

یک دست صدا ندارد

راهنمایی برای کار گروهی بهتر

برایان فیتزپاتریک و بن کالینز

ترجمه: روزین بابایان و صبا جمالیان



Licensed Under:

Creative Commons Attribution- Noncommercial-ShareAlike 3.0

شناسنامه

عنوان: یک دست صدا ندارد؛ راهنمایی برای کار گروهی بهتر

نویسندگان: برایان فیتز پاتریک، بن کالینز

مترجمان: روژین بابایان، صبا جمالیان

ویراستار: مونا اصفهانی

سال انتشار: ژانویه ۲۰۲۱

مجوز: **Creative Commons Attribution – Noncommercial - ShareAlike 3.0**

فهرست

۱.....	مقدمه مترجمان
۳.....	مقدمه
۱۰.....	فصل اول
۴۰.....	فصل دوم
۷۳.....	فصل سوم
۱۱۹.....	فصل چهارم
۱۳۹.....	فصل پنجم
۱۷۴.....	فصل ششم

مقدمه مترجمان

کتاب Debugging Teams یکی از کتاب‌های مورد علاقه ما در زمینه توسعه و پیشرفت کاری و حرفه‌ای است. حدود پنج سال پیش هر دوی ما نسخه اول این کتاب تحت عنوان Team Geek را خواندیم و تأثیرات مثبت آن را در زندگی کاری‌مان تجربه کردیم. محتوای کتاب، نه تنها باعث شد که عملکرد انفرادی بهتری داشته باشیم، بلکه به ما کمک کرد تا ایده‌های تقریباً خوبی را در تیم‌ها و شرکت‌هایی که در آن‌ها کار می‌کنیم به دست بیاوریم. به مرور با افراد دیگری نیز مواجه شدیم که گزارش‌های مستقل و مشابهی در مورد تأثیرات مثبت این کتاب ارائه می‌کردند. این شد که تصمیم گرفتیم تا با ترجمه کتاب به فارسی، یک پل هر چند کوچک بین قالب‌های ذهنی جامعه مهندسی کامپیوتر در شرکت‌های آمریکایی و دوستان و همکاران فارسی‌زبان ایجاد کنیم.

نداشتن تجربه و تخصص کافی خود را در ترجمه انکار نمی‌کنیم. یکی از دلایلی که این ترجمه به طور رایگان در اختیار خوانندگان قرار می‌گیرد، دقیقاً همین نکته است. پیشاپیش بابت اشتباهات احتمالی عذرخواهی می‌کنیم و از همه خوانندگان دعوت می‌کنیم تا نظرات، انتقادات و اصلاحات پیشنهادی خود را یا از طریق ایمیل به آدرس contact@debuggingteams-fa.com یا با ایجاد Pull Request به مخزن فایل‌های ترجمه در Github به آدرس <https://repo.debuggingteams-fa.com> به اطلاع ما برسانند تا در بهبود این ترجمه به ما کمک کنند. هرگونه خوانایی در جملات، انتخاب مناسب کلمات و پیروی از اصول ویرایشی و نگارشی، مدیون ویراستاری دقیق و حرفه‌ای سرکار خانم مونا اصفهانی است که با صبوری ما را در این پروژه یاری کردند.

نویسندگان کتاب، Brian Fitzpatrick و Ben Collins-Sussman توسعه‌دهندگان اصلی نرم‌افزار کنترل نسخه سابورژن و از مهندسين ارشد روزهای اول شرکت گوگل هستند. آن‌ها شعبه گوگل در شهر شیکاگو را تأسیس کرده و تیم‌ها و مهندسين زیادی را هم در شرکت‌های تجاری و هم در پروژه‌های متن‌باز، هدایت و رهبری کرده‌اند. Brian در حال حاضر مدیرعامل و مؤسس یک شرکت نرم‌افزاری تحت عنوان Tock است و Ben مدیر ارشد شرکت گوگل در شیکاگوست. سال‌هاست که هر دو سمینارها و سخنرانی‌های زیادی را در نقاط مختلف دنیا در زمینه توسعه تیمی نرم‌افزار برای مهندسين کامپیوتر برگزار می‌کنند.

این ترجمه با اجازه کامل و رسمی از هر دو نویسنده و تحت لیسانس Creative Commons Attribution-Noncommercial-ShareAlike 3.0 License انجام شده است. معنای آن این است که کلیه حقوق و محتوای این ترجمه متعلق به نویسندگان اصلی است و استفاده مادی از محتوای این ترجمه برای کلیه افراد در سراسر دنیا (من جمله ایران) کاملاً ممنوع بوده و پیگرد قانونی دارد. از کلیه خوانندگان خواهشمندیم در صورت مشاهده نسخه‌های غیرقانونی این ترجمه که به هر شکلی و به طور رایگان در اختیار عموم قرار نگرفته‌اند، خیلی فوری ما را از طریق ایمیل contact@debuggingteams-fa.com مطلع کنند.

در پایان لازم است که از دوستان و همکاران عزیزمان، محمدحسین فرشاد، فاطمه احمدی‌فخر، مهربان برات‌پور و محمد رحمانیان، که پس از مطالعه نسخه‌های ابتدایی ترجمه با نظرات و پیشنهادهایشان به بهبود اثر کمک کرده‌اند، کمال تشکر و قدردانی را داشته باشیم.

با احترام

روژین بابایان

صبا جمالیان

ژانویه ۲۰۲۱

مقدمه

«مهندسی آسونه. این آدم‌ها هستند که سخت‌اند.»

- بیل کاوگرن، معاون سابق کل شاخه مهندسی در گوگل

زندگی پر از اتفاقات پیش‌بینی نشده است. هیچ‌کدام از ما دو نفر هیچ‌وقت فکرش را هم نمی‌کردیم که یک روز کتابی در مورد کار تیمی و همکاری در مهندسی نرم‌افزار بنویسیم.

مثل خیلی از خوره‌های کامپیوتر، ما هم دائماً خودمان را با کامپیوتر مشغول می‌کردیم و تصمیم گرفتیم از این علاقه راهی برای کسب درآمد برای زندگی بعد از دانشگاه پیدا کنیم. مثل خیلی از هکرهای دوره و زمانه خودمان در سال‌های دهه ۹۰، وقتمان را برای ساختن کامپیوتر با قطعات استفاده شده، نصب نسخه‌های لینوکس از روی تعداد زیادی فلاپی دیسک و یاد گرفتن سیستم‌های یونیکس صرف می‌کردیم. ابتدا به عنوان ادمین سیستم‌عامل‌ها کارمان را شروع کردیم و با شروع دوران طلایی، اما حساب‌گونه عصر دات‌کام^۱ برنامه‌نویس شرکت‌های کوچک شدیم. وقتی این حساب ترکید، وارد شرکت‌هایی شدیم که از سقوط، جان سالم به در برده بودند و بعد از طی این مراحل در یک شرکت استارت‌آپ استخدام شدیم تا به صورت تمام‌وقت روی طراحی و برنامه‌نویسی یک پروژه کاملاً متن‌باز^۲ بر اساس سیستم ورژن کنترل^۳ و به اسم ساب‌ورژن کار کنیم.

اما بین سال‌های ۲۰۰۰ و ۲۰۰۵، اتفاقی غیرمنتظره برای ما رخ داد. زمانی که روی عملیات ساب‌ورژن کار می‌کردیم، وظایف و مسئولیت‌های ما به آرامی تغییر کردند. ما دیگر در خلوت خود و به تنهایی مشغول کدنویسی نبودیم، بلکه در حال مدیریت و رهبری یک پروژه بزرگ بودیم. این به این معنی بود که باید کل روز در چندین چتروم مختلف با گروه‌های متفاوت برنامه‌نویسی صحبت می‌کردیم که به صورت داوطلبانه روی پروژه کار می‌کردند. و این به معنی آن بود که تقریباً تمام امکانات و توسعه نرم‌افزار را باید از طریق ایمیل هماهنگ می‌کردیم. در طول این مسیر متوجه شدیم که کلید موفقیت یک پروژه برنامه‌نویسی فقط در یک کدنویسی فوق‌العاده خلاصه نمی‌شود، بلکه نحوه همکاری و ارتباطات افراد دخیل در پروژه هم به همین میزان حائز اهمیت است.

۱. dot-com bubble؛ به دوران اواخر دهه نود میلادی گفته می‌شود که به دلیل ارزش‌گذاری بیش از اندازه و خیالات اشتباه سرمایه‌داران، شرکت‌های اینترنتی زیادی ورشکست شدند.

2. Open Source

۳. Version Control؛ ها به سیستم‌هایی مثل Github یا Bitbucket گفته می‌شود که به برنامه‌نویسان کمک می‌کند به صورت اشتراکی روی یک پروژه کار کنند و به تاریخچه کدهای نوشته شده دسترسی داشته باشند.

سال ۲۰۰۵، شعبه دفتر مهندسی گوگل در شیکاگو را تأسیس کردیم و زندگی کاری مان را به عنوان دو برنامه‌نویس ادامه دادیم. در این نقطه از زندگی، ما هر دو غرق در دنیای متن‌باز شده بودیم و نه تنها پروژه ساب‌ورژن را اداره می‌کردیم، بلکه مالک بنیاد نرم‌افزاری/آپاچی^۱ هم شده بودیم. ما پروژه‌مان را به زیرساخت‌های گوگل منتقل کردیم و یک سرویس جدید برای میزبانی پروژه‌های متن‌باز که مشابه سرویس ردیابی منابع^۲ بود، ارائه دادیم. سپس به مرور در همایش‌ها و کنفرانس‌هایی که برای مهندسی نرم‌افزار برگزار می‌شد، مثل پایکان^۳ و آپاچی کان^۴ و آی.اس.کان^۵ و در نهایت گوگل آی.آی.^۶ سخنرانی کردیم. اینجا بود که متوجه شدیم به دلیل تجربه‌هایی که هم در محیط‌های شرکتی بزرگ و هم در کار با پروژه‌های متن‌باز به دست آورده بودیم، مهارت و دانش و کارایی تیم‌های نرم‌افزاری را یاد گرفته بودیم. سخنرانی‌های طنزآمیز ما در مورد روش‌های ناکارآمد و اشتباه توسعه نرم‌افزار، مثل ارائه «عادت‌های بد در ساب‌ورژن» کم‌کم به سخنرانی در مورد روش‌های محافظت تیم‌ها از چنگال آدم‌های احمق تبدیل شدند. به عنوان نمونه، سخنرانی «چطور پروژه‌های متن‌باز از دست انسان‌های سمی نجات پیدا می‌کنند؟» باعث شد تا کم‌کم افراد بیشتری وارد بحث‌های ما شوند، تا آنجایی که می‌شد به این گردهمایی‌ها یک‌جورهایی درمان گروهی هم گفت! همه با مشکلاتی که ما در موردشان صحبت می‌کردیم به نحوی احساس نزدیکی می‌کردند و می‌خواستند این مسائل را به صورت گروهی ریشه‌کن کنند.

حالا ما اینجا هستیم! شش سال بعد، با مقادیر زیادی سخنرانی در مورد مشکلات کار گروهی و تیمی در برنامه‌نویسی و مهندسی نرم‌افزار. ویرایشگر ما در انتشارات O'Reilly Media به ما پیشنهاد داد که این صحبت‌ها را به صورت یک کتاب در بیاوریم. بقیه داستان که دیگر مثل روز روشن است.

1. Apache Software Foundation

۲. SourceForge؛ یک نوع سرویس مبتنی بر وب است که به توسعه‌دهندگان نرم‌افزار یک مکان آنلاین و متمرکز برای کنترل و مدیریت پروژه‌های نرم‌افزاری رایگان ارائه می‌دهد تا بتوانند کدهای منبع یا ردیابی اشکال و معکوس کردن بارگیری‌ها را به صورت دقیق پیدا کنند. (و.)

3. PyCON

4. ApachCON

5. OSCON

6. Google I/O



این کتاب برای چه کسانی است؟

این کتاب در ابتدا برای برنامه‌نویسان نوشته شده بود. برای کسانی که در تلاش هستند تا در مقام و حرفه خودشان پیشرفت کنند و نرم‌افزارهای فوق‌العاده تولید کنند. ولی در دومین نسخه‌ای که از کتاب منتشر می‌کنیم، برای ما کاملاً مشخص شده است که مطالب این کتاب برای گروه وسیع‌تری از آدم‌ها قابل استفاده است. اگر شما مشغول کاری هستید که نیازمند همکاری با افراد دیگر روی یک موضوع خلاقانه است، این کتاب برای شما مناسب است. ممکن است شما عضو یک گروه محلی، مذهبی یا گروه دوستی یا عضو یک کمیته یا تیمی از معمارها باشید. در هر صورت، ما دو نکته را در مورد شمای خواننده مد نظر قرار می‌دهیم:

- شما در یک تیم هستید و با یک سری افراد خلاق دیگر 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 لذت‌بخش باشد و پاداشی همراه خود داشته باشد. اگر هدف شما از تولید محصول، صرفاً کسب درآمد برای امرارمعاش باشد، احتمالاً خیلی علاقه‌ای به خودسازی و رضایت شغلی نداشته باشید.

تجربه شخصی ما از مهندسی نرم‌افزار می‌آید. به همین دلیل بیشتر مثال‌هایی که ما در این کتاب می‌آوریم، از همین دنیای نرم‌افزار است. ولی تقریباً تمام فرایندها و استراتژی‌هایی که در این کتاب مطرح می‌کنیم قابل تعمیم به سایر رشته‌های خلاقانه نیز هستند.

حین بررسی اینکه مهندسی چطور می‌تواند با دیگران همکاری خوبی داشته باشند، به موضوعاتی برخورد می‌کنیم که در نگاه اول، به نظر نمی‌آیند بخشی از شرح مسئولیت‌های یک مهندس باشند. در بعضی صفحات این کتاب در مورد این صحبت خواهیم کرد که چطور یک تیم را رهبری کنیم یا یک سازمان را هدایت کنیم یا حتی با کاربران نرم‌افزار خود یک رابطه صمیمانه و سالم را برقرار کنیم. با یک نگاه کلی، به نظر می‌رسد که بخش‌های این کتاب برای مدیران تیم‌ها نوشته شدند، ولی ما به شما اطمینان می‌دهیم که بالاخره روزی خواهد رسید تا شما هم خودتان را در جایگاهی ببینید که این مهارت‌ها را به کار گرفته‌اید. ناباوری را کنار گذاشته و ادامه این کتاب را بخوانید. هرچه که در این کتاب نوشته شده، به کسانی که سازنده یک محصول هستند، ارتباط دارد.

هشدار! این یک دفترچه راهنما نیست.

قبل از اینکه شروع کنیم، لازم است که سطح توقعاتمان را تنظیم کنیم. برنامه‌نویس‌ها عاشق کتاب‌هایی هستند که صورت مسائل را با یک فرمول ریاضی دقیق ارائه بدهند و برای هر مسئله یک راه‌حل مرحله به مرحله تجویز

کنند.

این از آن کتاب‌ها نیست!

کتاب ما به بُعد انسانی تولید محصولات مبتکرانه می‌پردازد و انسان‌ها موجودات پیچیده‌ای هستند. همان طور که ما معمولاً دوست داریم در سخنرانی‌هایمان بگوییم: «انسان‌ها به طور کلی یک مشت باگ هستند که به صورت متناوب پدیدار می‌شوند!» مسائل و راه‌حلهایی که در موردشان صحبت می‌کنیم، بی‌قاعده‌تر از آنی هستند که بشود در جعبه‌های منطقی و از پیش تعریف شده قرار بگیرند. این کتاب اصولاً مجموعه‌ای از چندین مقاله است. در هر بخش یک سری مشکلات را که به هم مربوط هستند به صورت حکایت‌وار مطرح می‌کنیم و سپس در مورد راه‌حلهایی که به آن‌ها مربوط هستند بحث می‌کنیم. برای اینکه بتوانید دامنه توجّهتان را برای دقت کردن به چندین صفحه گسترش بدهید، از نیمه راست مغز خود کمک بگیرید تا فصل‌های مختلف کتاب را به هم ربط بدهید.

یک سری مسئولیت‌های دیگر را هم باید از روی دوش خودمان برداریم. معمولاً در سخنرانی‌هایمان به شوخی این طور می‌گوییم: «تمام این نظرات، نظرات شخصی خود ما و نتیجه تجربه‌های شخصی ماست. اگر با آن‌ها مخالف هستید، کاملاً آزادید که سخنرانی خودتان را داشته باشید.» خارج از شوخی، خوشحال خواهیم شد که در مورد نظرات، تصحیحات احتمالی، پیشنهادات و انتقادات شما صحبت کنیم. هر آنچه در این کتاب آمده، نتیجه درس‌هایی است که ما از اشتباهاتمان گرفتیم.

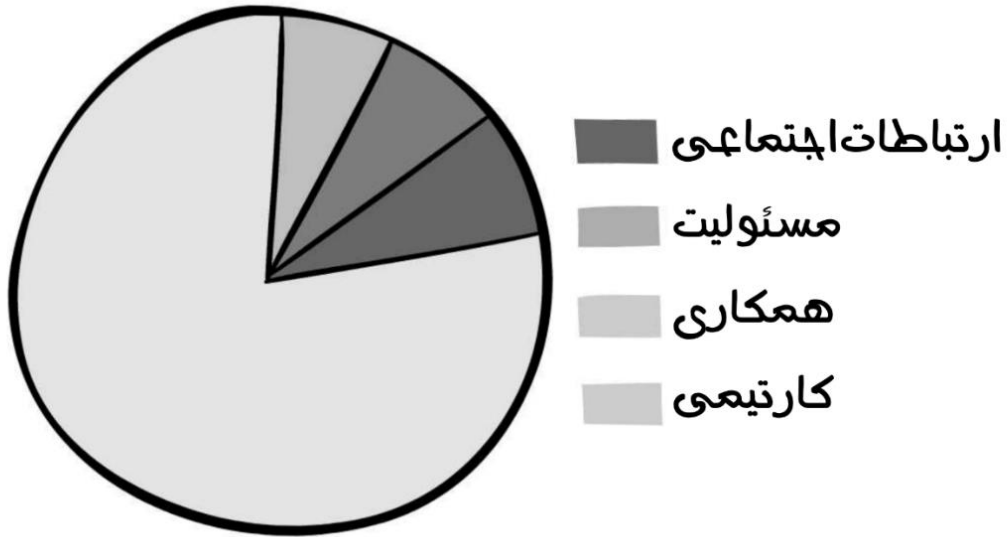
ضمناً لازم است بدانیم هر اسمی که در مثال‌ها مطرح شده، تغییر داده شده تا از هویت فرد بی‌گناه (یا گناهکار) محافظت شود.

محتوای این کتاب در دانشگاه‌ها تدریس نمی‌شوند

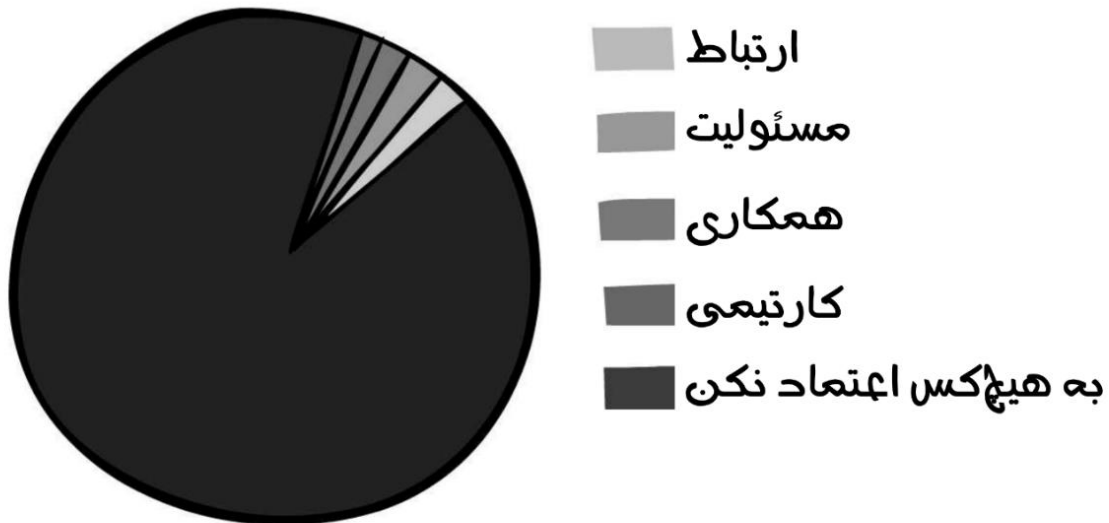
بیشتر مهندسی‌نرم‌افزاری که ما می‌شناسیم، بین ۴ تا ۱۰ سال در دانشگاه در مورد علوم کامپیوتر یا مهندسی نرم‌افزار تحصیل کرده‌اند. ولی با وجود این، تقریباً هیچ برنامه درسی^۱ در مورد کار تیمی یا همکاری در یک شرکت وجود ندارد. درست است که معمولاً دانشجویها بالاخره برای پروژه‌های درسی مجبور می‌شوند به صورت گروهی کار کنند، ولی بین تدریس کار گروهی به صورت اصولی و اجبار افراد به کار گروهی، تفاوت‌های اساسی وجود دارد. بیشتر دانشجویها از این تجربه خسته هستند.

۱. ما کتاب PeopleWare از Tom DeMarco را خواندیم و کتاب خیلی خوبی است. ولی این کتاب بیشتر برای مدیران نوشته شده که ببینند چطور می‌توانند یک تیم را موفق‌تر کنند، تا اعضای تیم بتوانند با هم به صورت کارآمد همکاری کنند.

آنچه قرار است در کار گروهی یاد بگیرید



آن چه در عمل در کار گروهی یاد میگیرد



ایده کلی کتاب

موفقیت یک برنامه‌نویس در یادگیری آخرین زبان‌های برنامه‌نویسی یا سریع‌ترین کدنویسی خلاصه نمی‌شود. برنامه‌نویس‌های حرفه‌ای تقریباً همیشه عضوی از یک تیم هستند. این‌طور به نظر می‌آید که تیم‌ها، خیلی بیشتر از آنچه که همه دوست دارند به آن اعتراف کنند، روی رضایت شغلی و میزان مفید بودن افراد تأثیر دارند.

