروهی و تیمی در برنامه نویسی و مهندسی نرم افزار. ویرایشگر ما در انتشارات O'Reilly Media به ما پیشنهاد داد که این صحبت ها رو بصورت یک کتاب در بیاریم. بقیه اش دیگه مثل روز روشنیه.



این کتاب برای چه کسانیست؟

این کتاب در ابتدا برای برنامه نویسان نوشته شده بود. برای کسانی که در تلاش هستند تا مقام و حرفه خودشان رو ترقی بدن و نرم افزارهای فوق العاده تولید کنند. ولی در دومین نسخه ای که از کتاب منتشر میکنیم، برای ما کاملاً مشخص شد که مطالب این کتاب گروه وسیع تری از افراد رو شامل میشه. اگر شما در کاری مشغول هستید که نیازمند همکاری با افراد دیگه روی یک موضوع خلاقانه هست، این کتاب برای شماست. ممکن است شما عضو یک گروه محلی، گروه مذهبی یا دوستی، یا عضو یک کمیته یا تیمی از معمارها باشید. در هر صورت ما در مورد شما خواننده دو نکته را فرض میکنیم:

- شما در یک تیم با یک سری افراد خلاق دیگه همکاری میکنید. احتمالاً عضو یک شرکت بزرگ یا محیط ساختاریافته مشابه دیگری هستید.
- شما از ساختن، لذت میبرید و اعتقاد دارید که ساختن چیزها باید لذت بخش و دارای پاداش باشد. اگر هدف شما از تولید محصول، صرفاً کسب درآمد برای امرار و معاش باشه، احتمالاً خیلی علاقه ای به خودسازی و رضایت شغلی نداشته باشید.

تجربه شخصی ما از مهندسی نرم افزار میاد. به همین دلیل بیشتر مثال هایی که ما در این کتاب میاریم، از همین دنیای نرم افزاره. ولی تقریباً تمام پروسه ها و استراتژی هایی که در این کتاب مطرح میکنیم قابل تعمیم به بقیه رشته های خلاقانه نیز هست.

حین بررسی اینکه مهندسین چطور می تونن با دیگران همکاری خوبی داشته باشند، به موضوعاتی برخورد میکنیم که در نگاه اول، به نظر نمایان که بخشی از شرح مسوولیت های یک مهندس باشند. در بعضی صفحات این کتاب در مورد این صحبت خواهیم کرد که چطور یک تیم را رهبری کنیم، یا یک سازمان رو هدایت کنیم، یا حتی با کاربران نرم افزار

خود به رابطه صمیمانه و سالم رو برقرار کنیم. با یک نگاه کلی، این بخش های کتاب این طور به نظر میرسن که برای مدیران تیم ها نوشته شدند، ولی ما به شما اطمینان میدیم که بالاخره به روزی میرسه که شما هم خودتون رو در جایگاهی پیدا میکنید که باید این مهارت ها رو به کار بگیرین. نابوری رو کنار بذارین و ادامه این کتاب رو بخونید. هر چه که توی این کتاب نوشته شده، به کسانی که سازنده یک محصول هستند، مرتبطه.

هشدار! این یک دفترچه راهنما نیست.

قبل ازینکه شروع کنیم، لازمه که سطح توقعات رو تنظیم کنیم. برنامه نویس ها عاشق کتاب هایی هستند که صورت مسائل رو با یک فرمول ریاضی دقیق ارائه بدن و برای هر مسئله یک راه حل مرحله به مرحله تجویز کنند. این ازون کتابها نیست!

کتاب ما به بعد / انسانی تولید محصولات مبتکرانه می پردازه، و انسان ها موجودات پیچیده ای هستند. همون طور که ما معمولا دوست داریم توی سخنرانی هامون بگیم: " انسان ها به طور کلی به مشیت "باگ" (bug) هستن که به صورت متناوب پدیدار میشن! " مسائل و راه حل هایی که درموردشون صحبت می کنیم، بی قاعده تر از آنی هستند که بشود در جعبه های منطقی از پیش تعریف شده قرارشون داد. این کتاب اصولا یک مجموعه مقالات است. در هر بخش به سری مشکلات که به هم مربوط هستند رو حکایت وار مطرح میکنیم و سپس در مورد راه حل هایی که به آنها مربوط هستند بحث می کنیم. برای این که بتونین دامنه توجهتون رو برای دقت کردن به چندین صفحه گسترش بدین، از نیمه راست مغز خودتون کمک بگیرین تا فصل های مختلف کتاب رو به هم ربط بدین.