ایده کلی این کتاب ساده است: توسعه نرم‌افزارها یک کار تیمی است و بعد انسانی ماجرا به اندازه عوامل فنی روی نتیجه کار تأثیر مستقیم خواهد داشت. بیشتر آدم‌ها وقت زیادی صرف یادگیری کار گروهی نکرده‌اند، حتی اگر دهه‌ها وقت صرف یادگیری ابعاد فنی کار کرده باشند. اینکه یاد بگیریم چطور با هم همکاری کنیم، به اندازه یادگیری مسائل فنی، به موفقیت ما کمک می‌کند. اگر وقت خودمان را در یادگیری مهارت‌های ارتباطی و روابط شخصی سرمایه‌گذاری کنیم، تأثیر به مراتب بیشتری را بدون تلاش اضافه‌ای، خواهیم داشت.

فصل اول

افسانه برنامه‌نویس نابغه

از آنجایی که این کتاب در مورد خطرات روابط اجتماعی در توسعه‌دهی محصولات خلاقانه است، به نظر منطقی می‌آید که تمرکزمان را روی تنها متغیری که شما قطعاً روی آن کنترل دارید، بگذاریم: یعنی خود شما.

به طور ذاتی، انسان‌ها هیچ‌وقت کامل نیستند. ولی قبل از اینکه عیب و ایرادهای همکارهای خودتان را بسنجید، لازم است که مشکلات و نقص‌های شخصی خودتان را درک کنید. ما می‌خواهیم در مورد رفتارها، خلق‌وخو و واکنش‌های خودتان فکر کنید و در نتیجه این تفکر امیدواریم که ایده‌بهتری در مورد اینکه چطور می‌شود یک برنامه‌نویس کارآمد و موفق شد به دست آورید. نهایتاً شما انرژی کمتری را برای سروکله زدن با انسان‌ها هدر خواهید داد و وقت بیشتری را صرف کدنویسی خواهید کرد.

نکته مهم و اصلی این فصل این است که شما درک کنید برنامه‌نویسی یک کار تیمی است. و برای اینکه در یک تیم مهندسی (یا هر تیم خلاقانه دیگری) موفق باشید، باید اخلاق و رفتار خودتان را روی سه محور اصلی تواضع، احترام و اعتماد برقرار کنید. قبل از اینکه جلوتر برویم، برای شروع بهتر است ببینیم برنامه‌نویس‌ها به طور کلی چگونه رفتار می‌کنند.

کمکم کن گدم رو مخفی کنم

در ده سال گذشته، هر دوی ما وقت نسبتاً خوبی را صرف سخنرانی در کنفرانس‌های مربوط به برنامه‌نویسی کردیم. بعد از اینکه در سال ۲۰۰۶ سرویس پروژهکت هاستینگ^۱ را به صورت متن‌باز از طرف گوگل عرضه کردیم، با سؤال‌ها و درخواست‌های زیادی روبه‌رو شدیم. اواسط سال ۲۰۰۸ یک روند کلی در کل این سؤال‌ها به وضوح مشخص بود:

- می‌شه لطفاً این قابلیت رو به نرم‌افزار ساب‌ورژن اضافه کنید تا بعضی از شاخه‌های فرعی^۲ از دید بقیه مخفی باشند؟
- می‌تونید این امکان رو فراهم کنید که پروژه‌های متن‌باز در ابتدا مخفی و خصوصی باشند، ولی وقتی آماده شدند بشه اونا رو عمومی کرد و در دسترس همه قرار داد؟
- من می‌خوام تمام کدهام رو از اول بنویسم. می‌شه کمک کنید تمام نسخه‌ها و تاریخچه کدهای گذشته من پاک بشن؟

وجه مشترک تمام این درخواست‌ها را متوجه می‌شوید؟

جواب مشخص است: عدم اعتماد به نفس.

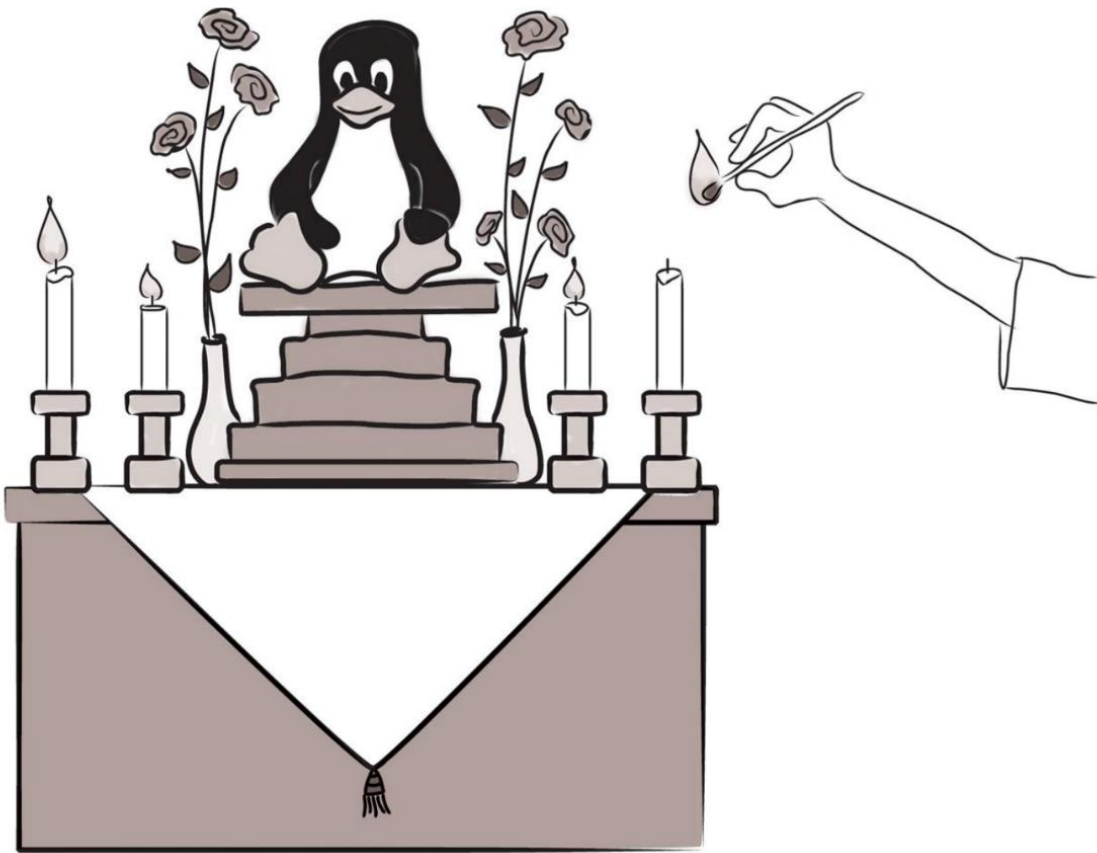
افراد از اینکه بقیه کار ناتمام و در حال پیشرفتشان را ببینند، واهمه دارند. به نظر منطقی می‌آید و بخشی از طبیعت انسان هم همین است. هیچ‌کس دوست ندارد مورد انتقاد قرار بگیرد، مخصوصاً در مورد کارهایی که هنوز کاملاً تمام نشده‌اند. این نحوه نگرش افراد ولی یک سرنخ کلی به ما در مورد دنیای برنامه‌نویسی داد. این عدم اعتماد به نفس، در واقع یک نشانه از وجود یک مسئله وسیع‌تر است.

۱. Project Hosting؛ یک سرویس قدیمی از گوگل برای میزبانی پروژه‌های نرم‌افزاری. (م.)

اسطورهٔ نبوغ

هر دوی ما در طول سال‌های دههٔ ۹۰ در شیکاگو زندگی می‌کردیم و شاهد قهرمانی‌های متعدد تیم شیکاگو بولز^۱ در بسکتبال بودیم. تلویزیون ملی پر شده بود از داستان‌های افتخارات این تیم فوق‌العاده. ولی رسانه‌ها بیشتر روی چه چیزی تمرکز می‌کردند؟ ستارهٔ خارق‌العادهٔ تیم، مایکل جردن! نه کل تیم. هر بسکتبالیستی در دنیا دوست داشت که بتواند مثل ام. جی بشود. ما رقصیدن مایکل جردن با سایر بازیکن‌ها را در زمین می‌دیدیم. ام. جی همه‌جا بود: در تبلیغات تلویزیونی، یا حتی فیلم‌های در پیتی که در آن‌ها با شخصیت‌های کارتونی بسکتبال بازی می‌کرد. او یک ستاره بود. هر بچه‌ای که در هر محله‌ای بسکتبال بازی می‌کرد، آرزو داشت وقتی بزرگ شد جای مایکل جردن را بگیرد.

برنامه‌نویس‌ها هم در تلاش برای پیدا کردن و پرستش بت‌های قهرمان، طبیعت مشابهی دارند. لینوس تروالدز، ریچارد استالمن، بیل گیتس، همه قهرمانانی هستند که با شاهکارهایشان دنیا را تغییر دادند. لینوس تروالدز به تنهایی لینوکس را ساخت، مگر نه؟



1. Chicago Bulls

ولی واقعیت این است که لینوس صرفاً کد مقدماتی یک هسته مشابه یونیکس^۱ را به عنوان یک نمونه کار نوشت و آن را برای یک سری آدم ایمیل کرد. البته که کار کمی نبود و یک موفقیت بزرگ به حساب می‌آمد. ولی کاری که صرفاً یک قسمت کوچک از کل ماجرا بود. لینوکس صدها برابر بزرگ‌تر از این حرف‌هاست و صدها شخص باهوش آن را توسعه داده‌اند. دستاورد واقعی لینوس این بود که این گروه از افراد را هماهنگ و رهبری کرد. لینوکس نهایتاً نتیجه درخشان این تلاش گروهی به حساب می‌آمد (ضمن اینکه خود یونیکس را هم در اصل یک گروه نخبه در Bell Labs نوشته بودند؛ نه فقط کن تامسون و دنیس ریچی).

یک مثال مشابه دیگر: آیا استالمن شخصاً تمام مجموعه نرم‌افزارهای بنیاد نرم‌افزار/آزاد^۲ را نوشته است؟ او نخستین نسل از نرم‌افزار/ای‌مکس^۳ را نوشت. ولی صدها شخص دیگر پشت سایر نرم‌افزارهایی مثل بش^۴ یا جی.سی.سی^۵ هستند که روی لینوکس اجرا می‌شوند. استیو جابز کل تیمی را هدایت می‌کرد که مکینتاش^۶ را ساختند، و همچنین اگرچه بیل گیتس، یک مفسر/اینترپرت^۷ برای زبان برنامه‌نویسی پیسیک نوشت که قابلیت اجرا روی کامپیوترهای ابتدایی خانگی را داشت، ولی دستاورد اصلی او تأسیس یک شرکت بزرگ بود که حول محوریت زبان برنامه‌نویسی ام.اس.داس^۸ به موفقیت رسید. با وجود این، همه این افراد به نمایندگی از یک سری موفقیت جمعی، تبدیل به نمادهایی همیشگی شدند.

مایکل جردن چطور؟ او هم همین طور. ما از او یک بت می‌سازیم، ولی واقعیت این است که او به تنهایی تمام بازی‌های تیمشان را پیروز نشده است. نبوغ واقعی او در این بود که می‌توانست به خوبی با هم‌تیمی‌هایش بازی کند. مربی تیمشان، فیل جکسون، بسیار زرنگ بود. ترفند مربی‌گری او شگفت‌آور است. او به خوبی می‌دانست که یک بازیکن به تنهایی نمی‌تواند باعث قهرمانی شود. بنابراین او یک تیم رؤیایی با محوریت مایکل جردن را گردآوری کرد. تیم او مثل ماشینی که خوب روغن‌کاری شده باشد، روان بود و اگر نگوییم خیلی بیشتر، حداقل این تیم اندازه خود مایکل جردن چشمگیر بود.

-
1. Unix
 2. Free Software Foundation
 3. Emacs
 4. bash
 5. GCC

۶. Macintosh؛ یکی از رایانه‌های شخصی که شرکت اپل اولین بار در تاریخ ۲۴ ژانویه ۱۹۸۴ آن را تولید کرد و اولین رایانه شخصی بود که رابط کاربر گرافیکی داشت. (و.)

۷. Interpreter؛ مفسری برای خواندن زبان‌های برنامه‌نویسی که کدها را خط به خط می‌خواند و به زبان ماشین تبدیل می‌کند. (و.)

8. MSDOS

خوب پس چرا ما در تمام این داستان‌ها، از اشخاص بت می‌سازیم؟ چرا توجه ما آدم‌ها جلب محصولات می‌شود که یک آدم معروف از آن‌ها تعریف کرده باشد؟ چرا دوست داریم لباس یا کفش مایکل جردن را داشته باشیم؟

شهرت، می‌تواند یک دلیل خیلی بزرگ برای این کار باشد. انسان‌ها به طور غریزی به دنبال رهبر و یا یک الگو می‌گردند تا از او یک بت بسازند و سعی کنند شبیه او رفتار کنند. همه ما نیاز به قهرمانانی داریم که از آن‌ها الهام بگیریم. دنیای برنامه‌نویسی هم از این قضیه مستثنا نیست. شهرت در دنیای تکنولوژی یک مفهوم کاملاً شناخته‌شده است و همه ما دوست داریم چیزی را توسعه بدهیم که دنیا را عوض کند، یا یک زبان برنامه‌نویسی فوق‌العاده طراحی کنیم.

ته دل همه ما آرزویی نهفته است، برای اینکه روزی از ما به عنوان یک نابغه یاد کنند. رؤیا و خیال‌پردازی شما این است که روزی یک ایده ناب به ذهنتان خطور کند. به غار تنهایی خودتان بروید و برای هفته‌ها یا ماه‌ها، ایده‌ای را پیاده‌سازی کنید که به ذهنتان رسیده است. سپس این ایده ناب را در دنیا رها کنید تا همه انگشت به دهان به نبوغتان خیره بمانند. همه همکارهای شما تحت‌تأثیر این هوش و ذکاوت استثنایی قرار بگیرند و به صف بایستند تا این نرم‌افزار را ببینند. شهرت و ثروت هم طبیعتاً با همه این‌ها خودش را به شما نشان می‌دهد.

قصد جسارت نداریم، ولی بیایدد واقع‌بینانه به قضیه نگاه کنیم، شما احتمالاً یک نابغه نیستید. بله ما مطمئنیم که شما یک خانم یا آقای باهوش هستید، ولی می‌دانید که نابغه‌های واقعی چقدر کمیاب هستند؟ بله، متوجه هستیم که شما کدنویسی بلدید، که مهارت آسانی نیست و خوب این ویژگی شما را در گروه افراد نسبتاً باهوش جامعه قرار می‌دهد. اصلاً حتی اگر فرض کنیم شما یک نابغه ناب هستید، باز هم این به تنهایی کافی نیست. نابغه‌ها هم اشتباه می‌کنند. همچنین، داشتن یک ایده ناب و مهارت خارق‌العاده برنامه‌نویسی به این معنی نیست که لزوماً نرم‌افزار نهایی به موفقیت خواهد رسید. این توانایی شما در همکاری کردن با بقیه است که می‌تواند باعث شود شما در کارتان یک آدم موفق یا ناموفق شوید.

کاملاً مشخص است که افسانه وجود یک نابغه خارق‌العاده همچنان از عدم اعتمادبه‌نفس ما نشئت می‌گیرد. بیشتر برنامه‌نویس‌ها از به اشتراک گذاشتن کاری که به تازگی شروع کردند صرفاً به این دلیل واهمه دارند که حس می‌کنند همکارها اشتباهاتشان را می‌بینند و متوجه می‌شوند که آن‌ها یک نابغه تمام‌عیار نیستند.

یک بار یک برنامه‌نویس در وبلاگ «بن» گفت:

«من واقعاً اعتمادبه‌نفسم را به کلی از دست می‌دهم وقتی حس می‌کنم آدم‌ها قراره کار من رو قبل از اینکه به اتمام برسه ببینند. حس می‌کنم به شدت کار من رو قضاوت خواهند کرد و فکر می‌کنند من چقدر احمقم.»

این حس، به شدت بین برنامه‌نویس‌ها متداول است، و واکنش طبیعی این است که در غار تنهایی خودتان پنهان شده و مدام کار کنید و کار کنید و کار. هیچ‌کس خرابکاری‌های شما را نخواهد دید و شما همیشه این

شانس را خواهید داشت تا موقعی از کارت‌تان پرده‌برداری کنید که کامل شده باشد. این قدر در خلوت خودتان یواشکی کار کنید تا نتیجه بی‌نقص و بی‌عیب شود.

یک دلیل دیگر برای پنهان کردن کار می‌تواند نگرانی از این موضوع باشد که برنامه‌نویس دیگری از کار شما ایده بگیرد و قبل از اینکه شما فرصت کنید اجرای آن را تمام کنید، او ایده شما را عملی کند. با مخفی کردن کارت‌تان، یک‌جورهایی از ایده‌تان حفاظت می‌کنید.

می‌دانیم که الان احتمالاً به چه چیزی فکر می‌کنید: «خوب که چی؟ آدم اجازه نداره هر طوری که دوست داشته باشه کار کنه؟!»

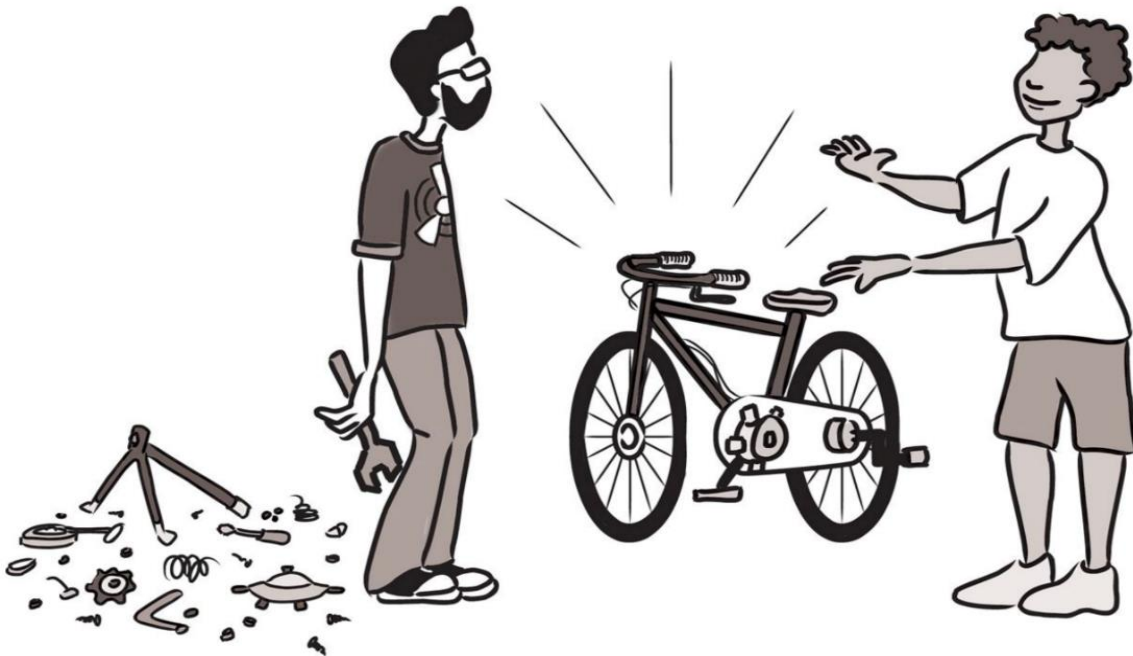
خوب راستش نه! توی این مورد خاص ما شدیداً معتقدیم که این کار شما یک اشتباه بزرگ است. در ادامه دلیل این عقیده را توضیح می‌دهیم.

پنهان کاری در برنامه‌نویسی، عامل ضرر و زیان

اگر شما تمام کارتان را در تنهایی و خلوت خودتان انجام دهید، خطر شکست و همین طور احتمال عدم پیشرفت خودتان را به شدت افزایش می‌دهید.

اولاً، شما از کجا می‌دانید که در مسیر درستی قرار دارید؟

تصور کنید شما یک سازنده و طراح دوچرخه هستید. یک روز یک ایده خارق‌العاده و ناب برای طراحی جدید دنده دوچرخه به ذهنتان می‌رسد. قطعاتی را که لازم دارید سفارش داده و هفته‌ها در کارگاهتان روی ساخت یک نمونه اولیه، سخت تلاش می‌کنید. وقتی همسایه شما، که اتفاقاً او هم در ساخت دوچرخه سررشته دارد از شما در مورد کاری که می‌کنید سؤال می‌پرسد، شما توضیحی نمی‌دهید. شما نمی‌خواهید تا زمانی که کارتان کامل و بی‌نقص تمام نشده، کسی در مورد پروژه شما چیزی بداند. چند ماه دیگر می‌گذرد و شما برای ساخت نمونه کارتان دچار مشکل می‌شوید. ولی خوب چون تمام این مدت را مخفیانه کار کردید، کمک گرفتن از دیگران در این مرحله کاملاً غیرممکن است. یک روز ناگهان همسایه‌تان را می‌بینید که دوچرخه خودش را از پارکینگ بیرون آورده و روی دوچرخه‌اش یک دنده خیلی مدرن و خاص سوار کرده است. کاشف به عمل می‌آید که اتفاقاً او هم روی یک ایده خیلی مشابه به ایده شما کار می‌کرده، ولی از چند نفر دیگر از دوستان و همکاران خودش هم کمک گرفته است. اینجاست که شما اعصابتان به هم می‌ریزد. کار خودتان را به همسایه نشان می‌دهید و او خیلی سریع برایتان توضیح می‌دهد که کجای کار شما اشکال دارد. اشکالاتی که احتمالاً همان هفته اول حل می‌شدند؛ اگر همان ابتدا همسایه را در جریان پروژه خود گذاشته بودید.



چند نکته در این حکایت نهفته است. اگر شما ایده ناب خودتان را از بقیه مخفی نگه دارید و از نشان دادن کار در حال پیشرفتتان به بقیه خودداری کنید، در حال یک قمار بزرگ هستید. اشتباهات اساسی در طراحی‌های اولیه خیلی به راحتی پیش می‌آیند. ممکن است خیلی راحت دوباره چرخ را اختراع کنید^۱، همچنین فواید استفاده از همکاری بقیه را نیز از دست می‌دهید. دقت کردید که چطور همسایه شما دنده دوچرخه را با کمک همکارانش سریع‌تر از شما پیاده‌سازی کرد؟

مردم به همین دلیل، قبل از اینکه در آب شیرجه بزنند، دمای آب را با نوک پنجه‌های پایشان چک می‌کنند. قبل از شروع هر کاری لازم است مطمئن شوید که اولاً شما روی موضوع درستی کار می‌کنید، ثانیاً نحوه اجرای کارتان صحیح است و ثالثاً این کار قبلاً انجام نشده است. احتمال خطا در ابتدای کار بسیار بالاست. هرچه بیشتر از بقیه نظرشان را بپرسید، احتمال اشتباهات اولیه را کمتر می‌کنید.^۲

اینکه کارتان را از ابتدا با بقیه به اشتراک بگذارید و نظرشان را بپرسید، فقط باعث کاهش اشتباه و بیراهه رفتن نمی‌شود، بلکه باعث تقویت ویژگی خاصی در پروژه شما می‌شود که ما اسم آن را «ضریب اتوبوسی»^۳ می‌گذاریم! ضریب اتوبوسی: برابر با حداقلی از افراد است که اگر از بین بروند، پروژه شما هم کاملاً تعطیل می‌شود و از بین می‌رود.

۱. مخصوصاً اگر واقعاً سازنده دوچرخه باشید:)

۲. البته نظرسنجی بیش از اندازه در ابتدای کار پروژه می‌تواند خطرناک باشد. در مورد این موضوع در فصل‌های آینده بیشتر صحبت خواهیم کرد.



دانش و اطلاعات چقدر بین افراد تیمتان پخش شده است؟ اگر شما تنها کسی هستید که جزئیات یک نمونه کار را بلد هستید، هرچند که ممکن است از لحاظ امنیت شغلی مورد خوبی به نظر بیاید، ولی اگر روزی یک اتوبوس با شما برخورد کند، کل پروژه از بین خواهد رفت.

ولی اگر شما و دوستان با هم روی پروژه کار کرده باشید، عملاً ضریب اتوبوسی پروژه را دو برابر کرده‌اید و اگر توانسته باشید یک تیم را در طراحی و پیاده‌سازی نمونه کار دخیل کنید، وضعیت شما بسیار بهتر هم خواهد بود. پروژه صرفاً با ناپدید شدن یک نفر، از بین نخواهد رفت. درست است که احتمالاً اعضای تیم خدایی نکرده با اتوبوس برخورد خواهند کرد، ولی اتفاقات پیش‌بینی‌نشده دیگری ممکن است پیش بیایند. مثلاً یک نفر ممکن است ازدواج کند یا به شهر دیگری منتقل شود یا از شرکت استعفا دهد یا مجبور شود از یکی از اعضای خانواده‌اش که مریض شده مراقبت کند. نکته بحث این است که شما باید در مورد پروژه این آینده‌نگری را داشته باشید و با بالا بردن ضریب اتوبوسی تیم، احتمال موفقیت پروژه را افزایش دهید.

صرف‌نظر از بحث ضریب اتوبوسی، مسئله سرعت کلی کار هم حائز اهمیت است. به راحتی می‌شود فراموش کرد که سرعت کار کردن انفرادی چقدر پایین است. خیلی پایین‌تر از میزانی که آدم‌ها دوست دارند به آن اعتراف کنند. وقتی تنها کار می‌کنید، چه چیزهای جدیدی یاد می‌گیرید؟ سرعت کارتان چقدر است؟ اینترنت دنیای اطلاعات است، ولی به‌هیچ‌وجه جای تجربه انسان‌ها را نمی‌گیرد. کار کردن در تیم باعث می‌شود که خرد جمعی پشت یک پروژه افزایش پیدا کند. وقتی روی یک مسئله‌ای گیر می‌کنید، چقدر از وقتتان را هدر می‌دهید تا خودتان را از چاله‌ای که در آن افتاده‌اید بیرون بکشید؟ حالا به این فکر کنید که چقدر این تجربه متفاوت خواهد

بود اگر چند نفر از همکارانتان کنار شما باشند و در لحظه به شما بگویند کجا اشتباه کرده‌اید و چطور می‌توانید مسئله را حل کنید. دقیقاً به همین دلیل است که اعضای تیم کنار هم می‌نشینند (یا به صورت دو نفره برنامه‌نویسی می‌کنند).

بیشتر اوقات شما به یک جفت چشم دیگر هم نیاز خواهید داشت.

اجازه دهید یک مثال دیگر بزنم. فرض کنید شما در حال کدنویسی با یک زبان برنامه‌نویسی هستید که دارای کامپایلر^۱ است. آیا روزها می‌نشینید و ده هزار خط کد می‌نویسید و وقتی مطمئن شدید همه چیز را کامل و بی‌نقص تمام کردید، دکمه کامپایل را برای اولین بار فشار می‌دهید؟! البته که نه! چه فاجعه‌ای به بار می‌آید.

ما برنامه‌نویس‌ها با تغییرات کوچک کار می‌کنیم تا بتوانیم سریع نتیجه آن‌ها را ببینیم. یک تابع می‌نویسیم، کامپایل می‌کنیم، یک تست به آن اضافه می‌کنیم، دوباره کامپایل می‌کنیم، کمی کد را تغییر می‌دهیم، مجدداً کامپایل می‌کنیم. اشتباهات تایپی و اشکالات کد را بلافاصله حل می‌کنیم. دائماً باید به کامپایلر تکیه کنیم تا هوای ما را داشته باشد. با این کار می‌توانیم کیفیت کدمان را بالا ببریم و نرم‌افزار را کم‌کم بهتر و قوی‌تر کنیم.

این نیاز به پیشرفت کار در بخش‌های کوچک، نه فقط در سطح کدنویسی، بلکه در سطح کل پروژه نیز وجود دارد. تغییرات اجتناب‌ناپذیرند و پروژه‌ها باید دائماً خود را با این تغییرات تطبیق دهند. پروژه‌ها معمولاً با موانع از قبل پیش‌بینی نشده‌ای برخورد می‌کنند. گاهی اوقات کاملاً متوجه می‌شویم هیچ چیز طبق برنامه‌ای که در نظر داشتیم پیش نمی‌رود. نیازهای پروژه ناگهان تغییر می‌کنند. چطور ما می‌توانیم به سرعت برنامه‌ها و طرح‌هایمان را با تغییرات جدید تطبیق دهیم؟ جواب: با کار کردن به صورت تیمی.

اریک ریموند یک جمله معروف دارد:

«وقتی تعداد چشم‌ها زیاد شوند، باگ‌ها و ایرادهای نرم‌افزار نمایان می‌شوند.»

شاید این جمله را بتوان کامل‌تر و بهتر به این صورت بیان کرد: «وقتی تعداد چشم‌ها زیاد شوند، پروژه شما از مسیر درست خارج نمی‌شود.»

آدم‌هایی که داخل غار تنهایی خود کار می‌کنند، روزی بیرون می‌آیند و می‌بینند که با وجود اینکه کارشان را به اتمام رساندند، ولی دنیای بیرون غار تغییر کرده و ایده اولیه پروژه آن‌ها دیگر در این دنیای جدید معنی نمی‌دهد.

مهندسی و دفتر کار شخصی

بیست سال پیش، ذهنیت کلی آدم‌ها این بود که یک اتاق شخصی که بتوان در آن را بست، بهترین مکان برای یک مهندس است تا بتواند بالاترین کارایی را داشته باشد. فرض و گمان آن روزها این بود که این دفتر شخصی، باعث می‌شود یک برنامه‌نویس مدت زمان بیشتری برای خودش داشته باشد و فقط روی کد نوشتن تمرکز کند. ولی ما فکر می‌کنیم نه تنها اتاق و دفتر شخصی برای بیشتر مهندسی‌نرم‌افزار اصلاً ضروری نیست، بلکه حتی خطرناک هم هست. امروزه، نرم‌افزارها را تیم‌های مهندسی می‌نویسند، نه اشخاص. داشتن یک کانال ارتباطی قابل اتکا با سایر اعضای تیم، حتی از داشتن اتصال پرسرعت به اینترنت هم مهم‌تر و ضروری‌تر است. اگر تمام وقت دنیا را هم بدون هیچ مزاحمی داشته باشی، وقتی روی کار اشتباهی تمرکز کرده باشی، فقط وقت را تلف کرده‌ای.

متأسفانه به نظر می‌آید بعضی از شرکت‌های مدرن دنیای تکنولوژی، این روزها از آن سمت بام افتاده‌اند. وقتی وارد شرکت آن‌ها می‌شوید، ۵۰ یا حتی تا ۱۰۰ نفر را می‌بینید که در سالن‌های بسیار بزرگ در کنار هم کار می‌کنند. این طرح پلان آزاد و باز در دفاتر مهندسی این روزها موافق و مخالف‌های خودش را دارد. همه از کوچک‌ترین مکالمه‌ها خبردار می‌شوند و آدم‌ها برای صحبت کردن احساس راحتی نمی‌کنند، چون نگران هستند که باعث اذیت و آزار بقیه شوند. این مشکل هم به اندازه دفاترهای شخصی مضر است.

ما معتقدیم که حد وسط این دو حالت احتمالاً بهترین راه‌حل باشد. تیم‌ها را به گروه‌های ۶ تا ۱۲ نفره تقسیم کنید و هر تیم را در اتاق مخصوص خودشان قرار دهید. بدین صورت صحبت‌های خودجوش راحت‌تر شکل می‌گیرند و کسی از حرف زدن خجالت نمی‌کشد و باعث آزار دیگران هم نمی‌شوند. قطعاً در این حالت هم مثل هر حالت دیگری، تک‌تک افراد همچنان نیاز دارند که جلوی سروصداهای مزاحم را بگیرند. به همین دلیل است که خیلی از آدم‌ها راه‌حل‌های خلاقانه‌ای پیدا کردند تا بتوانند به بقیه نشان دهند که در کاری عمیق شدند و نمی‌خواهند کسی مزاحم آن‌ها شود. ما در تیمی کار می‌کردیم و بین خودمان یک قانون گذاشته بودیم که هر موقع هر کسی با فرد دیگری کار داشت، اسمش را صدا می‌کرد و می‌گفت: «چند دقیقه وقت داری؟»^۱ اگر فرد مورد نظر فرصت صحبت کردن داشت با صدای خود می‌چرخید و با هم صحبت می‌کردند و اگر فرصت نداشت، می‌گفت: «چند لحظه فقط»^۲ به این معنی که در اولین فرصتی که پیدا می‌کرد می‌آمد و با همکاری صحبت می‌کرد. تیم‌های دیگری را هم دیدیم که از هدفون‌های مخصوص کم کردن صدای محیط^۳ استفاده می‌کردند تا سروصدای محیط پیرامون را به حداقل برسانند. به طور کلی، گذاشتن هدفون روی سر می‌تواند یک نشانه تلقی شود به این معنی: «الان مزاحم من نشوید، مگر اینکه کارتان خیلی واجب باشد.» حتی تیم‌هایی را هم دیدیم که

1. Breakpoint
2. Ack
3. Noise Cancellation

از عروسک یا وسایل دیگری استفاده می‌کردند تا نشان دهند که در حال حاضر سرشان شلوغ است و یا در کاری عمیق غرق شده‌اند.

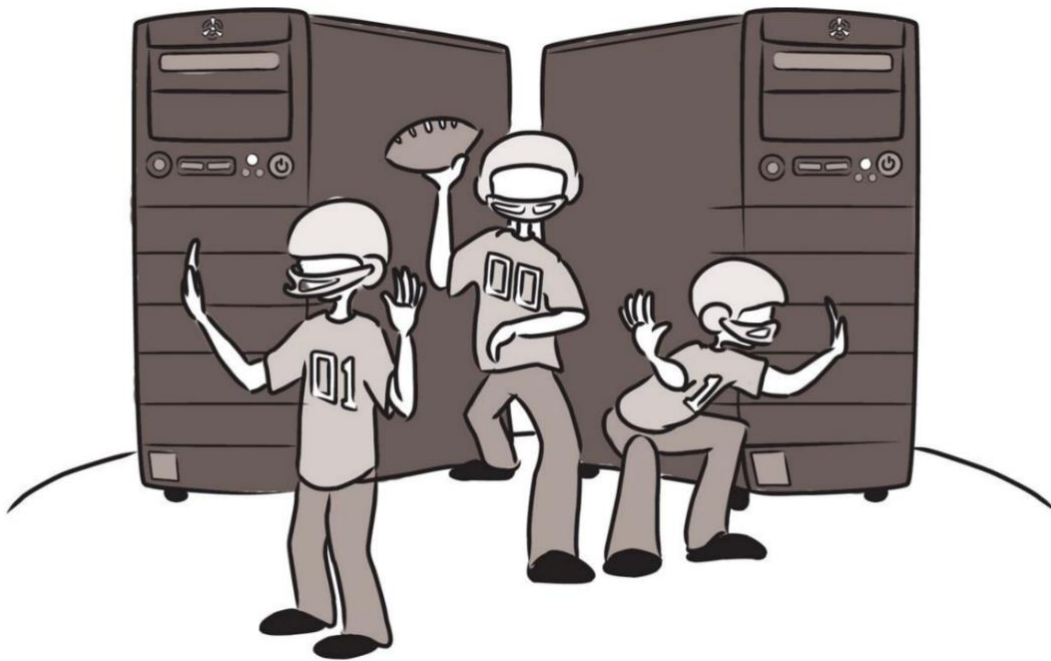
اشتباه برداشت نکنید! بله ما هم باور داریم که برنامه‌نویس‌ها نیاز به بازه‌های زمانی بدون وقفه و مزاحم دارند تا در نوشتن کدهای برنامه‌نویسی عمیق شوند. ولی در عین حال همچنان انتظار داریم که آن‌ها یک کانال ارتباطی قوی و همیشه در دسترس با سایر هم‌تیمی‌های خودشان داشته باشند. هنر واقعی، پیدا کردن حد وسط بین این دو نیاز است.

خوب نتیجه این شد: «ریسکِ تنهایی کار کردن ذاتاً از کارِ گروهی بیشتر است.» به جای اینکه نگران دزدیدن ایده‌تان یا نحوهٔ قضاوت کارتان باشید، باید بیشتر از این بترسید که وقتتان را روی یک کار اشتباه تلف می‌کنید. متأسفانه، مسئلهٔ مخفی کردن ایده‌ها فقط محدود به رشتهٔ مهندسی نرم‌افزار نیست. این یک مشکل فراگیر در همهٔ رشته‌هاست. برای مثال، دنیای علم و دانش مثلاً قرار است که یک دنیای باز و آزاد از همکاری و تبادل اطلاعات باشد، ولی استیصال نیاز برای چاپ مقاله و رقابت برای دستیابی به بودجه‌های تحقیقاتی باعث شده است تا نتیجهٔ عکس به دست بیاید. متفکران بزرگ ایده‌های خود را با بقیه تقسیم نمی‌کنند. به شدت در تلاش‌اند تا کارهایشان را از دیگران پنهان کنند و در غار تنهایی خود به تحقیق می‌پردازند و از ایرادهای کارشان بی‌خبر می‌مانند. نهایتاً مقالهٔ خودشان را چاپ می‌کنند و طوری وانمود می‌کنند که انگار پشت کارشان تلاش و کوشش بی‌وقفه‌ای نبوده است. نتایج عموماً فاجعه‌بار هستند: یا به طور اتفاقی کار یک نفر دیگر را تکرار کرده‌اند، یا کارشان ایرادهایی دارد که متوجهش نبودند و یا نهایتاً کاری را عرضه کرده‌اند که هر چند زمان شروعش موضوعی ناب و جالب بوده است، ولی الان دیگر بی‌معنی و بی‌فایده است. مقدار زمانی که در این کار تلف می‌شود، بسیار غم‌انگیز است. شما یکی مثل بقیه نباشید!

مهم، تیم است!

خوب اجازه دهید به عقب برگردیم و تمام این ایده‌هایی را که مطرح کردیم روی هم بگذاریم. اصل صحبت ما این است که در دنیای برنامه‌نویسی، فردگرایی خیلی به ندرت پیدا می‌شود. حتی اگر هم وجود داشته باشد، افراد فردگرا هیچ‌وقت به تنهایی به موفقیت‌های خارق‌العاده دست پیدا نمی‌کنند. پشت سر دستاوردهایی که توانسته‌اند دنیا را با آن تکان دهند، جرقه‌ی یک ایده‌ی ناب به همراه تلاش قهرمانانه‌ی یک تیم منسجم نهفته است. هدف واقعی، ساختن یک تیم از فوق ستاره‌هاست و این کار بسیار سخت است. بهترین تیم‌ها آن‌هایی هستند که از فوق ستاره‌هایی که دارند، هوشمندانه استفاده می‌کنند. عملکرد یک گروه به عنوان یک تیم، همیشه از حاصل جمع کارایی تک‌تک افراد حاضر در تیم بیشتر است.

به زبان ساده‌تر، برنامه‌نویسی یک کار تیمی است. ممکن است درک این موضوع سخت باشد، چرا که تمام پیش‌فرض‌های ما از قهرمان‌سازی نابغه‌های دنیای کامپیوتر را به هم می‌ریزد؛ ولی مثل یک شعار این جمله را تکرار می‌کنیم: «برنامه‌نویسی، یک کار تیمی است!»



وقتی در لانه‌ی تنهایی خودتان کدنویسی کنید، نابغه بودن کافی نیست. هیچ‌وقت با قایم شدن و کار پنهانی روی یک پروژه سری، دنیا را تکان نخواهید داد و نتیجه‌ی کارتان بقیه را شگفت‌زده نخواهد کرد. لازم است تا حتماً با

بقیه کار کنید. رؤیا و هدف خودتان را با دیگران در میان بگذارید. کار را بین افراد تقسیم کنید و از نحوه کار همکارهایتان چیزهای جدید یاد بگیرید.

چه تعداد نرم‌افزار موفق یا قطعه‌ای از نرم‌افزار را می‌توانید در دنیا نام ببرید که خیلی از آدم‌ها به صورت گسترده از آن استفاده می‌کنند، ولی حقیقتاً فقط و فقط یک نفر آن را نوشته باشد؟^۱ این شعار را که برنامه‌نویسی یک کار تیمی است، بارها در بخش‌های مختلف این کتاب تکرار خواهیم کرد. یک تیم منسجم و کارآمد مثل طلا نایاب است و کلید اصلی دروازه موفقیت به شمار می‌آید. تمام تلاش شما باید در راستای رسیدن به این هدف باشد. و این دقیقاً موضوع اصلی کتابی است که در دست شماست.

^۱. ممکن است یک نفر بگوید LATEX! ولی خوب به سختی می‌توان گفت که LATEX را تعداد زیادی از افراد در دنیای نرم‌افزار استفاده می‌کنند. مگر اینکه نویسندگان مقالات علمی را هم در دنیای کامپیوتر و برنامه‌نویسی لحاظ کنیم.

ستون‌های سه‌گانه

پیش‌تر، منظورمان را از کار تیمی در مهندسی نرم‌افزار بیان کردیم. اگر کار تیمی، بهترین راه برای تولید قوی‌ترین نرم‌افزارهاست، چطور می‌توان یک تیم خوب را درست یا پیدا کرد؟ کار ساده‌ای نیست. برای رسیدن به مدینه فاضله در کار تیمی، ابتدا لازم است سه مهارت اجتماعی را که ما به آن‌ها «ستون‌های سه‌گانه» می‌گوییم به خوبی یاد بگیرید و به آن عمل کنید. این سه مهارت، فقط عامل روغن کاری چرخ‌های روابط اجتماعی نیستند، بلکه شالوده و بنیادی هستند که روابط تیمی سالم روی آن‌ها استوار می‌شوند.

تواضع:

- شما مرکز دنیا نیستید. شما نه معصوم هستید و نه همه‌چیزدان. شما دیدِ بازی نسبت به بهبود شخصیت خود دارید.

احترام:

- شما از ته دل به همکارانتان اهمیت می‌دهید. با آن‌ها مثل انسان رفتار می‌کنید و قدر مهارت‌ها و دستاوردهای آن‌ها را می‌دانید.

اعتماد:

- اطمینان دارید که سایرین، انسان‌های شایسته‌ای هستند که کار خودشان را بلدند. شما به آن‌ها این فرصت را می‌دهید که هر جا مناسب بود اختیار امور را در دست بگیرند.