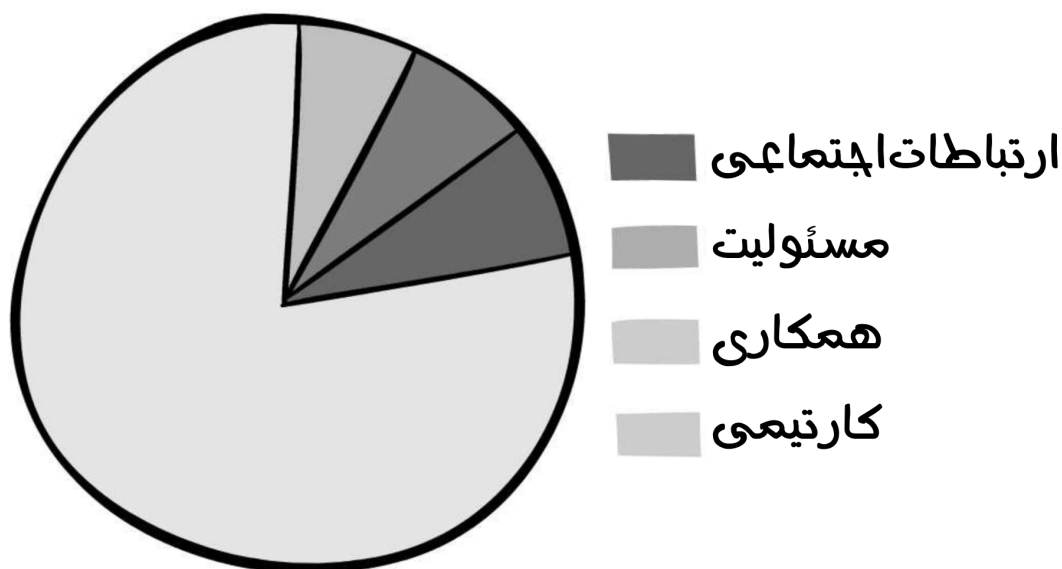
یک سری مسوولیت های دیگه رو هم باید از روی دوش خودمون برداریم. معمولا توی سخنرانی های خودمون به شوخی این طور می گوییم که: " تمام این نظرات، نظرات شخصی خود ما و نتیجه تجربه های شخصی ماست. اگه با اونا مخالفین، کاملا آزاد هستید که سخنرانی خودتون رو داشته باشید. " خارج از شوخی، خوشحال میشیم که در مورد نظرات، تصحیحات احتمالی، پیشنهادات و انتقادات شما صحبت کنیم. هر آنچه در این کتاب آمده، نتیجه درس هایی هست که ما از اشتباهاتمون گرفتیم.

ضمنا لازمه بدونیم که هر اسمی که مثال ها مطرح شده، تغییر داده شده تا از فرد بی گناه (یا گناهکار) حفاظت بشه.