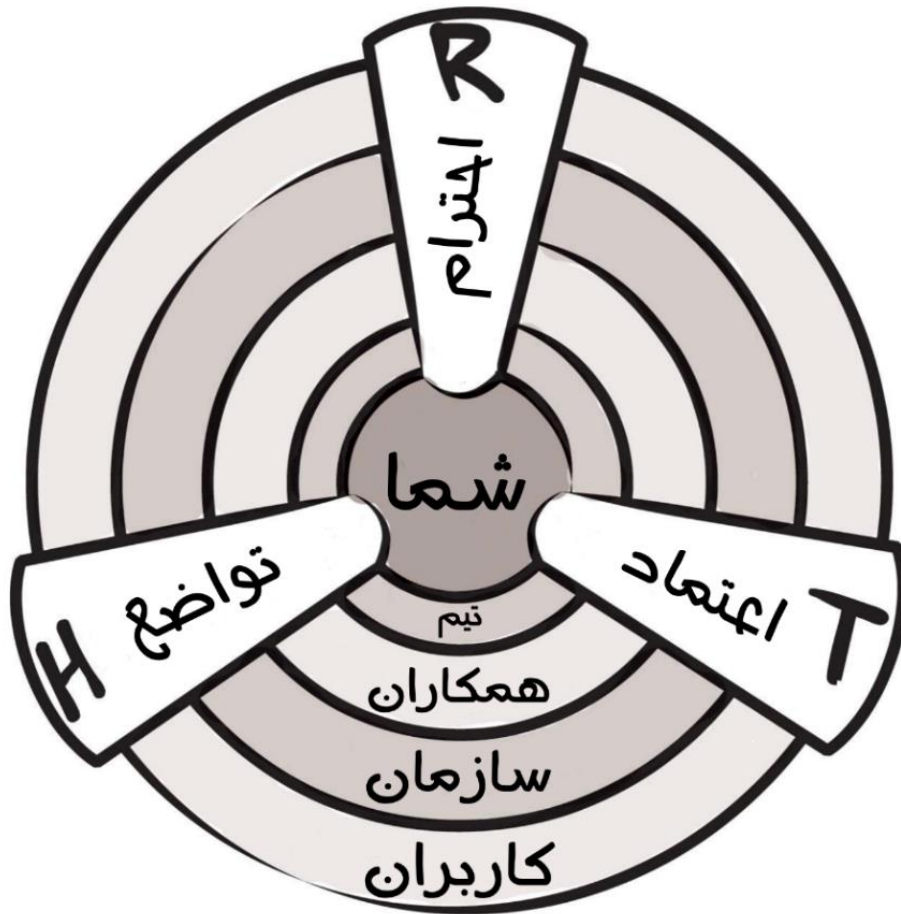
به مجموعه این اصول سه‌گانه، به اختصار *HRT*¹ می‌گوییم و آن را هارت (Heart به معنی قلب) تلفظ می‌کنیم و نه هرت (Hurt به معنی آسیب و درد). چرا که هدف از این اصول، کاهش درد و آسیب‌رسانی به یکدیگر است.

قضیه‌ای که ما در طول این کتاب سعی در اثباتش داریم به همین سه ستون برمی‌گردد: تقریباً ریشه تمام درگیری‌های ارتباطی در تیم‌ها را می‌شود در عدم یا کمبود حداقل یکی از اصول تواضع، احترام و یا اعتماد پیدا کرد. ممکن است ادعای سنگینی به نظر بیاید، ولی بیشتر به آن فکر کنید. یک رابطه اجتماعی زشت را که باعث ناراحتی شما در زندگی شده به خاطر بیاورید. در بطن این مجادله، آیا همه طرفین به اندازه کافی فروتنی و تواضع

1. حروف ابتدای کلمه انگلیسی برای هر یک از این سه ستون. یعنی: Humility, Respect, Trust

به خرج داده‌اند؟ آیا رفتار آدم‌ها با هم بر پایهٔ احترام و ملایمت بوده است؟ آیا بین افراد دخیل در صحبت، احترام دوطرفه‌ای وجود داشت؟

این اصول برای ما آن‌چنان حائز اهمیت هستند که کل کتاب را بر اساس آن‌ها طبقه‌بندی کرده‌ایم. کتاب با تمرکز روی شما شروع می‌شود تا شما را قانع کند اصول HRT را درون خودتان تقویت کنید و آن‌ها را مرکز توجهتان در ارتباطاتتان قرار دهید. فصل اول کتاب در همین راستاست. در فصل دوم دربارهٔ ساختن یک تیم بر اساس ستون‌های سه‌گانهٔ HRT صحبت خواهیم کرد. سپس دربارهٔ ساختن فرهنگ تیمی صحبت می‌کنیم که نقطهٔ عطفی در فرایند ساختن یک تیم رؤیایی است. در مرحلهٔ بعدی تمرکزمان را روی آن دسته از همکارهایی می‌گذاریم که اگرچه ممکن است عضوی از هستهٔ اصلی تیم ما نباشند، ولی به طور روزانه با ما در ارتباط هستند. مثلاً همکارانی که در تیم‌های دیگر کار می‌کنند یا داوطلبانی که صرفاً به پروژهٔ ما کمک می‌کنند. خیلی از این افراد ممکن است نه تنها به اصول HRT پایبند نباشند، بلکه حتی شاید به طور اصولی انسان‌های سمی و خطرناکی باشند! اولین قدم این است که یاد بگیرید چطور تیمتان را در مقابل این دسته از آدم‌ها محافظت کنید. هدف نهایی خارج کردن نیش و دندان این آدم‌ها از فرهنگ تیمی شماست که خود این کار می‌تواند یک راه‌حل عالی برای گسترش تیمتان باشد.



خیلی از تیم‌ها، عضوی از یک شرکت بزرگ هستند که خود محیط و فرهنگ شرکت می‌تواند برایشان به اندازه آدم‌های سمی، دست و پاگیر و مخرب باشد. اینکه بلد باشیم چطور بین موانع بازدارنده محیط یک شرکت مانور بدهیم، می‌تواند مستقیماً روی موفقیت یا عدم موفقیت تولید یک محصول نرم‌افزاری تأثیر بگذارد.

و در نهایت نباید کاربران نرم‌افزارتان را فراموش کنید. گاهی خیلی راحت وجود آن‌ها را از یاد می‌بریم ولی کاربران، خون جاری در رگ‌های پروژه شما هستند. نرم‌افزار شما بدون افرادی که از آن استفاده کنند، هیچ هدف و مقصودی نخواهد داشت. تمام اصول HRT که باید بین افراد تیمتان رعایت کنید، دقیقاً در نحوه ارتباط شما با مشتری‌هایتان نیز صدق می‌کند. فواید این روش بازخورد با مشتری، بسیار زیاد خواهد بود.

یک لحظه توقف کنید!

احتمالاً وقتی این کتاب را برداشتید، انتظار نداشتید که در حال ثبت نام در یک گروه حمایتی باشید که هر هفته جلسات دوره‌می برگزار می‌کنند؛ درک می‌کنیم. سروکله زدن با مشکلات روابط اجتماعی گاهی اوقات

می‌تواند سخت باشد. تعامل با دیگر انسان‌ها، گاهی آشفته، شلوغ، غیرقابل پیش‌بینی و حتی آزاردهنده است. به جای تلاش برای برنامه‌ریزی و سیاست‌مداری در روابط اجتماعی، آدم وسوسه می‌شود کلاً قضیه را فراموش کند. وقتی آدم می‌تواند با یک کامپایلر منطقی و قابل پیش‌بینی وقت بگذراند، چرا باید خودش را درگیر مسائل روابط اجتماعی بکند؟

توجه شما را به بخشی از سخنرانی معروف ریچارد همینگ^۱ جلب می‌کنم:

«اگر همیشه با منشی‌ام جوری رفتار کنم که هم محترمانه باشد و هم او را سر حال نگه دارد، کمک و سرویس فوق‌العاده بهتری دریافت می‌کنم. مثلاً یک روز به یک دلیل احمقانه تعدادی از افرادی که کارهای خدماتی در موری هیل^۲ ارائه می‌دادند، دیگر کار خود را انجام ندادند؛ نپرسید چرا، کار نمی‌کردند. من هم نیاز داشتم یک سری از کارها حتماً انجام شود. منشی من با یک نفر در هالم‌دل^۳ تماس گرفت، سوار ماشین شرکت شد، یک ساعت رانندگی کرد و کار مرا راه انداخت و برگشت. به نوعی جواب همه رفتارهای محترمانه من با او و تلاش‌های من برای خندانندش را داد. به وسیله یادگیری و مطالعه دقیق یک سیستم، شما یاد می‌گیرید که چطور کاری کنید که سیستم در راستای اهداف شما کار کند.»

نصیحتی که در این گفته نهفته این است: «قدرت روابط اجتماعی را دست‌کم نگیرید.» در مورد فریب‌کاری یا گول زدن افراد صحبت نمی‌کنیم. در مورد ساختن روابط مفید اجتماعی در راستای انجام دادن کارها صحبت می‌کنیم. همیشه عمر دوستی‌ها بیشتر از عمر پروژه‌ها خواهد بود.

1. Richard Hamming - "You And Your Research" <http://www.cs.virginia.edu/~robins/YouAndYourResearch.pdf>
2. Murray Hill
3. Holmdel

اصول HRT در عمل

شاید تا الان این طور به نظر رسیده باشد که ما روی منبر رفته‌ایم و در مورد تواضع، احترام و اعتماد فقط حرف زده و موعظه کرده‌ایم. باید دید که چطور می‌توان این صحبت‌ها را در زندگی روزمره به واقعیت تبدیل کرد. در این بخش، به دنبال راهکارهای عملی هستیم تا به وسیله آن‌ها یک سری رفتارهای شخصی را بازبینی کنیم. ممکن است خیلی از مثال‌هایی که می‌آوریم به نظر واضح و مبرهن بیایند، ولی وقتی که در آن‌ها عمیق شوید متوجه می‌شوید که بیشتر اوقات شما یا همکارانتان خطا کار هستید.

غرورت را کنار بگذار

این جمله با زبان بی‌زبانی عدم تواضع را به روی آدمی می‌آورد که خیلی رفتارهای فروتنانه‌ای ندارد. هیچ‌کس دوست ندارد با کسی همکار باشد که دائماً فکر می‌کند مهم‌ترین انسان در اتاق است. حتی اگر واقعاً شما باهوش‌ترین آدم در یک بحث باشید، لازم نیست مدام هوشتان را در چشم سایرین فرو کنید. به عنوان مثال، آیا در مکالمه‌ها دوست دارید همیشه حرف اول یا حرف آخر را شما بزنید؟ آیا فکر می‌کنید لازم است تا درباره هر موضوع و مبحثی نظر بدهید؟ یا شاید احتمالاً آدمی را می‌شناسید که این‌طور باشد.

توجه کنید که تواضع و فروتنی به این معنی نیست که اجازه دهید بقیه سوار شما بشوند. اعتماد به نفس هیچ اشکالی ندارد. فقط مثل آدم‌های همه‌چیزدان رفتار نکنید. همیشه سعی کنید تمرکزتان روی برانگیختن غرور تیمی باشد، نه فردی. به جای اینکه شخصاً نشان دهید آدم فوق‌العاده‌ای هستید، سعی کنید حس موفقیت و غرور تیمی را بین همکارانتان تقویت کنید. یک مثال خوب در این راستا، بنیاد نرم‌افزار آپاچی است که کارنامه بلندبالایی در ساخت تیم‌های منسجم در تولید نرم‌افزارهای مختلف دارد. تیم‌های این بنیاد به صورت فوق‌العاده‌ای به همکاری و کار گروهی خود می‌بالند و اجازه نمی‌دهند که یک شخص با فردگرایی برای پز دادن از تیم سوءاستفاده کند.

غرور، خودش را به روش‌های زیاد و متفاوتی نشان می‌دهد و بیشتر اوقات مانع کارایی شما می‌شود. به قسمتی دیگر از سخنرانی ریچارد همینگ در همین زمینه توجه کنید:

«جان تاکی^۱ همیشه خیلی معمولی لباس می‌پوشید. وقتی وارد دفتر کار مهمی می‌شد، ممکن بود مدت‌ها طول بکشد تا بقیه متوجه شوند که او چه شخص مهمی است و بهتر است به صحبت‌های او گوش دهند. مدتی طولانی، جان مجبور بود با این طرز رفتار بقیه کنار بیاید. غرور و تکبر کار بیهوده‌ای هست. من هیچ‌وقت نگفته‌ام که جلوی دیگران کوتاه بیایید. بحث من این است که اگر با بقیه کنار بیایید و خودبزرگ‌بینی را کنار بگذارید، حتماً سودش را می‌بینید. ولی برعکس اگر غرور و تکبر را انتخاب کنید و معتقد باشید که همیشه حرف حرف شماست، به مرور در طول عمر کاری خود هزینه‌اش را می‌پردازید... یادگیری سیستم‌های متنوع به شما کمک می‌کند که از آن‌ها در راستای اهداف و نیازهای خودتان استفاده کنید. اگر نه، مجبورید دائماً در جنگ با سیستم‌های دور و برتان باشید.»

۱. John Tukey؛ ریاضیدان معروف آمریکایی و مبدع الگوریتم Fast Fourier Transform

نحوه انتقاد کردن و انتقادپذیری

داستانی در مورد شخصی به نام جو که به تازگی کار جدیدی را تحت سِمَت برنامه‌نویس آغاز کرده بود، مطرح می‌کنیم. او همان هفته اول شدیداً روی کد نرم‌افزار و پروژه تمرکز کرد. از آنجا که برای کارش اهمیت بالایی قائل بود و به نحوه برنامه‌نویسی دقت داشت، به مرور از هم‌تیمی‌های خودش در مورد کارشان سؤال کرد. محترمانه در مورد نحوه طرز فکر و پیش‌فرض‌های آن‌ها اطلاعات بیشتری درخواست کرد و تکه کدهایی را برایشان ایمیل می‌کرد تا هم نظرشان را بپرسد و هم مؤدبانه پیشنهاد بدهد که چطور می‌توانند منطق کدشان را بهتر کنند. چند هفته که گذشت، به دفتر رئیس بخش احضار شد. جو با تعجب پرسید: «مشکل چیست؟ من چه کار اشتباهی انجام دادم؟»

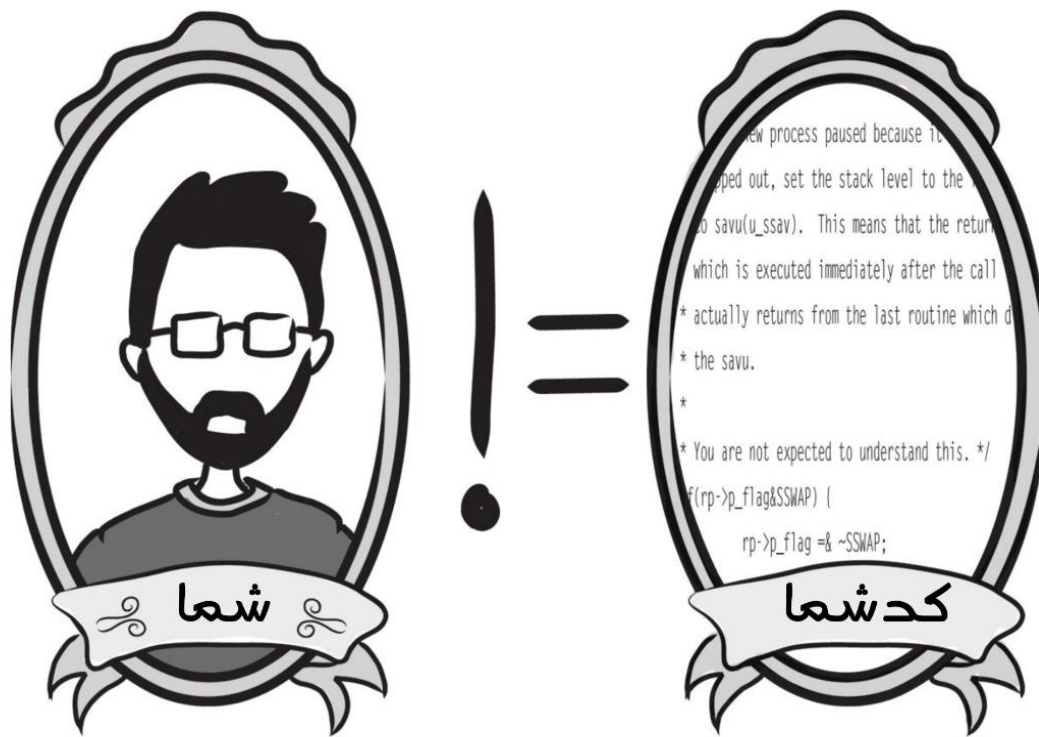
رئیسش با نگرانی گفت: «ما شکایت‌های زیادی در مورد تو شنیدیم جو؛ به نظر می‌آید که رفتار تو با همکاران خیلی تند و خشن است. چپ و راست از آن‌ها ایراد می‌گیری، از دست تو ناراحت‌اند. لازمه که کمی ملایم‌تر باشی!»

جو مات و مبهوت بود. در یک محیط مناسب و بر پایه اصول HRT، همکاران جو باید با خوشحالی از نظرات و پیشنهادهای جو استقبال می‌کردند. ولی در این مورد خاص، به نظر می‌آمد که جو باید به خاطر عدم اعتماد به نفس همکارهای خودش ملایم‌تر رفتار می‌کرد و نظرات و پیشنهادهای خودش را به صورت لطیف‌تری مطرح می‌کرد.

در یک محیط حرفه‌ای مهندسی نرم‌افزار، انتقادات هیچ‌وقت شخصی نیستند؛ بلکه بیشتر اوقات به خاطر بهتر شدن محصول نهایی مطرح می‌شوند. نکته ریز اساسی این است که شما و همکارانتان بتوانید یک انتقاد سازنده در مورد یک کار خلاقانه را از توهین به شخصیت طرف، تشخیص دهید. حمله به شخصیت آدم‌ها هیچ سودی ندارد و تقریباً غیرممکن است که آدم بتواند بابت آن کاری انجام دهد؛ ولی انتقاد سازنده، باعث می‌شود تا پیشرفت بیشتری بکنید. مهم‌تر از همه، یک انتقاد سازنده همیشه محترمانه مطرح می‌شود، چرا که بر اساس علاقه به بهبود و پیشرفت کار طرف مقابل خواهد بود. اگر واقعاً شخصی برای شما محترم باشد، همیشه در انتخاب واژه‌ها و عبارات دقت می‌کنید. نزاکت در انتخاب کلمات و جملات مفید در انتقاد و پیشنهاد، مهارتی است که فقط با تمرین زیاد به دست می‌آید.

در مقابل، شما نیز لازم است هنر انتقادپذیری را یاد بگیرید. یعنی نه تنها لازم است تا در مهارت و توانایی‌های خود فروتن باشید، بلکه باید اعتماد داشته باشید که همکارانتان قلباً خیر و صلاح و پیشرفت شما (و البته کار و پروژه) را در نظر دارند. برنامه‌نویسی مهارتی است که مثل هر توانایی دیگری با تمرین و ممارست بهتر می‌شود. اگر همکاران در مورد فرضاً توانایی شعبده‌بازی شما پیشنهاد یا نظری ارائه می‌داد، آیا این نظر را یک حمله شخصی به هویت خود تلقی می‌کردید؟ فکر می‌کنیم، یعنی امیدواریم که این‌طور نباشد. برای مثال، کدی که شما می‌نویسید یا محصول خلاقانه‌ای که خلق می‌کنید برابر با شخصیت و ارزش هویت شما نیست. تکرار می‌کنیم:

«شما، کدی نیستید که می‌نویسید!» شما هم این را تکرار کنید: «شما، چیزی که خلق می‌کنید نیستید.» نه تنها لازم است که خودتان این قضیه را باور کنید، بلکه باید همکارانتان را هم قانع کنید که انسان‌ها، کدی که می‌نویسند، نیستند.



برای نمونه، این یک شیوه غلط حرف زدن با همکاری است که اعتماد به نفسش پایین است:

«بابا کل منطقی که تو روی این متد پیاده‌سازی کردی اشتباهه، باید از فلان مدل استاندارد که همه هم

می‌دونند استفاده می‌کردی.»

این نحوه انتقاد، سرشار از الگوهای اشتباه است: اولاً می‌گویید کارش کلاً «اشتباه» است (انگار که دنیا سیاه و سفید است). طلبکارانه از او می‌خواهید کاری را که انجام داده مطابق نظر شما تغییر دهد و او را متهم می‌کنید چیزی را ساخته که در تضاد با باور و اعتقاد همه آدم‌هاست (حس حماقت به او القا می‌کنید). جوابی که دریافت خواهید کرد، یک عکس‌العمل عموماً احساسی و تدافعی خواهد بود.

روش بهتر برای انتقال همان منظور ولی با کلمات مناسب‌تر می‌تواند این‌طور باشد:

«بین این منطقی که توی این متد پیاده‌سازی شده کمی من رو گیج کرده. به نظرم شاید اگه فلان مدل پیاده‌سازی بشه باعث خوانایی بهتر کد و نگهداری راحت‌تر بشه.»

دقت کنید که چطور با فروتنی، مشکل را به خودتان مرتبط می‌کنید نه به شخص مقابل. او اشتباهی مرتکب نشده، بلکه این شما هستید که در فهم کد او دچار مشکل شدید. پیشنهاد شما صرفاً برای این است که طرف مقابل به شما کمک کند تا کد را بهتر متوجه شوید و احتمالاً پروژه نیز راحت‌تر به اهداف بلندمدت خود برسد. همچنین، شما از شنونده، طلبکار نیستید. به همکاران این فرصت را می‌دهید که با صلح و آرامش نظر و پیشنهاد شما را قبول نکنند. به این شکل موضوع بحث حول محور کد باقی می‌ماند و وارد موضوع شخصیت و هنر برنامه‌نویسی افراد نمی‌شود.

یادگیری سریع از اشتباهات

یک داستان معروف و شاید کلیشه‌ای در دنیای بازار، در مورد مدیری وجود دارد که با یک اشتباه بزرگ باعث می‌شود شرکتی که در آن کار می‌کند ۱۰ میلیون دلار ضرر کند. روز بعد با ناراحتی وارد دفتر شده و شروع به جمع کردن وسایلی می‌کند. همان طور که انتظار داشت، به او می‌گویند مدیرعامل در دفترش منتظر اوست. با ناراحتی و خیلی آرام وارد دفتر شده و یک برگ کاغذ به مدیرعامل تحویل می‌دهد. وقتی مدیرعامل می‌پرسد این چیست؟ جواب می‌دهد: «استعفانامه من. مگه نمی‌خواین من رو اخراج کنین؟» مدیرعامل در کمال ناباوری پاسخ می‌دهد: «اخراج کنم؟! من همین الان ۱۰ میلیون دلار خرج تجربه و آموزش تو کردم. حالا اخراجت کنم؟!»^۱

مطمئناً اغراق در این داستان زیاد است، ولی نکته این است که مدیرعامل این داستان به خوبی می‌داند اخراج مدیری که اشتباه کرده، نه تنها ۱۰ میلیون دلار ضرر را بر نمی‌گرداند، بلکه باعث می‌شود شرکت یک مدیر خوب را نیز از دست بدهد که به خاطر تجربه‌ای که به دست آورده، مطمئناً اشتباهش را تکرار نخواهد کرد.

یکی از شعارهای مورد علاقه ما در گوگل این بود: «گزینه شکست، همیشه روی میز موجوده.» باور شرکت این بود که اگر شما هر از گاهی شکست نخورید، یعنی به اندازه کافی در کارتان ابتکار نداشتید و ریسک نکردید. شکست خوردن، یک فرصت طلایی است برای یادگیری و بهتر شدن در تلاش‌های بعدی. یک جمله که به توماس ادیسون نسبت داده شده، این طور می‌گوید: «من اگر ده هزار راه پیدا کنم که هیچ کدام به نتیجه نرسند، شکست نخورده‌ام و دل سرد نمی‌شوم، چون هر تلاش اشتباهی که تمام می‌شود، یک قدم من را به هدف نزدیک‌تر می‌کند.»

در بخش گوگل ایکس^۲، که بیشتر پروژه‌های آینده‌نگرانه مثل گوگل گلس^۳ و ماشین‌های خودران در آنجا انجام می‌شود، شکست به صورت عمدی، جزئی از سیستم تشویقی پیش‌بینی شده است. اعضای تیم‌ها مدام ترغیب می‌شوند تا ایده‌های دیوانه‌وار همدیگر را به چالش بکشانند. هر کس به تعداد دفعاتی که در طول یک زمان مشخص، عدم موفقیت یک ایده جدید را ثابت می‌کند، پاداش می‌گیرد و حتی با سایرین رقابت می‌کند. فقط وقتی هیچ‌کس نمی‌تواند تحت هیچ شرایطی یک ایده جدید را رد کند، ایده وارد مرحله پیاده‌سازی اولیه می‌شود.

راز یادگیری از اشتباهات این است که تجربیات را خوب و دقیق ثبت کنید. در دنیای کسب و کار به این نوع بررسی علت و معلول وقوع یک اشتباه، «کالبدشکافی» گفته می‌شود. کالبدشکافی‌های اشتباهات خود را مکتوب کنید. این کار را با دقت مضاعفی انجام دهید و اطمینان حاصل کنید که نوشته شما یک لیست به‌دردنخور از عذر و بهانه‌های الکی نباشد. یک کالبدشکافی مناسب، همیشه درس‌هایی را که آدم از اشتباهاتش یاد می‌گیرد دقیق

۱. روایت‌های زیادی از این داستان مطرح شده که به شرکت‌ها و مدیرعامل‌های متفاوتی نسبت داده شده‌اند.

توضیح می‌دهد و کارهایی را که لازم است در نتیجه این دروس، شروع شوند یا تغییر کنند لیست می‌کند. حتماً این نوشته را جایی در دسترس قرار داده و مطمئن شوید کارهایی که نوشتید پیگیری خواهند شد. به یاد داشته باشید که نوشتن اشتباهات کمک می‌کند تا سایرین نیز (چه الان و چه در آینده) از اشتباهات شما درس بگیرند و تاریخ را تکرار نکنند. ردپاهای خودتان را پاک نکنید. برعکس آن‌ها را شفاف و پررنگ کنید تا راهنمایی برای بقیه باشند.

گزارش کالبدشکافی خوب، شامل موارد زیر است:

- تاریخچه اتفاقات: از زمان پیدا شدن اشتباه یا حادثه تا ارائه راه‌حل
- عامل اولیه اتفاق یا اشتباه
- برآورد دقیق هزینه و آسیب ایجاد شده
- مجموعه موارد تحت اقدام سریع برای غلبه فوری بر مسئله
- مجموعه موارد جهت اقدام برای جلوگیری از اشتباه مشابه در آینده
- درس‌هایی که از اتفاق گرفته شده

برای یادگیری، زمان بگذارید

سیندی یک ستاره بود. یک مهندس نرم‌افزار فوق‌العاده که حقیقتاً در کارش استاد شده بود. به راحتی به عنوان مدیرفنی پروژه ارتقا یافت و مسئولیت‌های بیشتری را روی دوش خودش احساس کرد. خیلی زود، مشاور و مربی اعضای تیمش شد و به آن‌ها راه و روش مهندسی را آموزش می‌داد. در کنفرانس‌ها و سمینارهای مختلف سخنرانی می‌کرد و به آرامی مدیریت چند تیم مختلف دیگر را نیز به عهده گرفت. اگرچه از اینکه در کارش خبره شده بود لذت می‌برد، ولی اندک‌اندک در کارش بی‌حوصله شد. یک جایی در این مسیر، یادگیری موضوعات جدید را فراموش کرده بود. حس باهوش‌ترین و ماهرترین بودن در هر جمعی، آرام‌آرام تازگی و طراوت خود را برای او از دست داد. با وجود تمام نشانه‌های موفقیت، جای یک چیزی هنوز خالی بود. یک روز که مشغول کار بود، متوجه شد تخصصی که دارد دیگر خیلی در دوره جدید به کار نمی‌آید. همه درگیر موضوعات تازه‌تر شده بودند و دنبال تخصص‌های جدیدتر بودند. سیندی چه اشتباهی مرتکب شده بود؟

تعارف که نداریم، خیلی خوش می‌گذرد وقتی آدم داناترین فرد یک جمع باشد. مربیگری برای بقیه می‌تواند واقعاً ارزشمند باشد، ولی مشکل اینجاست که وقتی شما به یک ماکسیمم نسبی در تیمتان دست پیدا می‌کنید، یادگیری را کنار می‌گذارید و وقتی یادگیری را فراموش کنید، حوصله شما در کار سر خواهد رفت. عادت به رهبری تیم خیلی راحت تبدیل به یک اعتیاد می‌شود. فقط با تواضع و فروتنی می‌توان مسیر را عوض کرد و راه را برای یادگیری موضوعات جدید باز کرد. مجدداً اینجا هم تواضع و تمایل برای یادگیری است که اهمیت دارد. از حاشیه امن خودتان خارج شوید و خودتان را برای یادگیری و پیشرفت به چالش بکشانید. در طولانی‌مدت آدم خوشحال‌تری خواهید بود.

یاد بگیرید صبور باشید

سال‌ها پیش فیتز روی پروژه‌ای برای تبدیل مخزن‌های کد CVS به کدهای مخرب و بعدها گیت^۱ کار می‌کرد. به قدری CVS بدقلق بود که فیتز مدام با باگ‌های جدید مواجه می‌شد. یکی از دوستان قدیمی فیتز به اسم کارل، قبلاً روی CVS خیلی کار کرده بود و به خوبی با آن آشنا بود. این شد که فیتز و کارل تصمیم گرفتند با هم روی این پروژه کار کنند و باگ‌ها را برطرف کنند.

وقتی که دونفره برنامه‌نویسی را شروع کردند، با یک مشکل مواجه شدند: فیتز از پایین به بالا به مسائل نگاه می‌کرد. او با سر درون هر مشکلی شیرجه می‌زد و با سعی و خطا و امتحان روش‌های مختلف سعی می‌کرد تا قضیه را حل کند. کارل اما از بالا به پایین صورت مسائل را می‌شکافت. او سعی می‌کرد به همه متدها و کدهای موجود نگاه کلی داشته باشد و سپس مشکل را پیدا کند. نتیجه این تفاوت در نگاه، باعث کشمکش‌های مختلف و گاهی اوقات جر و بحث‌های شدید بین این دو نفر شد. به قدری این مشکل برای آن‌ها خسته‌کننده شد که دیگر حاضر نبودند هم‌زمان و دونفره برنامه‌نویسی کنند. باید بگوییم که آن دو دوست‌های بسیار قدیمی بودند که برای همدیگر اعتماد و احترام خیلی بالایی قائل بودند. این موضوع در کنار صبر و بردباری بالای آن‌ها باعث شد تا بتوانند راه دیگری برای همکاری پیدا کنند. تصمیم گرفتند با هم و دونفره کدنویسی کنند تا زمانی که به یک باگ برخورد کنند؛ آن وقت برای حل آن باگ، از هم جدا می‌شدند تا هر کس تنهایی روی مسئله کار کند. وقتی مشکل حل می‌شد، مجدداً برنامه‌نویسی دونفره خودشان را ادامه می‌دادند. صبوری این دو نفر، همراه با تمایلشان برای همکاری، نه تنها سرنوشت پروژه را نجات داد، بلکه دوستی آن‌ها را نیز پایدار کرد.

۱. Git؛ یک نوع نرم‌افزار که به تیم اجازه می‌دهد به صورت هم‌زمان روی یک پروژه کار کنند و در عین حال هیچ مشکل و تداخلی برای پروژه پیش نیاید. (و.)

تأثیرپذیر باشید

هرچه آدم تأثیرپذیرتری باشید، شخص تأثیرگذارتری نیز خواهید بود. هرچه آسیب‌پذیرتر باشید، قوی‌تر به نظر خواهید آمد. این جمله‌ها در نگاه اول تناقض‌های عجیبی به نظر می‌آیند. ولی همه ما حداقل یک نفر را اطراف خودمان می‌شناسیم که بی‌نهایت آدم کله‌شقی است! هر چقدر هم که آدم‌ها تلاش کنند تا قانعش کنند، همیشه مرغش یک پا دارد. معمولاً سرنوشت این‌طور آدم‌ها چه می‌شود؟ تجربه ما نشان داده که بیشتر اوقات مردم آن‌ها را به عنوان موانعی در نظر می‌گیرند که باید دور زده شوند. بقیه به نظرات یا مخالفت‌هایشان خیلی محل نخواهند گذاشت. قاعدتاً نمی‌خواهید که شما هم به چنین آدمی تبدیل شوید. پس این موضوع را خوب در ذهن خودتان فرو کنید: هیچ اشکالی ندارد که یک نفر نظر شما را تغییر بدهد. موضوعاتی را که حاضرید به خاطر آن‌ها مبارزه کنید با دقت انتخاب کنید. یادتان باشد برای اینکه حرف شما به خوبی شنیده شود، ابتدا لازم است که شنونده خوبی برای حرف‌های بقیه باشید. قبل از اینکه نظر خودتان را بیان کنید یا در مورد مسئله‌ای با قطعیت تصمیم بگیرید، به نظرات بقیه با دقت گوش کنید. اگر شما دائماً تصمیم خودتان را عوض کنید، آدم بی‌اراده‌ای به نظر خواهید آمد.

اما در مورد موضوع آسیب‌پذیری، باید گفت که این مهارت در نگاه اول خیلی عجیب به نظر می‌رسد. اگر یک نفر به ضعف‌های خودش یا به عدم دانشش در یک موضوع خاص معترف باشد، مگر این نیست که بقیه او را کمتر جدی می‌گیرند؟ آسیب‌پذیری یک ضعف است و باعث خرابی دیوارهای اعتماد می‌شود؛ مگر نه؟

خیر، اصلاً درست نیست! پذیرش اشتباهاتتان یا اعتراف به اینکه کاری را به خوبی انجام ندادید، باعث پیشرفت شما در طولانی‌مدت می‌شود. واقعیت این است که تمام اصول HRT در این قضیه قرار دارند. پذیرش اشتباه حاصل تواضع و بیانگر مسئولیت‌پذیری شماست. همچنین نشانه‌ای است بیانگر اینکه شما می‌توانید به بقیه اعتماد کنید و در مقابل بقیه هم به شما، صداقت و قدرتان احترام بیشتری خواهند گذاشت.

گاهی اوقات بهترین کاری که می‌توانید انجام دهید فقط این است که بگویید: «نمی‌دونم.»



همه می‌دانند که سیاست‌مدارهای حرفه‌ای هیچ‌وقت به اشتباه خودشان یا به عدم اطلاع از یک موضوع خاص اعتراف نمی‌کنند؛ حتی وقتی که اشتباهشان برای همه کاملاً واضح و مبرهن باشد. برای همین است که هیچ‌کس یک کلمه از حرف‌هایشان را باور نمی‌کند. این رفتار سیاست‌مدارها شاید به این دلیل است که دائماً از طرف احزاب مخالفشان تحت حمله قرار می‌گیرند؛ ولی در دنیای مهندسی، هیچ دلیلی وجود ندارد که شما دائماً در حالت تدافعی قرار داشته باشید. هم‌تیمی شما، همکار شماست، نه رقیب شما.

قدم‌های بعد

اگر تا این بخش از کتاب با ما همراه بودید، در مسیر رسیدن به «هنر کار کردن با دیگران» قرار دارید. شما قطعاً باید با تمرکز و بررسی رفتارهای شخصی خودتان شروع کنید. وقتی مهارت‌هایی را که در این فصل بیان شد در زندگی روزمره خودتان استفاده کنید، متوجه خواهید شد که همکاری شما با هم‌تیمی‌هایتان طبیعی‌تر خواهد شد و کارایی تیمتان بالاتر خواهد رفت.

تغییرات مهم همیشه با خود شما شروع می‌شوند و به دیگران سرایت پیدا می‌کنند.

در فصل آینده، در مورد این موضوع صحبت خواهیم کرد که چطور می‌توان فرهنگ تیمی را بر اساس اصول HRT ارتقا داد.

فصل دوم

ساخت فرهنگ تیمی فوق‌العاده

فرهنگ تیم‌ها با هم تفاوت‌های بسیاری دارد. هر تیمی ارزش‌ها و هنجارهای مخصوص به خودش را تعریف می‌کند. بعضی از تیم‌ها در راستای موفقیت حرکت می‌کنند، در حالی که بعضی رو به شکست هستند. ولی حتی بین آن دسته از تیم‌ها که فرهنگ کار گروهی‌شان آن‌ها را در مسیر موفقیت قرار داده نیز تیم‌هایی پیدا می‌شوند که به شدت کارآمد هستند و تلاش بیشتر اعضا را در راستای رسیدن به هدف تیمی قرار می‌دهند. تیم‌هایی هم وجود دارند که فرهنگشان مزاحم رسیدنشان به هدف است. در این فصل ما درباره فرهنگ کار تیمی صحبت خواهیم کرد و به طور خاص روی مهارت‌های مناسب ارتباطی بین اعضای تیم تمرکز خواهیم کرد. ما نشان خواهیم داد که چطور استفاده از این تکنیک‌ها باعث بهبود کارایی تیم‌ها در تولید یک محصول خلاقانه خواهند شد.

فرهنگ تیمی چیست؟

وقتی کلمه «فرهنگ» را می‌شنوید، یاد چه چیزی می‌افتید؟ یک کنسرت موسیقی سنتی یا یک شب شعر مجلل با شاعران برجسته و فرهنگی؟ یا شاید به یک ظرف ژله فکر می‌کنید که در دوران دبیرستان و در آزمایشگاه سعی داشتید یک محیط کشت باکتری در آن درست کنید؟^۱ باید گفت که فرهنگ کار تیمی خیلی متفاوت با کشت باکتری در آزمایشگاه نیست.



۱. این مثال به این دلیل آورده شده که معادل انگلیسی کلمه فرهنگ یعنی culture، به معنی عمل «کشت باکتری در آزمایشگاه» نیز هست.

تا حالا یک نان خیلی خوشمزه خوردید؟ اگر دنبال کسی که این نان را پخته بگردید متوجه می‌شوید که رمز این خوشمزگی در مخمر این نان نهفته است و باکتری لاکتوباسیلوس^۱ که از آرد و آب تغذیه می‌کند. مخمر باعث می‌شود که نان به خوبی پف کند و آن باکتری باعث می‌شود که خوشمزه و لذیذ شود. ولی همه باکتری‌های لاکتوباسیلوس شبیه همدیگر نیستند. بعضی از باکتری‌ها باعث می‌شوند که نان خوشمزه‌تر شود، در حالی که بعضی دیگر از باکتری‌ها خیلی مزه مطلوبی به نان نمی‌دهند. وقتی که یک نفر یک ترکیب طلایی از مخمر و باکتری برای پخت نان پیدا می‌کند، سعی می‌کند که این ترکیب را با تغذیه از آب و آرد حفظ کرده و افزایش دهد. سپس برای پخت هر نان، یک تکه کوچک از این ترکیب را به عنوان مایه اولیه جدا کرده و یک نان خوشمزه درست می‌کند. علت موفقیت این استراتژی این است که ترکیب مناسب مخمر و باکتری، نه تنها مزه مطلوب را ایجاد می‌کند، بلکه اثر سایر باکتری‌ها و مخمرهای مزاحم موجود در نان یا هوا و محیط اطراف آشپزخانه را نیز خنثی می‌کند.



فرهنگ تیمی شما، مثل یک تکه نان خوشمزه است. محیط اولیه (بنیان‌گذاران تیم) شرایط را برای کشت مخمر و باکتری‌ها (اعضای تیم) فراهم می‌کنند تا یک نان خوشمزه (تیم شما) تولید شود. اگر مخمر و باکتری‌های اولیه قوی باشند، به راحتی می‌توانند از پس مزاحمت‌هایی که باکتری‌های جدید می‌توانند ایجاد کنند برآیند^۲. ولی اگر محیط اولیه ضعیف باشد، نان شما نسبت به نفوذ ناخالصی و یا باکتری‌های جدید و مخرب کاملاً آسیب‌پذیر

1. lactobacillus

۲. البته محیط قوی مخمر و باکتری، امکان این انتخاب را هم دارد که فواید و خصوصیات ناب و مفید یک باکتری جدید را هم جذب کند.

خواهد بود. باکتری‌های ناآشنا با خود نتایج غیرقابل پیش‌بینی به همراه می‌آورند. در نتیجه بهتر است همیشه محیط اولیه را قوی نگه دارید.

فرهنگ تیمی صرفاً راه و روشی که اعضای تیم وظایفشان را انجام می‌دهند، برنامه‌نویسی می‌کنند یا شیوه‌ای که با یکدیگر رفتار می‌کنند نیست؛ بلکه مجموعه‌ای از تجارب مشترک، ارزش‌ها و اهدافی است که برای هر تیم به شکل منحصربه‌فرد مشخص می‌شوند. بنیان‌گذاران یک تیم یا یک شرکت، نقش بسیار پررنگی در تعریف فرهنگ تیم دارند، ولی این فرهنگ تیمی در طول زمان زندگی یک تیم تغییر و بسط پیدا می‌کند.

عوامل متفاوتی در ماهیت فرهنگ یک تیم وجود دارند. بعضی از این عوامل مستقیماً به کار مربوط هستند. مثلاً به طور خاص در مورد برنامه‌نویسی، ممکن است فرهنگ تیمی عادات‌هایی مثل مرور کد، یا کدنویسی بر اساس نوشتن تست و یا به خوبی مکتوب کردن برنامه‌ها را تشویق کند. اما بعضی دیگر از عناصر یک فرهنگ تیمی ممکن است بیشتر وجه اجتماعی داشته باشند. مثلاً برنامه‌ناهار تیمی هر پنجشنبه در یک رستوران خاص، یا دوره‌می تیمی در عصرهای پنجشنبه. بعضی از این عادات ممکن است برای یک آدم خارج از گروه خیلی عجیب و غریب یا حتی احمقانه به نظر برسند. شعبه‌دفتر گوگل در شهر پیتزبرگ نزدیک به ریل قطارهای باری قرار داشت. عادت همه این بود که هر وقت یک قطار با صدای بلند از کنار دفتر عبور می‌کرد، همه بلند می‌شدند و با تفنگ‌های اسباب‌بازی به همدیگر گلوله‌های لاستیکی شلیک می‌کردند.^۱ تمام این‌ها فرهنگ یک تیم را تشکیل می‌دهند و روی کارایی تیمی و توانایی جذب و حفظ نیروهای خوب، تأثیر می‌گذارند.

امروزه به هر شرکت بزرگ و موفق (مثل گوگل، اپل، مایکروسافت، اوراکل و ...) که نگاه کنید متوجه می‌شوید که هر یک فرهنگ منحصربه‌فرد خود را دارند. ریشه فرهنگ کاری هر شرکت را می‌توان در بنیان‌گذاران و نخستین کارمندان شرکت پیدا کرد. به مرور با بزرگ شدن و گسترش این شرکت‌ها، فرهنگ آن‌ها نیز به تدریج و با ورود اعضای جدید تغییر کرد، ولی شخصیت وجودی این شرکت‌ها هنوز منحصربه‌فرد و متفاوت است. شخصیت اصلی شرکت‌ها تعیین‌کننده اصلی این موضوع است که یک محصول چطور طراحی و ساخته شود یا با کارمندان چطور رفتار شود و یا با سایر شرکت‌ها چطور رقابت صورت گیرد.

۱. وقتی فیتز اولین بار از این دفتر بازدید کرد و با این حرکت کارکنان مواجه شد، به شدت وحشت کرد.

چرا باید به فرهنگ تیمی اهمیت داد؟

جواب کوتاه این است که اگر به فرهنگ تیمتان توجه نکنید، خیلی زود انسان‌هایی که صرفاً اعتماد به نفس بیشتر یا نظر و صدای بلندتری دارند، فعالیت‌های تیم شما را تحت‌الشعاع قرار می‌دهند. البته ممکن است نتیجه خیلی هم بد نشود و به طور اتفاقی تیم شما با یک فرهنگ سالم، خیلی کارآمد به وظایفش ادامه دهد. ولی اکثر اوقات خوش‌شانس نخواهید بود و متوجه می‌شوید که انرژی زیادی که تیم می‌توانست صرف طراحی و پیاده‌سازی نرم‌افزار کند، برای بحث و جدل‌های الکی هدر داده است. از سوی دیگر، باید برای داشتن فرهنگ تیمی‌ای تلاش کرد که برای تمام اعضای تیم ارزشمند باشد و همگی حاضر باشند از آن دفاع کنند. اگر ارزش‌های تیمی شما مورد قبول همه نباشد، نمی‌توان یک شخصیت منحصربه‌فرد برای تیم تعریف کرد. گذشته از این، اعضای جدیدی که به تیم اضافه می‌شوند خیلی راحت می‌توانند فرهنگ تیمی را خراب کنند.