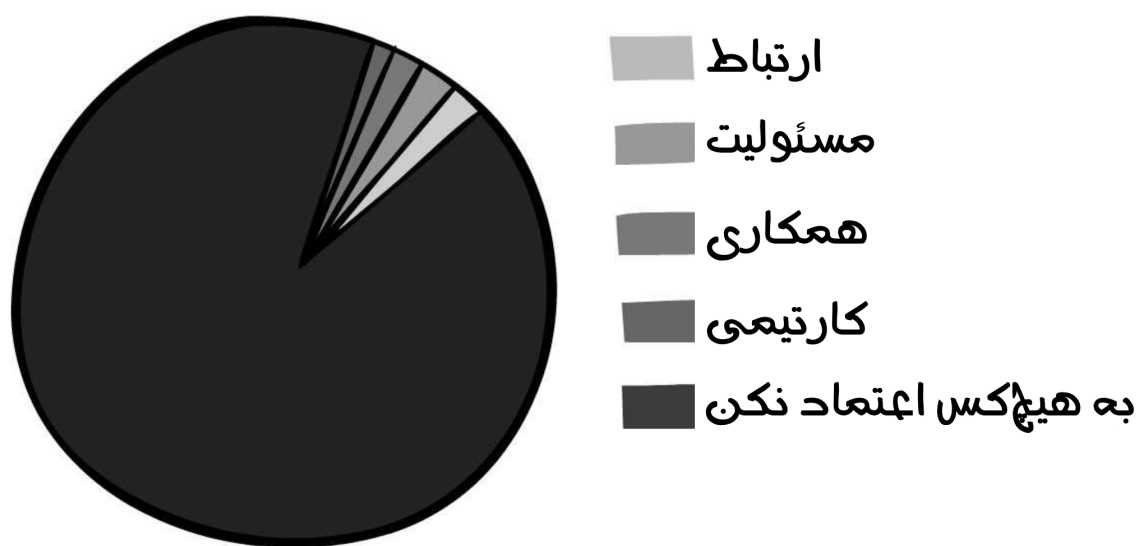
محتوای این کتاب در دانشگاه ها تدریس نمی شوند

بیشتر مهندسين نرم افزاری که ما می شناسیم، بین ۴ تا ۱۰ سال در دانشگاه در مورد علوم کامپیوتر یا مهندسی نرم افزار تحصیل کرده اند. ولی با این وجود تقریبا هیچ برنامه درسی ای^۳ در مورد کار تیمی یا همکاری در یک شرکت وجود نداره. درسته که معمولا دانشجویها بالاخره برای پروژه های درسیشون مجبور میشن که به صورت گروهی کار کنند، ولی بین درس دادن کار گروهی به صورت اصولی و مجبور کردن افراد به کار کردن در گروه، تفاوت های اساسی وجود داره. بیشتر دانشجویها ازین تجربه خسته هستند.

آنچه قرار است در کار گروهی یاد بگیرید



آن چه در عمل در کار گروهی یاد میگیرد



endlessorigami.com

ایده کلی کتاب

موفقیت یک برنامه نویس در یادگیری آخرین زبان های برنامه نویسی یا سریع ترین کدنویسی خلاصه نمیشه. برنامه نویس های حرفه ای تقریباً همیشه عضوی از یک تیم هستند. این طور به نظر میاد که تیم ها، خیلی بیشتر از اونی که همه دوست دارن بهش اعتراف کنند، روی رضایت شغلی و میزان مفید بودن افراد تاثیر دارند.

ایده کلی این کتاب ساده است: توسعه نرم افزار به کار تیمیه و بعد انسانی ماجرا به اندازه عوامل فنی روی نتیجه کار تاثیر مستقیم خواهد داشت. بیشتر آدم ها وقت زیادی صرف یادگیری کار گروهی نکردند، حتی اگه ده ها وقت

صرف یادگیری ابعاد فنی کار کرده باشند. این که یاد بگیریم چطور با هم همکاری کنیم، به اندازه یادگیری مسائل فنی، به موفقیت ما کمک می کند. اگر وقت خودمان را در یادگیری مهارت های ارتباطات و روابط شخصی سرمایه گذاری کنیم، تاثیر به مراتب بیشتری را بدون تلاش اضافه، خواهیم داشت.

^۱ dot-com bubble: به دوران اواخر دهه نود میلادی گفته می شود که به دلیل ارزش گذاری بیش از اندازه و خیالات اشتباه سرمایه داران، شرکت های اینترنتی زیادی ورشکست شدند.

^۲ Version Control ها به سیستم هایی مثل Github یا Bitbucket گفته میشود که به برنامه نویسان کمک میکند که بصورت اشتراکی روی پروژه کار کنند و به تاریخچه کدهای نوشته شده دسترسی داشته باشند.

^۳ ما کتاب PeopleWare از Tom DeMarco رو خوندیم و کتاب خیلی خوبیه. ولی این کتاب بیشتر برای مدیران نوشته شده که ببینن چطور می تونن یک تیم رو موفق تر کنن، تا برای اعضای تیم تا بتونن با هم کارآمد تر همکاری کنند.