اولین اشتباه بیشتر تیم‌ها این است که فرض می‌کنند این رهبران تیم‌ها هستند که فرهنگ تیم را تعریف می‌کنند. درست است که بستر اولیه ارزش‌های تیمی را بنیان‌گذاران تیم مشخص کرده‌اند و رهبران تیم برای تقویت فرهنگ تیمی تلاش می‌کنند، ولی همه اعضای تیم در تعریف، مراقبت و دفاع از این فرهنگ مسئول هستند. وقتی عضو جدیدی وارد تیم می‌شود، خلق و خوی تیم را نه از رهبر تیم بلکه از تک‌تک اعضای گروه یاد می‌گیرد. به عنوان مثال، وقتی شما کار عضو جدید تیم را با او مرور کرده و توضیح می‌دهید که تیم شما چطور کار می‌کند، در حقیقت او را با ارزش‌های تیمی خودتان آشنا می‌کنید. اعضای جدید، به نحوه کار کردن و یا بحث کردن سایر افراد تیم با همدیگر نیز دقت می‌کنند و با خلق و خوی تیم آشنا می‌شوند.

فرهنگ تیمی «قوی» فرهنگی است که نسبت به تغییرات مثبت قابل انعطاف، ولی در مقابل تغییرات رادیکال و منفی مقاوم باشد. موفق‌ترین تیم‌ها آن‌هایی هستند که بیشتر سعی و کوشش اعضای آن‌ها روی طراحی و تولید یک نرم‌افزار خوب متمرکز شده است. اگر تمرکز اصلی و اولیه تیم شما روی هر چیز دیگری (مثل مهمانی گرفتن، جلسه‌های طولانی، رقابت و سبقت گرفتن از همدیگر) باشد، اگرچه ممکن است که دوستی و رفاقت بین اعضای تیم تقویت شود، ولی نهایتاً وقت و انرژی زیادی برای توسعه نرم‌افزار پیدا نخواهید کرد. اگر شما با کدنویسی، توسعه و تولید نرم‌افزار به رضایت شغلی می‌رسید، کاملاً به نفع شماست که یک تیم با یک فرهنگ قوی پیدا کنید و تلاش کنید که این فرهنگ حفظ شود. ممکن است تیم شما بدون یک فرهنگ قوی و کارآمد، موفق شود که یک نرم‌افزار خوب تولید کند، ولی قطعاً این کار برای چنین تیمی هزینه‌های مادی، زمانی و صرف انرژی بسیار بیشتری خواهد داشت. فرهنگ قوی در تیم بی‌شک عامل بهره‌وری بیشتر، توانایی و شادی بالاتر اعضا خواهد بود.

نکته جالب اینجاست که یک فرهنگ خوب تیمی، خود عامل جذب بهترین نیروها خواهد بود. در دنیای نرم‌افزارهای متن‌باز، اگر شما پروژه خود را بر پایه اصول HRT جلو ببرید و تمرکز تلاش خود و اعضای تیم را حول محور کدنویسی منظم و خوب قرار دهید، پروژه شما افرادی را به خود جذب خواهد کرد که برای تواضع، احترام

و اعتماد در ارتباطات تیمی ارزش قائل هستند. ولی در مقابل اگر خلق و خوی تیمی شما، پرخاشگرانه و پر از جروب‌های بیخود باشد، افراد مشابهی را جذب خواهد کرد.

این قضیه را بارها در بنیاد نرم‌افزار آپاچی تجربه کردیم. بنیاد، مجموعه‌ای از تیم‌های توسعه نرم‌افزار است که بر اساس مدل و اصول مشترک و توافقی کار می‌کنند. وقتی یک عضو جدید وارد این انجمن می‌شود و به دلیل عدم اطلاع یا شاید سوء قصد طوری رفتار می‌کند که مطابق اصول و ارزش‌های گروه نیست، سایرین سعی می‌کنند که به او (گاهی به آرامی و بعضی اوقات ... هم... نه چندان آرام) آموزش بدهند. اگر عضو جدید از خلق و خوی تیمی انجمن خیلی خوشش نیاید، معمولاً از گروه خارج شده و به دنبال پروژه‌های دیگری می‌گردد که با فرهنگ او منطبق‌تر باشند.

در شرکت‌های نرم‌افزاری، انتخاب عضو جدید معمولاً از طریق مصاحبه و استخدام انجام می‌شود. ارزیابی تطابق فرهنگی یا تلویحاً در طول سنجش مهارت‌ها انجام می‌شود یا صریحاً در یک مرحله مشخص در طول مصاحبه مورد بررسی قرار خواهد گرفت. شرکت گوگل روش شفاف و صریح را انتخاب کرده و یک مرحله مشخص از کل فرایند مصاحبه استخدامی را به ارزشیابی و بررسی این موضوع اختصاص می‌دهد که آیا کاندید مورد نظر از لحاظ خلق و خو با تیم سازگار است یا خیر. هر اندازه که یک داوطلب جایگاه شغلی از لحاظ فنی و مهندسی، شخص باهوش و با استعدادی باشد، اگر در توانایی‌های کار تیمی ضعیف باشد، مصاحبه‌کننده باز خورد منفی‌ای خواهد داشت.

اگر به تطابق فرهنگی رفتار افراد در حین مراحل استخدام توجه نکنید، نهایتاً مجبور می‌شوید که وقت و انرژی زیادی را صرف این موضوع کنید که یا عضو جدید را به نحوی با تیم تطبیق دهید، یا او را از تیم خارج کنید. صرف نظر از نتیجه، هزینه این کار به قدری زیاد خواهد بود که شما را وادار کند قبل از استخدام یک فرد، او را به خوبی از لحاظ تطابق رفتاری با تیم ارزیابی کنید و مطمئن شوید که با تیم شما به خوبی کار خواهد کرد.

ارزیابی تطابق فرهنگی

تنها روشی که می‌توان به وسیله آن مطمئن شد اعضای جدیدی که به تیم اضافه می‌شوند با تیم به خوبی کار خواهند کرد، این است که این امر در مراحل مصاحبه استخدامی مورد ارزیابی قرار گیرد. خیلی از شرکت‌های بزرگ (مثل گوگل) اصولاً یکی از معیارهای ارزیابی استخدامی‌شان، تطابق فرهنگی است. بعضی دیگر از شرکت‌ها این قضیه را حتی یک مرحله جدی‌تر در نظر می‌گیرند و بررسی خلق و خو و تطابق فرهنگی را حتی قبل از ارزشیابی مهارت‌های فنی و مهندسی انجام می‌دهند. آن‌ها افرادی را که از نظر فرهنگی با تیمشان منطبق نیستند، حتی برای ارزیابی فنی لحاظ نخواهند کرد. این پروسه دقیق، برای ساخت یک تیم قوی بسیار حیاتی است. تیم‌های فوق‌العاده، اتفاقی به وجود نمی‌آیند، بلکه برعکس با وسواس و دقت بنیان‌گذاران و کارمندان اولیه در حفظ فرهنگ تیمی شکل می‌گیرند.

بعد انسانی فرهنگ تیمی

فعالیت‌های خلاقانه مثل برنامه‌نویسی، با کارهای یدی مثل قرار دادن یک سری قطعه روی نوار تولید کارخانه متفاوت است. برای انجام دادن بعضی از کارها، چند روز آموزش و یک سری ابزارآلات ساده کافی هستند. وقتی کارگر شما تصمیم می‌گیرد که کار را رها کند، یا کلاً خیلی خوب کار نکند، شما به راحتی می‌توانید شخص جدیدی را جایگزین کنید تا کار ادامه پیدا کند. در کارهای یدی، کارگران فعالیت‌های به نسبت ساده‌ای را با پیروی از دستورالعمل‌های مشخص انجام می‌دهند و کارشان خیلی نیازی به تفکر خلاقانه یا قدرت حل مسئله نخواهد داشت. ولی در دنیای نرم‌افزار، مهندسی به توانایی بسیار زیادی در راستای تفکر خلاقانه نیاز دارند.^۱ اگر شما به دنبال تولید یک نرم‌افزار فوق‌العاده هستید، به مهندسی خارق‌العاده نیز احتیاج دارید و اگر می‌خواهید که مهندسی خارق‌العاده شما، به نحو احسن کار کنند و در تیم شما باقی بمانند، باید فرهنگ تیمی‌ای خلق کنید که به مهندسی اجازه دهد تا با خیال راحت نظراتشان را مطرح کنند و در تصمیم‌گیری‌های اساسی نقش داشته باشند.

برای داشتن مهندسی عالی در تیمتان، طبیعتاً ابتدا باید با استخدام تعدادی مهندس عالی شروع کنید. درست است که این جمله شاید عجیب به نظر برسد، ولی واقعیت امر این است که بیشتر مهندسی خوب، دوست دارند که با مهندسی عالی دیگر هم‌تیمی باشند. بیشتر مهندسی خوبی که ما می‌شناسیم به سمت تیم‌هایی جذب

۱. بعضی‌ها فکر می‌کنند که با استخدام یک طراح نرم‌افزار خارق‌العاده و یک سری برنامه‌نویس معمولی، می‌توانند نرم‌افزار خوبی تولید کنند. ما مخالف این نظر نیستیم، ولی کار کردن با یک تیم فوق‌العاده و الهام‌بخش که دائماً شما را در جهت پیشرفت به چالش کشانده و به شما چیزهای جدید یاد بدهند، به مراتب هیجان‌انگیزتر خواهد بود.

می‌شوند که در آنها بتوانند از بزرگان صنعت برنامه‌نویسی چیزهای جدید یاد بگیرند.^۱ ولی سؤال مهم اینجاست که در مرحله اول چطور می‌شود چنین بزرگانی را جذب تیم کرد؟

برای شروع باید گفت که بزرگان دنیای برنامه‌نویسی، نه تنها دوست دارند که در تولید نرم‌افزار و برنامه‌نویسی شرکت کنند، بلکه دوست دارند که در طول مراحل طراحی و تصمیم‌گیری محصول نیز نقش داشته باشند. این یعنی مدیریت شما باید تا حدودی «اجماع محور» و با توجه به رأی و نظر کلی گروه باشد. در حالت مدیریت بالا به پایین، مهندس برتر (آلفا)، رهبر فنی تیم خواهد بود و سایر مهندسين به عنوان اعضای تیم استخدام شده‌اند. دلیل این امر این است که سایر اعضای تیم هزینه کمتری دارند و به راحتی می‌توان بر آنها ریاست کرد. نتیجه این خواهد شد که شما برای استخدام برنامه‌نویسان درجه اول با مشکل مواجه می‌شوید. مهندسین خارق‌العاده چرا باید مسافر اتوبوس شما باشند، در حالی که می‌توانند در شرکت‌های دیگر راننده اتوبوس باشند؟ در حالت مدیریت اجماع محور اما، تمام اعضای تیم در مراحل تصمیم‌گیری نقش دارند.

خیلی از آدم‌ها وقتی اسم «تیم اجماع محور» را می‌شنوند، به یک عده آدم ژولیده و آشفته فکر می‌کنند که در بیابان دور یک آتش در حال خواندن «کومبایا کومبایا» هستند و می‌رقصند و هیچ تصمیمی نمی‌گیرند و کاری را به آخر نمی‌رسانند؛ ولی این کلیشه بیشتر در مورد یک تیم ناکارآمد صدق می‌کند تا یک تیم اجماع محور. منظور ما از کلمه «اجماع» در اینجا این است که تمام اعضای تیم نسبت به کار، حس مالکیت داشته و در قبال موفقیت محصول نهایی احساس مسئولیت می‌کنند. و رهبران تیم از ته دل و با احترام به نظرات اعضای تیم گوش می‌کنند - با یک تأکید خاص روی ستون «احترام» از اصول HRT - این به آن معناست که گاهی اوقات بحث و تبادل نظر طولانی برای موفقیت محصول نرم‌افزاری مورد نیاز است و بعضی اوقات تیم به این نتیجه می‌رسد که باید سریع‌تر جلو بروند. در حالت تصمیم‌گیری‌های سریع، ممکن است تیم به این نتیجه برسد که بهتر است به تصمیم رهبران فنی تیم اعتماد کند.^۲ برای اینکه این امر به وقوع بپیوندد، همه اعضای تیم باید روی اهداف کلی تیم هم عقیده باشند. چه باور کنید چه باور نکنید، کلید رسیدن به این حالت در نوشتن «شرح مأموریت» تیم نهفته است که در ادامه این فصل به آن می‌پردازیم.

نحوه رفتار و برخورد اعضای تیم با همدیگر نیز به اندازه نحوه تصمیم‌گیری در تیم شما اهمیت دارد. اگر فرهنگ تیمی شما طوری باشد که افراد با هم دست به یقه می‌شوند یا بر سر هم جیغ و فریاد می‌کشند، تیم شما، اعضای

۱. مهندسین خیلی خوب، به دنبال رهبران خیلی خوب هم هستند. چرا که رهبران بد، نه تنها اعتماد به نفس کافی برای سروکله زدن با مهندسین حرفه‌ای را ندارند، بلکه عمدتاً در مورد کارمندانمان مانند ارباب‌ها رفتار می‌کنند.

۲. وقتی یک اجماع کلی به دست نمی‌آید، بعضی از تیم‌ها تصمیم‌گیری را به رهبران فنی خود واگذار می‌کنند، در حالی که بعضی دیگر بین اعضای تیم رأی‌گیری می‌کنند. مهم نیست تیم شما چه روشی را برای حل اختلاف انتخاب می‌کند، مهم این است که همیشه برای حل این نوع اختلافات یک روش را تکرار و دنبال کنید.

خشن و پرخاشگر را جذب خواهد کرد و فضایی پر از خودپرستی و غرور نابه‌جا تولید می‌کند (و حقیقتاً به طور خاص، بیشتر خانم‌هایی که ما می‌شناسیم از این‌گونه تیم‌ها دوری می‌کنند). اگر فرهنگ تیمتان را بر پایه ستون‌های HRT استوار کنید تا اعضای تیم با یکدیگر با احترام رفتار کرده و برای انتقاد سازنده از همدیگر تلاش کنند، نه تنها افراد بیشتر و بهتری را جذب می‌کنید، بلکه میزان بالاتری از وقت و انرژی خود را روی تولید نرم‌افزار متمرکز می‌کنید. داشتن غرور و افتخار تیمی اگرچه مفید است، ولی تیمی که در غرورهای فردی اعضا محو شود، در مسیر شکست قرار می‌گیرد. در فصل چهارم درباره راه‌های جلوگیری از این اتفاق صحبت خواهیم کرد.

انتقاد سازنده، از ضرورت‌های رشد و پیشرفت هر شخص و یا هر تیمی است. ولی بیشتر انسان‌ها تمام تلاش خود را می‌کنند تا از دریافت هرگونه انتقادی فرار کنند. علت این فرار در بعضی مواقع شاید عدم اعتمادبه‌نفس باشد، ولی در اکثر اوقات علت امر این است که آدم‌ها فکر می‌کنند باید در قبال هر نظر یا انتقادی کاری انجام بدهند و تغییر کنند، حتی اگر آن انتقاد خاص را قبول نداشته باشند. بهترین خصوصیت دریافت انتقاد سازنده این است که شما این توانایی را دارید که انتخاب کنید کدام قسمت از یک انتقاد را قبول دارید و در مقابل کدام قسمت نمی‌خواهید کار خاصی انجام دهید. مثلاً فرض کنید که شما می‌خواهید برای یک مصاحبه شغلی آماده شوید و لباس مورد علاقه خودتان را پوشیده‌اید. پیش یک دوست قابل‌اعتماد می‌روید و نظرش را درباره ظاهر خود می‌پرسید. اگر او بگوید: «گوشه دندان شما یک تکه اسفنج چسبیده و من از لباست خوشم نمی‌آد»، شما به راحتی می‌توانید از نخ دندان استفاده کنید، ولی لزومی ندارد که لباستان را عوض کنید. انتقاد سازنده یک هدیه است که شما می‌توانید آن را قبول یا رد کنید.

اگر به دنبال بهبود نحوه کارتان یا حل کردن ایرادهای شخصی خود هستید، همین دوستان و همکاران هستند که می‌توانند به شما کمک کنند تا موانع پیشرفت را به شما نشان دهند. بدون نظرخواهی از بقیه، شما اشتباهات و ایرادهای خودتان را دائماً تکرار خواهید کرد. مگر اینکه یک قدرت خارق‌العاده از خودآگاهی و خویش‌شناسی داشته باشید. به عنوان مثال، پیش از انتشار نهایی این کتاب در رسانه‌ها، ما از چندین نفر خواستیم که نظرشان را در مورد نوشته ما بگویند. بیشتر این نظرات و انتقادهای بسیار دقیق و باارزش بود. صرف‌نظر از اینکه نظر شما در مورد این کتاب چیست، این کتاب شدیداً بدتر می‌بود اگر که ما به پیشنهادهای ارزشمند دیگران اهمیت نمی‌دادیم یا می‌ترسیدیم که نظر بقیه را بپرسیم.

درخواست انتقاد و پیشنهاد، نیازمند داشتن یک سطح مشخص از اعتمادبه‌نفس است. ما فکر می‌کنیم انتقادهای سازنده راحت‌ترین نظری است که می‌شود دریافت کرد. از طرف دیگر، دادن انتقاد سازنده به دیگران کار بسیار سختی است. معمولاً کار راحت‌تر این است که به راحتی کارشان را زیر سؤال ببریم یا مسخره کنیم. البته خود عمل درخواست کردن از دیگران برای نظرشان هم کار آسانی نیست. مردم فرض می‌کنند شما صرفاً به دنبال تعریف و تمجید از کارتان هستید. اگر دوست یا همکاری را پیدا کردید که وقتی از او نظرش را می‌پرسید با شما یک نظر صادقانه و انتقاد سازنده را در میان می‌گذارد، دو دستی به او بچسبید و از دستش ندهید.

آدم‌های تند و پرخاشگر معمولاً در یک تیم ساکت و آرام، خیلی احساس ناراحتی نمی‌کنند. ولی انسان‌های آرام و درون‌گرا عموماً در یک محیط پر حرارت نمی‌توانند موفق شوند یا از کار لذت ببرند. نه تنها شنیدن نظر آن‌ها در چنین محیطی سخت‌تر است، بلکه خود آن‌ها نیز از بیان نظرات و ایده‌های خود دل‌سرد می‌شوند.^۱ اگر دنبال داشتن تیمی هستید که در آن بیشتر افراد احساس راحتی کنند و با کارایی بیشتری فعالیت داشته باشند، باید فرهنگ تیمتان را بر اساس تواضع، احترام و اعتماد قرار دهید.

فرهنگ تیم‌های خونسرد و آرام در مقابل انسان‌های پرخاشگر آسیب بیشتری می‌بینند تا فرهنگ‌های پر حرارت در مقابل انسان‌های آرام و خوش‌برخورد. تیم‌های ملایم و با آرامش باید حواسشان به نفوذ انسان‌های پرخاشگر باشد. بهترین کار این است که با این‌طور انسان‌ها درگیر نشوند. بعضی اوقات این وظیفه اعضای ارشد تیم است که روبه‌روی انسان سلطه‌جو بایستند تا از آسیب او به سایر افراد خوش‌برخورد تیم جلوگیری کنند. در فصل چهارم درباره نحوه برخورد با این‌طور انسان‌های «سمی» صحبت خواهیم کرد.

۱. سوزان کین یک TED Talk: The Power of Introverts عالی است و همچنین یک کتاب در مورد همین موضوع دارد: The Power

الگوهای ارتباطی در فرهنگ‌های موفق

ارتباطات اجتماعی ممکن است برای اعضای تیم به یک چالش اساسی تبدیل شوند. مخصوصاً برای مهندسين نرم‌افزار که معمولاً دوست دارند بعدازظهر خود را در کنار یک کامپایلر منطقی و قابل پیش‌بینی سپری کنند تا در کنار یک انسان احساسی و غیرقابل پیش‌بینی. بیشتر اوقات برنامه‌نویسان، نیاز به ارتباط با یکدیگر را به عنوان یک مانع و مزاحم برای کدنویسی تلقی می‌کنند. ولی اگر اعضای یک تیم از کار هم بی‌خبر باشند یا در مورد یک موضوع توافق نداشته باشند، هیچ‌گاه مشخص نمی‌شود که آیا کد و برنامه در جهت درستی در حال توسعه است یا خیر.



اگر هر تیم موفق و کارآمدی را بررسی کنید، متوجه می‌شوید که روی کانال‌های ارتباطی مؤثر به خوبی سرمایه‌گذاری کرده‌اند. به طور مثال گروه‌های ایمیل، مستندهای طراحی نرم‌افزار، شرح مأموریت، کامنت‌های مناسب در کدنویسی، خودآموزهای تولید نرم‌افزار و ... اطمینان از اینکه همه اعضای تیم روی مسیر حرکت و اهداف تیم هم‌عقیده هستند و از آخرین تصمیم‌ها باخبر باشند، کار سخت و پُرزحمتی است. ولی تمام این زحمات حکم یک سرمایه‌گذاری را دارد که نهایتاً با افزایش کارایی و رضایت تیمی نتیجه خواهد داد.

یک قاعده و قانون کلی در ارتباطات تیمی این است که تا جایی که می‌شود تعداد افراد کمتری را در تماس‌های هم‌زمان و هم‌گام مثل جلسه‌های حضوری یا تماس‌های تلفنی دخیل کنید و در مقابل تعداد افراد بیشتری را حین ارتباطات غیر هم‌زمان و ناهم‌گام مثل ایمیل یا کامنت‌ها یا نرم‌افزارهای مدیریت کار در جریان بگذارید. ارتباطات هم‌گام هزینه بالایی دارند، چرا که شرکت‌کنندگان را مجبور می‌کنند تا روند کارشان را قطع کرده و خودشان را با برنامه زمانی شما تطبیق بدهند. اما ارتباطات غیر هم‌گام این امکان را به افراد می‌دهند تا در زمان و مکان مناسب به آن‌ها توجه شود. هر زمانی که باعث وقفه در کار افراد شوید، آن‌ها مجبور خواهند شد مدت زمانی را صرف این کنند تا دوباره به همان مرحله از سرعت و دقت برسند. همیشه به این موضوع توجه کنید.

ولی مهم‌تر از هر چیزی این است که شما باید کاری کنید تا همیشه تمام اطلاعات، مستند شده و در دسترس تمام افراد تیم قرار بگیرد. تعدادی از روش‌های متداول ارتباطات تیمی را با هم در این فصل مرور می‌کنیم. بعضی از این روش‌ها ممکن است خیلی واضح و مبرهن باشند، ولی نکات ظریفی دارند که بهتر است بررسی شوند. در مورد یک چیز اما خاطر جمع باشید: اگر در راستای بهبود ارتباطات تیمی خود تلاش نکنید، زمان قابل ملاحظه‌ای را صرف دوباره کاری یا کارهای بیهوده در تیمتان خواهید کرد.

همگام‌سازی اعضای تیم

به طور کلی، هدف از همگام‌سازی اعضای تیم این است که اولاً تمام افراد روی اهداف مشترک تیم توافق داشته باشند و ثانیاً روش‌های ارتباطی بین افراد، مورد تأیید همه اعضای تیم باشد.

شرح مأموریت - نه، واقعاً!

وقتی کلمه «شرح مأموریت» را می‌شنوید، احتمالاً اولین چیزی که به ذهنتان می‌رسد آگهی‌های بازاریابی بی‌روح و بی‌مزه بعضی از شرکت‌های بزرگ است. مثلاً در زیر یکی از این شرح مأموریت‌ها را که متعلق به یک شرکت بزرگ مخابراتی است و بهتر است اسمش را مطرح نکنیم، می‌بینید:

مقصد ما تبدیل شدن به تحسین‌آمیزترین و باارزش‌ترین شرکت دنیا است. هدف ما غنی‌سازی زندگی شخصی مشتریانمان، موفق‌تر کردن زندگی کاری آن‌ها با عرضه هیجان‌انگیزترین و پرکاربردترین سرویس‌های مخابراتی است، در حالی که هم‌زمان برای سهام‌داران شرکت نیز سودآوری کنیم.

جالب اینکه من هنوز هیچ‌کس را پیدا نکردم که این شرکت خاص را تحسین کرده باشد!

یک مثال دیگر از یک شرکت بزرگ دیگر:

عرضه کردن راه‌حل‌های کاربردی در کمترین زمان در جهت پاسخگویی به نیاز مشتریان. اصلاً این یعنی چه؟ عملاً جمله بالا برای هر کاری و هر شرکتی می‌تواند معنی داشته باشد. اگر ما در شرکت بالا کار می‌کردیم، احتمالاً نمی‌توانستیم تشخیص دهیم که بین شستن ماشین، درست کردن درز لوله‌ها یا تحویل دادن پیتزا، کدام کار اهمیت بیشتری برای شرکت دارد. این چنین مغلطه‌هایی از طرف شرکت‌های عظیم هستند که واژه «شرح مأموریت» را بدنام کرده است.

نوشتن شرح مأموریت تیمی، بهترین راه برای تیم‌های با کیفیت و کارآمد است تا بتوانند محدوده کاری خود را مشخص کرده و مسیر خودشان را اجمالاً تعریف کنند. درست است که نوشتن این شرح مأموریت ممکن است زمان‌بر و طاقت‌فرسا باشد، ولی می‌تواند با تعریف آنچه که تیم شما باید یا نباید انجام دهد، به صورت بالقوه چیزی به اندازه سال‌ها کار را در وقت و زمان شما صرفه‌جویی کند.

وقتی گوگل تصمیم گرفت که توسعه^۲ یا به اختصار GWT را به شکل یک پروژه متن‌باز ادامه بدهد، ما به عنوان مشاور در کار دخیل شدیم. ما تفاوت‌های یک پروژه متن‌باز با غیر متن‌باز را با دقت مرور کردیم و به طور خاص

۱. «نباید»‌ها بی‌اندازه اهمیت دارند. «نه» گفتن به آنچه که حواس تیم شما را از مسیر اصلی پرت کند، از عوامل مهم موفقیت تیمی است.

به این خصوصیت از دنیای توسعه نرم افزارهای متن باز توجه کردیم که در آن هر کسی از هر جایی می‌تواند در پروژه وارد شود، نظرش را مطرح کند، در بخشی از کد مشارکت داشته باشد یا طرح و برنامه‌ای را زیر سؤال ببرد.^۱ بعد از بررسی تمام این چالش‌ها، نهایتاً به تیم پیشنهاد دادیم تا یک شرح مأموریت تیمی شامل تمام اهداف پروژه و همین طور لیست مواردی که هدف این پروژه نیست تهیه کنند و در معرض عموم شرکت کنندگان در پروژه قرار دهند.

بعضی از اعضای تیم این پیشنهاد را رد کردند، در حالی که بعضی دیگر به نظر کنجکاو شدند. خوشبختانه رهبر تیم از این پیشنهاد خیلی استقبال کرد. وقتی دور هم نشستیم تا این شرح مأموریت را تعریف کنیم، بحث‌های زیادی در مورد محتوا و نحوه نگارش این نوشته شکل گرفت. نهایتاً بعد از چند روز مذاکره نه تنها یک شرح مأموریت دقیق و عالی تنظیم شد، بلکه تیم موفق شد که یک سند دیگر در مورد اینکه «چطور GWT را بهتر کنیم» تهیه کند. آن‌ها حتی یک قسمت از شرح مأموریت را به طور خاص به مواردی که هدف پروژه نیست، اختصاص دادند. در زیر این شرح مأموریت آورده شده است:

مأموریت GWT این است که تجربه کاربران در دنیای وب را به طور بنیادی ارتقا دهد. این کار از طریق فراهم کردن ابزارهای مبتنی بر Java برای توسعه‌دهندگان وب انجام می‌شود تا آن‌ها بتوانند به راحتی برای هر مرورگر مدرنی، امکانات AJAX فراهم کنند.

جمله بالا محتوای زیادی را در خود جمع کرده است و ما معتقدیم که این یک مثال خیلی خوب از یک شرح مأموریت ایدئال است. این شرح مأموریت هم هدف را تعریف می‌کند: ارتقای تجربه کاربران در دنیای وب و هم محدوده فعالیت را مشخص می‌کند: ابزارهای مبتنی بر Java.

چندین سال بعد وقتی با مدیر این تیم شام می‌خوردیم، فیتز از حمایت‌های او در مورد نوشتن این شرح مأموریت بسیار تشکر کرد؛ ولی مدیر آن تیم اعتراف کرد که وقتی اولین بار این ایده را شنید، احساس کرد که این کار یک اتلاف وقت اساسی خواهد بود، ولی وقتی که با مهندسين تیمش شروع به بحث و مناظره در مورد محتوای این شرح مأموریت کرد، متوجه اهمیت این کار شد، چرا که فهمید مهندسين ارشد تیم کلاً و اصولاً با اهداف پروژه موافق نیستند!

نوشتن شرح مأموریت به تیم GWT کمک کرد تا همان ابتدا با اختلاف نظرات بین اعضای تیم مواجه شده و سعی کنند در مورد خط‌مشی کلی محصول نهایی به توافق برسند. مسئله‌ای که می‌توانست بعدها روی سرعت یا حتی موفقیت پروژه تأثیر بگذارد. آن‌ها شرح مأموریت تیمی خودشان را در معرض دید همه روی وب قرار دادند

۱. ما معمولاً روند توسعه نرم افزار را به ساختن یک خانه کاغذی روی یک تشک کشسان حرکات آکروبا تیک تشبیه کرده‌ایم. این کار نیاز به صبر بالا، دقت کافی و همین طور حوصله برای سروکله زدن با افراد بی‌دقتی دارد که بی‌محابا روی تشک می‌پرند. (م.)

و نه تنها تمام اعضای تیم با دقت از آن تبعیت کردند، بلکه زمان زیادی که می‌توانست صرف جروب‌بحث با افراد جدید در مورد اهداف و مسیر پروژه بشود، حفظ شد. فقط کافی بود که تازه‌واردها را با شرح مأموریت تیم آشنا کنند تا جواب بسیاری از سؤالاتشان را بگیرند.



شرح مأموریت، تضمین می‌کند که پروژه شما از مسیر خود خارج نشود. این سند ولی نباید یک مانع سنگین جلوی تغییرات در کار شما ایجاد کند. اگر محیط کاری یا طرح و برنامه شرکت یا پروژه شما به شکل اساسی‌ای تغییر کرد، اعضای تیم باید با خودشان صادق باشند و شرح مأموریت را مجدداً ارزیابی کنند و محتوای آن را به‌روز کنند. ایجاد تغییر در قانون اساسی تعمداً کار سختی است تا افراد هر موقع که دلشان خواست نتوانند آن را تغییر دهند. ولی در یک سری شرایط خاص، تغییرات در قانون اساسی امکان‌پذیر هستند و می‌توانند مورد بررسی قرار بگیرند. اگر سمت و سوی یک محصول یا یک شرکت به ناگهان و به هر دلیلی تغییر کند، شرح مأموریت تیم نیز باید متناسب با آن به‌روز شود.

جلسه‌های کارآمد

بیشتر آدم‌ها به «جلسه‌ها» به عنوان یک نیاز ولی یک بلا برای گریزناپذیر نگاه می‌کنند. با وجود آنکه می‌توان با کمی مهارت این جلسه‌ها را بسیار مؤثر و کارآمد برگزار کرد، اکثر اوقات این اتفاق نمی‌افتد و جلسات، بیش از اندازه طولانی و نامنظم انجام می‌شوند. ما جلسه‌ها را همان طور دوست داریم که تصفیه‌خانه‌های فاضلاب شهری را دوست داریم: دورادور، کم‌تعداد و هم‌سو با مسیر باد!

با مخوف‌ترین نوع جلسه‌ها شروع می‌کنیم: جلسات ایستاده دوره‌ای^۱

معمولاً این جلسه هر هفته انجام می‌شود و محتوای آن باید به معرفی افراد یا تغییرات جدید و همین‌طور اطلاعیه‌های ساده محدود شود. اعلان آخرین وضعیت به نوبت از تک‌تک افراد گروه صرف‌نظر از اینکه آیا خبر جدیدی دارند یا نه، حوصله‌سَربر و باعث اتلاف وقت است. این‌طور جلسات با شما کاری می‌کنند که دل‌تان می‌خواهد با مشت به گلوئی خودتان بزنید تا هرچه سریع‌تر تمام شوند!

هر موضوعی که نیاز به بحث عمیق‌تر داشته باشد باید خارج از این نوع جلسات و فقط با حضور افراد مرتبط مورد بررسی قرار بگیرد. کسی که مدیریت جلسه را به عهده دارد باید از باز شدن یک موضوع خاص در جلسه جلوگیری کند و لیستی از موضوعات «حاشیه» تهیه کند تا بعد از جلسه، فقط اعضای مربوط به آن‌ها رسیدگی کنند. اگر این کار در تیم شما تبدیل به یک عادت شود، خیلی راحت می‌توان موضوع جلسه‌های ایستاده روزانه را حول اعلان وضعیت‌های کلی متمرکز کرد. نکته مهم در این نوع از جلسه‌ها این است که اعضای تیم باید با ترک جلسه وقتی که اعلان‌های اصلی تمام شد، احساس راحتی کنند. اگر موضوعی برای اعلان وجود نداشته باشد، یا یک ایمیل برای اشتراک‌گذاری آخرین وضعیت‌ها کافی باشد، شک نکنید که می‌توانید از جلسه گرفتن صرف‌نظر کنید. زیاد دیده شده که افراد به جلسات اعلان وضعیت می‌روند چون آنجا را جایی برای اطلاع پیدا کردن از آخرین اخبار می‌بینند و احساس می‌کنند که اگر شرکت نکنند، از تحولات تیمی جدا می‌مانند. مته به خشخاش نمی‌گذاریم، ولی این کار به وضوح دیوانگی‌ست!

جلسات ایستاده روزانه

بعضی از مهندسين روی این نوع جلسات روزانه قسم می‌خورند و معتقدند که از آخرین اصول علمی توسعه نرم‌افزار به روش چابک^۱ پیروی می‌کنند. این جلسات قابل قبول‌اند اگر که کوتاه، مختصر و بدون حاشیه باشند. معمولاً در ابتدا این نوع از جلسه‌ها کوتاه شروع می‌شوند (حدود پانزده دقیقه)، به این طریق که همه حقیقتاً از جا بلند می‌شوند و در مورد آخرین وضعیت کار خود به دیگران یک گزارش ساده می‌دهند؛ اما بدون مراقبت و احتیاط مداوم، خیلی زود این جلسات می‌توانند به دورهمی‌های طولانی سی دقیقه‌ای یا بیشتر تبدیل شوند که افراد حاضر، بی‌هدف در مورد کار خود توضیحات مفصل می‌دهند. گاهی اوقات افراد این‌طور نشست‌ها را با جلسه‌های روان‌درمانی اشتباه می‌گیرند! اگر تیم شما به جلسات ایستاده روزانه احتیاج دارد، یک نفر باید تصدی امور را به دست گرفته و جلسه را مدیریت کند.

اگر در تلاش هستید تا چیزی جدید طراحی کنید، سعی کنید که جلسه شما بیشتر از پنج نفر نشود. تصمیم‌گیری و نقشه‌کشی با حضور بیشتر از پنج نفر در اتاق عملاً غیرممکن است، مگر آنکه فقط یک نفر تصمیم‌گیرنده اصلی باشد. اگر باور نمی‌کنید با پنج نفر از دوستانتان به مرکز شهر بروید و شش نفره سعی کنید سر یک مسیر برای بازدید از تعدادی از مراکز توریستی شهر توافق کنید. به احتمال قوی شما نیمی از روز را در یک گوشه از خیابان مشغول جروب‌بحث خواهید بود، مگر آنکه یک نفر را به عنوان داور منصف انتخاب کنید که جای تیم تصمیم‌گیری کند.

1. Agile



اصولاً جلسه‌ها باعث تداخل در زمان مفیدی می‌شوند که ما با الهام از پال گراهام^۱ آن را «زمان سازندگی» می‌نامیم. برای هر کسی مخصوصاً برنامه‌نویسان، ورود به «قلمرو تمرکز» در کنار مزاحمت جلسه‌های گاه و بیگاه، می‌تواند دشوار باشد. در تقویم خود محدوده‌های زمانی سه یا چهار ساعته را تحت عنوان «مشغول» یا حتی «سازندگی» ثبت کنید و روی کار خودتان تمرکز کنید. اگر مجبورید که یک جلسه را برنامه‌ریزی کنید، سعی کنید که زمان جلسه بلافاصله بعد یا قبل از یک وقفهٔ دیگر مثل ساعت ناهار یا انتهای روز باشد. در گوگل یک سنت قدیمی – که متأسفانه گاهی اوقات نادیده گرفته می‌شود – تحت عنوان «پنجشنبه‌های بی‌جلسه»^۲ وجود دارد تا افراد بتوانند از زمان خود برای تمرکز روی کار استفاده کنند. این می‌تواند یک قدم خوب اولیه باشد در مسیری به سمت بیست یا شاید سی ساعت زمان سازندگی در بازه‌های زمانی بزرگ‌تر.

1. <http://www.paulgraham.com/makersschedule.html>

۲. یکی از معاونین شاخهٔ مهندسی گوگل، وین روزینگ، این سنت را در سال ۲۰۰۱ با هدف افزایش کیفیت زندگی مهندسين آغاز کرد. فیتز همیشه پنجشنبه‌های خودش را کامل مسدود می‌کرد تا کسی با او جلسه نگذارد. این کار نسبتاً خوب جواب می‌داد، هر چند که باید با دقت از آن مواظبت می‌کرد و هر از گاهی نیز نیاز بود تا به افرادی که در روز پنجشنبه جلسه تنظیم کرده‌اند، ایمیل‌های نه چندان خوش‌اخلاقانه‌ای ارسال کند.

پنج قانون ساده برای مدیریت جلسات:

- فقط افرادی را دعوت کنید که قطعاً نیاز است در جلسه باشند.
- دستور جلسه داشته باشید و قبل از جلسه آن را به تمام افراد ارسال کنید.
- اگر به اهداف مورد نظر در جلسه رسیدید، جلسه را زودتر تمام کنید.
- از به حاشیه کشیده شدن جلسه جلوگیری کنید.
- سعی کنید که زمان جلسه نزدیک به سایر وقفه‌های روزانه مثل ساعت ناهار یا انتهای روز باشد.

اگر قرار است جلسه‌ای را تنظیم کنید، یک دستور جلسه آماده کنید که حاوی موضوعاتی باشد که قرار است در جلسه مورد بحث قرار بگیرند و آن را حداقل یک روز قبل از جلسه برای تمام افراد ارسال کنید تا بدانند چه چیزی در انتظارشان است. تا جایی که می‌شود تعداد کمی را به جلسه دعوت کنید: فراموش نکنید که ارتباطات همگام چقدر هزینه‌بر هستند. ما افرادی را می‌شناسیم که آشکارا به دعوت‌هایی که حاوی دستور جلسه نیستند بی‌محرمانگی می‌کنند.

فقط افرادی را دعوت کنید که نیاز است در جهت اهداف جلسه حضور داشته باشند. بعضی از افراد بعد از اینکه دیدند گاهی اوقات عده‌ای در جلسه مشغول پاسخ دادن به ایمیل یا کار شخصی هستند، ورود لپ‌تاپ‌ها به جلسات را ممنوع کردند. اما این صرفاً راه‌حلی است برای یکی از علائم مشکل و نه راه‌حل علت اصلی مسئله. کسانی که در جلسه مشغول خواندن ایمیل‌های خودشان هستند، احتمالاً از همان ابتدا نیازی به حضور در جلسه نداشتند.

مسئول جلسه باید واقعاً مسئولیت جلسه را به عهده گرفته و در قطع کردن صحبت‌های حاشیه‌ای یا افرادی که موضوع را انحصاری می‌کنند، دریغ نکند. این کار، هر چند سخت ولی مفید و باارزش خواهد بود. مهم‌تر از هر چیزی، از اتمام جلسه زودتر از موعد و وقتی که اهداف جلسه به دست آمده، نترسید.

کار کردن در یک تیم با «چالش‌های جغرافیایی»

وقتی عضو تیمی هستید که اعضای آن در شهرهای مختلف پخش هستند، یا شما از سایر تیم‌ها دور هستید، نه‌تنها باید روش‌های متفاوتی برای برقراری ارتباط پیدا کنید، بلکه مجبور خواهید بود تلاش بیشتری در ارتباطات خود داشته باشید. وقتی اعضای تیم از هم دور باشند، تصمیم‌ها باید مکتوب شوند و معمولاً با ایمیل بین کل تیم به اشتراک گذاشته شود. اگرچه گفت‌وگوها ممکن است از طریق سیستم‌های پیام‌رسان آنلاین انجام شود، تیم باید راهی پیدا کند که اطلاعات برای همه در دسترس باشد. تماس‌های ویدئویی نیز روش خیلی خوبی برای برقراری تماس‌های کوتاه هستند، مخصوصاً که این روزها تقریباً همه لپ‌تاپ‌ها مجهز به دوربین هستند.

ما در پروژه ساب‌ورژن یک شعار داشتیم: «اگر صحبتی در ایمیل نیامده باشد، هیچ‌وقت مطرح نشده است.» افراد ممکن است در تالارهای آنلاین مدت زمان زیادی را صرف گفت‌وگو و ایده‌پردازی کنند، ولی برای اینکه مصوبه‌ها را «واقعی» کنند، باید به فکر سایر افراد تیم که در تالار آنلاین حضور نداشتند نیز باشند. با اجبار به اشتراک‌گذاری این نوع گفت‌وگوها در ایمیل برای کل تیم، به کل اعضای تیم این فرصت را می‌دهیم تا ببینند چطور یک تصمیم گرفته شده است (و اگر نیاز است، نظر خودشان را نیز مطرح کنند). این موضوع به‌خصوص برای تشکیل تیم‌های «اجماع محور» اهمیت دارد.

صحبت کردن با یک نفر در یک مکان دور باید به اندازه سر زدن به میز کارشان، بی‌دردسر باشد. اگر شما از سایر افراد تیم دور هستید، خیلی بیشتر از همیشه با سایر افراد تیم (از روش‌های مختلف مثل نرم‌افزارهای پیام‌رسان، تالارهای گفت‌وگوی آنلاین، ایمیل، تماس ویدئویی یا تلفنی) ارتباط داشته باشید. مطمئن شوید که همه نه تنها از حضور شما مطلع هستند، بلکه در جریان کاری که شما انجام می‌دهید نیز هستند. مهم‌تر از همه، قدرت گفت‌وگوهای دونفره را دست‌کم نگیرید.

زمانی بن در تیم شخصی خود فردی به اسم گری داشت که در دفتر شرکت در یک شهر دیگر روی یک پروژه جدید کار می‌کرد. گری در برقراری ارتباط با سایر اعضای تیم با دشواری‌های زیادی روبه‌رو بود. او این مشکل را در جلسه هفتگی با بن مطرح کرد و بن از او خواست که یک هفته به دفتر کار سایر اعضای تیم بیاید تا پروژه را با هم شروع کنند. گری در ابتدا به خاطر هزینه‌های پرواز و هتل نسبت به این سفر دودل بود و به فواید آن خیلی فکر نمی‌کرد. نهایتاً با یک سفر دو روزه پیش سایر افراد تیم آمد و خیلی سریع متوجه مزایای دیدار هم‌تیمی‌ها شد. نه تنها از مکالمات راحت و حضوری با هم‌تیمی‌های جدید خودش استفاده کرد، بلکه با ناهار خوردن‌های مشترک و بیرون رفتن‌های تیمی با خصوصیات اخلاقی آن‌ها نیز آشنا شد. گری و سایر افراد تیم همدیگر را به عنوان یک انسان بیشتر شناختند. در نتیجه، مکالمات بعدی راحت‌تر ادامه پیدا کرد، با وجود اینکه گری هزار مایل از سایرین دور بود.

هیچ چیز جای دیدار حضوری را نمی‌گیرد

نکته‌ای که در اینجا قابل توجه است این است که با وجود تمام پیشرفت‌های شبکه‌های اجتماعی و تکنولوژی‌های تماس ویدئویی، هنوز هیچ چیز نمی‌تواند جای صمیمیت و راحتی ارتباطات حضوری و واقعی دو نفر را بگیرد. اگر شما می‌خواهید پروژه جدیدی را با یک نفر شروع کنید یا صحبت خیلی مهمی با یک نفر در شرکتتان دارید و بودجه سفر نیز در دسترس است، تقریباً همیشه زحمت و هزینه دیدار حضوری به صرفه خواهد بود. صحبت حضوری اثری در حافظه انسان می‌گذارد که هیچ صحبت تلفنی یا تصویری‌ای نمی‌تواند اثری مشابه یا حتی نزدیک داشته باشد. خیلی از افراد به دلیل هزینه‌های سنگین سفر، با این ایده موافق نیستند. درست است که شاید

شرکت‌های کوچک نتوانند از عهده هزینه چنین سفرهایی برآیند، ولی بیشتر شرکت‌های بزرگ‌تر استطاعت مالی این کار را دارند. هزینه عدم دیدار حضوری اعضای تیم، به مراتب بیشتر از آن چیزی است که فکر می‌کنید.

صرف نظر از اینکه چقدر با همکاران خود از طریق ایمیل یا تماس تلفنی و ویدئویی ارتباط دارید، هر از گاهی بلیت هواپیما بگیرید و با هم‌تیمی‌ها دیدار کنید. این قضیه نه تنها برای کارمندان و تیم‌های دورکار، بلکه برای افرادی که در دفتر غیر اصلی شرکت کار می‌کنند نیز صدق می‌کند. هر از گاهی به شهری که دفتر اصلی شرکت در آنجاست سفر کنید و با اعضای شرکت به شکل حضوری صحبت کنید.

مستندات طراحی نرم افزار

برای مهندسين نرم افزار بسيار سخت است تا بتوانند هنگام شروع يك پروژه، جلوي تمايل شديد به كدنويسي را بگيرند و روي مستندات نوشتاري طرح و برنامه ريزي اوليه پروژه كار كنند. ولي كدنويسي بدون طرح و برنامه معمولاً نتيجه مفيدي نمي دهد، مگر آنكه بخواهيد روي يك نسخه آزمائشي يا نمونه اوليه يك ايده كار كنيد. بيشتر مهندسين قبل از طراحي و نقشه كشي كلي نرم افزار، براي كدنويسي عجله مي كنند و اين كار پروژه را به سمت شكست هدايت مي كند.

يك مستند مکتوب طرح نرم افزار معمولاً يك مالک دارد و دو يا سه نفر آن را نوشته اند و عده بيشتري آن را مرور کرده اند. اين نوشته، نه تنها برنامه كلي آينده پروژه شما را مشخص مي كند، بلکه يك روش بهينه و به صرفه ارتباطي براي در جريان گذاشتن تمام اعضاي تيم درباره چرائي و چگونگي جزئيات فني پروژه است. در اين مرحله، چون هنوز هفته ها (يا ماه ها) را صرف كدنويسي نکرده ايد، قبول کردن انتقادات و عملي کردن پيشنهادهاي بسيار راحت تر است و نهايتاً شما يك محصول بهتر با پياده سازي مفيدتر خواهيد داشت. از اين گذشته، مستند طرح نرم افزار شما مي تواند به شما در زمان بندي پروژه و تقسيم كار بين اعضا نيز كمك كند. وقتي كار برنامه نويسي را شروع كرديد، توجه كنيد كه مستند طرح نرم افزار يك نوشته زنده است و مطالب آن روي سنگ حكاكي نشده اند. شما و تيم شما همراه با رشد و پيشرفت پروژه بايد طرح و نقشه نرم افزار را مرتباً به روز کرده و تغييراتي را كه به مرور ايجاد کرده ايد مستند كنيد. البته كه به عمل كار برآيد و به سخن داني نيست. بيشتر تيم ها در عمل هيچ نوشته مستندي ندارند و بعضي ديگر مدت زمان نسبتاً زيادي در ابتدای كار، مستندات طراحي فوق العاده اي آماده مي كنند، ولي به مرور آن ها را به روز نمي كنند.

تمام اين صحبت ها را كرديم، ولي دقت كنيد تا از آن سمت بوم نيفتيد! ما آدم هاي وسواسي را هم ديده ايم كه براي يك برنامه ساده و کوتاه صد خطي، چهار صفحه طراحي نوشته اند. اگر مدت زماني كه طول مي كشد تا كل پروژه را از ابتدا پياده سازي كنيد، كمتر از مدت زماني است كه صرف نوشتن طرح و نقشه نرم افزار کرده ايد، اين مستند سازي به وضوح يك اتلاف وقت بوده است. از تجربه خود براي قضاوت در مورد نياز يا عدم نياز به طرح نرم افزار استفاده كنيد.

مکالمات روزمره

با فرض بر اینکه اهداف کلی پروژه مورد تأیید تمام اعضای تیم باشد، هنوز باید به چگونگی مکالمات روزانه و ابزارهای ارتباطی افراد با یکدیگر توجه کنید. وسایل ارتباطی اگرچه مفید هستند، ولی از توانایی‌هایی محدودی برخوردارند و به ما امکاناتی نظیر مشاهده حالت چهره یا زبان بدن را نمی‌دهند. بنابراین امکان سوءتفاهم و آسیب به اصول HRT را افزایش می‌دهند. با وجود این باید گفت که ابزارهای ارتباطی برای بیشتر تیم‌ها فوق‌العاده باارزش هستند و با کمی تلاش و دقت می‌توان از آن‌ها در جهت افزایش بهره‌وری تیم استفاده کرد.

فهرست‌های ایمیل

امروز ما شخصی را نمی‌شناسیم که عضوی از تیمی باشد که حداقل از یک فهرست ایمیل استفاده نکند. با دقت به یک سری نکات می‌توان از این فهرست‌های ایمیل به شکل مفیدتری استفاده کرد.

تعداد زیادی از پروژه‌های بزرگ و موفق چندین فهرست ایمیل متفاوت دارند و صحبت‌هایی با موضوعات متفاوت نظیر برنامه‌نویسی، مرور کدها، مذاکره با کاربران، اطلاعیه‌ها و امور مدیریتی و متفرقه را از همدیگر جدا می‌کنند. بعضی اوقات پروژه‌های کوچک‌تر سعی می‌کنند از آن‌ها الگو برداری کنند و چندین فهرست ایمیل متفاوت درست می‌کنند، در حالی که آن‌ها صرفاً سه برنامه‌نویس و دو کاربر دارند. کار این پروژه‌های کوچک‌تر مثل این است که شش اتاق کنفرانس را برای صحبت بین فقط پنج نفر تهیه کنید. نتیجه تعدادی اتاق خالی، صحبت‌های غیر منسجم و تکراری خواهد بود. بهترین کار این است که همیشه با یک فهرست ایمیل آغاز کنید و فقط زمانی فهرست بعدی را اضافه کنید که مدیریت صحبت‌ها دشوار شود. یک استثنا در این قانون البته این است که می‌توان ایمیل‌های اتوماتیک یا گزارش‌هایی را که «بات»‌ها صادر می‌کنند در یک فهرست ایمیل جداگانه قرار داد.

آداب و رسوم مناسبی در صحبت‌ها و مکالمه‌های انجام‌شده در ایمیل‌ها تعریف کنید و اجازه ندهید اقلیتی با بیشتر صحبت کردن، کنترل مذاکره‌ها را به دست بگیرند.^۱

وقتی که شما با تیمتان یک دفتر یا اتاق مشترک دارید، مکالمه ایمیلی احتمالاً انتخاب اول شما برای برقراری ارتباط با سایر اعضا نخواهد بود. ولی بهتر است تا یک رونوشت از مصوبات جلسات، تصمیم‌های گرفته‌شده،

۱. اقلیت پرصدا معمولاً یک یا دو نفر هستند که مدام به تمام پیام‌ها جواب می‌دهند و نظرانی را که بر خلاف میلشان است رد می‌کنند. یک نگاه سرسری به مکالماتی که اقلیت پرصدا در آن‌ها مشارکت دارند به شما این تصویر اشتباه را می‌دهد که فکر کنید تیم شما مملو از ناراضی‌هاست. این رفتار نیازمند توجه سریع و با احتیاط از جانب شماست. فصل چهارم نحوه برخورد با این‌طور آدم‌ها را توضیح می‌دهد.

مستندات طرح نرم‌افزار و یا هرگونه اطلاعات رد و بدل شدهٔ دیگر را در قالب ایمیل مکتوب کنید تا مرجع اطلاعاتی مناسب و راحتی برای آینده تهیه شود. پیام‌های رد و بدل شده در ایمیل‌ها را در قالب یک پایگاه داده با قابلیت جست‌وجو آرشیو کنید و اگر پروژه شما به صورت متن‌باز اداره می‌شود، این مرجع داده را در دسترس همگان روی اینترنت قرار دهید. بدین صورت شما یک مکان متمرکز اطلاعات شامل تاریخچهٔ پروژه خواهید داشت که حتی در جریان قرار دادن سریع اعضای جدید تیم را نیز برایتان آسان‌تر خواهد کرد. اگر گفت‌وگوهای تیم را جایی ثبت نکنید، صحبت‌ها و بحث‌ها مرتباً تکرار خواهند شد.

پیام‌های آنلاین

نرم‌افزارهای پیام‌رسان آنلاین یک ابزار خارق‌العاده برای گفت‌وگو بین اعضای یک تیم است. مخصوصاً که این امکان را به کاربر می‌دهند تا پیامی را برای شخص دیگری بفرستند، بدون آنکه مزاحم روند کاری و تمرکز او بشوند (البته مستلزم این است که گیرنده، نرم‌افزار پیام‌رسان خود را طوری تنظیم کرده باشد که پیام‌ها کار او را قطع نکنند). پیام‌رسان‌ها، همچنین ابزار مناسبی هستند برای تیم‌هایی که سریع عمل می‌کنند و برای عده‌ای که آخر هفته یا بعد از ساعات اداری نیز کمی کار می‌کنند و یا برای افرادی که پس از بازگشت از مرخصی می‌خواهند پیام‌های خود را ببینند. پیام‌های خصوصی بین دو نفر نیز گاهی اوقات ابزار مفید و کارآمدی است، ولی ما شدیداً پیشنهاد می‌کنیم که از تالارها و کانال‌های گروهی برای پیام‌رسانی بین اعضای تیم استفاده شود.

سال‌ها قبل وقتی نرم‌افزارهای پیام‌رسان این‌چنین محبوب و معمول نبودند، تیم‌های برنامه‌نویسی به پروتکل Internet Relay Chat یا به اختصار IRC اتکا می‌کردند و بیشتر مکالمه‌های خود را در قالب کانال‌های گروهی انجام می‌دادند. گاهی اوقات این صحبت‌ها شلوغ و پرهممه می‌شد، ولی اگر موضوع صحبت برای کل تیم مفید نبود، افراد همچنان می‌توانستند از گروه خارج شده و مکالمات خصوصی دونفره داشته باشند. با وجود این، بیشتر صحبت‌ها در مقابل کل اعضای تیم انجام می‌شد و فرصتی را برای تمام افراد تیم فراهم می‌کرد تا در جریان همهٔ مکالمات قرار بگیرند یا حتی صحبت‌های گذشته را که از دست داده بودند پیگیری کنند. این راه مناسب و آسان، نه تنها به اعضای تیم اجازه می‌دهد که به سادگی با یکدیگر مکالمه کنند، بلکه به ارتباطات تیمی وقتی افراد در فواصل مختلف جغرافیایی از یکدیگر قرار دارند نیز کمک می‌کند. بیشتر اوقات شگفت‌زده خواهید شد از اینکه یک عضو جدید در تیم صرفاً با دنبال کردن و مشاهدهٔ مکالمات مختلف در کانال پیام‌رسان تیم، چقدر مطلب جدید یاد می‌گیرد.

نرم‌افزارهای پیام‌رسان در یک برههٔ زمانی باعث شدند که مکالماتی که قبل‌ها به صورت گروهی در IRC انجام می‌شد، به صحبت‌های خصوصی و دونفره تبدیل شوند، چرا که این نوع مکالمه، تنظیم پیش‌فرض اکثر این نرم‌افزارها بود. برای برنامه‌نویس‌ها راحت‌تر بود تا سؤال خود را به طور خصوصی از یکی از هم‌تیمی‌هایشان بپرسند تا اینکه بخواهند با عدم اعتماد به نفسشان مبارزه کرده و سؤالی را که شاید به نظرشان احمقانه بیاید روبه‌روی تمام

اعضای تیم و در گروه بپرسند. متأسفانه این کار باعث شد که بار روی تیم بیشتر شود، چرا که هیچ مکان متمرکزی برای اطلاعات وجود نداشت و امکان داشت یک سؤال مرتباً بین اعضای تیم تکرار شود.

خوشبختانه حدود سال‌های ۲۰۱۴ تا ۲۰۱۵، ورود اسلک^۱ به عنوان یک نرم‌افزار رایگان برای مکالمات آنلاین گروهی یک تجربهٔ مدرن و جایگزین IRC را ارائه داد. قابلیت یکپارچگی و همبستگی اسلک با سایر نرم‌افزارها باعث شد که این نرم‌افزار تبدیل به انتخاب اول شرکت‌ها برای مکالمات درون‌تیمی شود. انجمن‌های گستردهٔ آنلاین نیز برای مکالمات بین اعضای خود نیز شروع به استفاده از این سرویس کرده‌اند. با وجود آنکه همچنان می‌توان از اسلک به عنوان ابزاری برای مکالمات مستقیم و خصوصی دونفره استفاده کرد، این نرم‌افزار هر هفته گزارشی از نسبت کلی مکالمات خصوصی به گروهی ارائه می‌دهد تا شما بتوانید متوجه شوید که آیا لازم است تیمتان را بیشتر به مکالمات گروهی تشویق کنید یا خیر.

صرف‌نظر از اینکه شما از چه نرم‌افزار پیام‌رسانی استفاده می‌کنید، ما به جد توصیه می‌کنیم که برای تیمتان امکانات ساده و در دسترس برای مکالمات گروهی را فراهم کنید. برقراری کانال‌های اضافه برای راحت کردن ارتباطات تیمی قطعاً ارزش تلاش مضاعف را دارد.

1. Slack

مکالمات گروهی یا یک به یک‌های خصوصی؟

امروزه بیشتر افراد وقتی اسم IRC را می‌شنوند یاد محیط ماقبل تاریخ مبتنی بر نوشتار آن می‌افتند، چرا که حتی نرم‌افزارهای مدرن تر IRC نیز همچنان قیافهٔ منسوخ و ازمدافته‌ای دارند. گول ظاهر قدیمی آن‌ها را نخورید. قابلیت خارق‌العادهٔ آن‌ها این است که می‌توان به سادگی مکالمات گروهی و غیر هم‌زمان را برقرار کرد. در بیشتر نرم‌افزارهای مبتنی بر IRC می‌توان تاریخچهٔ پیام‌های گذشته را تا بی‌نهایت مشاهده کرد و مکالمات بین سایر افراد را وقتی که شما در گروه حضور نداشتید هم مرور کرد. نرم‌افزاری مثل اسلک، یک نسخهٔ مدرن و به‌روز از همان نرم‌افزارهای IRC است و با وجود همهٔ زلم زیمبوه‌های گرافیکی و اسمایلی‌های متنوع در اصل همان قابلیت‌های اصلی IRC را ارائه می‌دهد. امکانات جانبی و تجملی نظیر مکالمات ویدئویی، تخته وایت‌بردهای اشتراکی و ... اگرچه خیلی وسوسه‌انگیزند، ولی ممکن است که شما را از هدف اصلی که همان ارتباطات غیر هم‌زمان باشد، دور کند. اگر می‌خواهید از نرم‌افزاری غیر از اسلک یا نرم‌افزارهای مبتنی بر IRC استفاده کنید، مطمئن شوید که امکانات مکالمهٔ گروهی را به خوبی ارائه می‌دهد.

بعضی از آدم‌ها با چت کردن آنلاین و صحبت در نرم‌افزارهای پیام‌رسان راحت‌تر هستند. یک بار در یک مسابقهٔ برنامه‌نویسی با گروهی از برنامه‌نویسان که بعضی‌ها را هیچ‌وقت حضوری ندیده بودیم قرار گذاشته بودیم تا دور هم نشسته و روی یک پروژه کار کنیم. وقتی وارد شدیم با یک سالن تقریباً ساکت با حدود ده دوازده میز که سر هر میز شش تا هشت برنامه‌نویس مشغول تایپ کردن بودند مواجه شدیم. فکر کردیم که خوب شاید دیر رسیده‌ایم و همه شروع به برنامه‌نویسی کرده‌اند. بنابراین لپ‌تاپ‌هایمان را باز کرده و با این فرض که احتمالاً تعدادی موفق نشده‌اند خودشان را به محل برسانند و از طریق اینترنت با ما خواهند بود، وارد گروه مکالمات آنلاین تیممان شدیم. در میان یک سری صحبت‌هایی که در حال انجام بود، سلام کردیم و به همه اطلاع دادیم که ما به محل مسابقه رسیدیم. در کمال تعجب دیدیم که بقیه هم سلام کردند و گفتند که در همان اتاق و در چند متری ما نشسته‌اند. تا حدودی شاید علت این کار این بود که همه به صحبت‌های آنلاین عادت داشتند، ولی احتمالاً علت اصلی این بود که عدهٔ زیادی مکالمات آنلاین را به دیدار حضوری ترجیح می‌دادند و در کانال‌های IRC احساس راحتی بیشتری داشتند. ما که به تازگی از یک پرواز چهار ساعته آمده بودیم و دلمان کمی معاشرت حضوری می‌خواست، بلند شدیم و سر میزهای تک‌تک اعضای تیم رفتیم تا رو در رو سلام و احوال‌پرسی کنیم.

اینکه چه زمانی بهتر است از پیام آنلاین استفاده کرد و چه زمانی ایمیل، هیچ قاعده و قانونی ندارد. پیام‌های آنلاین برای مکالمات سریع که یک تصمیم آنی می‌تواند با حضور همهٔ افراد گرفته شود مفیدتر است. اگر که همهٔ اعضای تیم حضور ندارند و برای گرفتن تصمیم می‌توان صبر کرد، شاید ایمیل گزینهٔ بهتری باشد. ولی نکتهٔ مهم و قابل توجه این است که باید حواسمان را دربارهٔ هزینهٔ ارتباطات هم‌زمان در مقابل غیر هم‌زمان (همان طور که در بخش «الگوهای ارتباطی در فرهنگ‌های موفق» توضیح داده شد) جمع کنیم.

نرم افزار مدیریت پروژه

اگر از نرم افزارهای مدیریت پروژه برای رهگیری کارها و باگها استفاده می کنید (که بهتر است بکنید)، باید سیستم و پروسه‌ای برای رده بندی و برنامه ریزی فعالیتها تعریف کنید تا اعضای تیم بتوانند کارها و رفع اشکالات را بر اساس اولویت صحیح پیگیری کنند. اگر این کار را نکنید، کاربران و اعضای تیم به مرور حوصله شان برای ثبت گزارش اشکالات و کارها را از دست داده و شکایات خود از محصول شما را پیگیری نمی کنند. در این حالت اگر حتی نهایتاً شما به نرم افزار مدیریت پروژه خود سر بزیند، احتمالاً روی مشکلاتی کار خواهید کرد که لزوماً اهمیت بالایی نخواهند داشت.

توجه داشته باشید که یک نرم افزار پیگیری باگ یا کارهای پروژه در عمل یک تالار گفت و گوی اینترنتی یا تابلو اعلانات است. تمام خصوصیتی که باید در مکالمات آنلاین نظیر ارتباطات ایمیلی رعایت کرد، در مورد این نرم افزارها نیز صدق می کنند. صحبت هایی که در مکان های مختلف در مورد یک موضوع یا باگ نرم افزاری انجام می شود باید در این نرم افزار و برای استفاده همه اعضای تیم نیز ثبت شوند تا تفکرات و تصمیم گیری ها به شکل «رسمی» مستند شوند. طبیعتاً لحن صحبت ها باید مؤدبانه و محترمانه باشد.

بعضی اوقات دیده شده که بحث روی یک ایراد نرم افزار یا اولویت برای اضافه کردن یک قابلیت خاص درون چنین نرم افزارهایی بسیار بالا می گیرد. مخصوصاً وقتی که مدیر پروژه یک دور لیست کارها و باگها را زیر و رو کرده و از اعضای تیم گزارش می خواهد. اگر مناظره و مباحثه ها طولانی شد، بهتر است آن ها را به مکان دیگری مثل ایمیل ها منتقل کرد که جای بهتری برای این گونه بحث های درون تیمی است.

ارتباطات، بخشی از کار مهندسی

صدها کتاب متفاوت در مورد نحوه مهندسی نرم‌افزار و برنامه‌نویسی نوشته شده است. ما قصد نداریم که وارد جزئیات تکراری شویم، ولی نکات برجسته‌ای در مورد نحوه ارتباطات بین اعضای یک تیم برنامه‌نویسی وجود دارد که حائز اهمیت و قابل ذکر هستند. صرف‌نظر از اینکه از چه روشی برای برنامه‌نویسی استفاده می‌کنید، یا اصولاً عضوی از تیم مهندسی غیر نرم‌افزاری هستید، این قسمت درس‌هایی برای آموختن خواهد داشت. مخصوصاً درس‌هایی درباره آنچه که «نباید» انجام داد.

کامنت‌های روی کد

نحوه کامنت‌گذاری روی کد بین افراد متفاوت است. کامنت‌های طولانی معمولاً توضیح خیلی خوبی در مورد چرایی و چگونگی یک قطعه از کد ارائه می‌دهند و می‌توانند مفید باشند. ولی نگهداری و به‌روزرسانی این کامنت‌ها وقتی که کد دائماً در حال تغییر و تحول باشد بسیار سخت است و ممکن است به اطلاعات غلط و گمراه‌کننده منجر شود. به همین نحو، عدم کامنت‌گذاری کافی باعث سردرگمی برنامه‌نویسان آینده خواهد شد تا وقت زیادی را در پیدا کردن جواب سؤال‌هایشان تلف کنند. کامنت خوب، چرایی و چگونگی قطعه کد را توضیح می‌دهد، نه اینکه صرفاً نحوه کار کد را شرح دهد.

مفیدترین کامنت‌ها در سطح یک تابع یا API نوشته می‌شوند و وارد جزئیات به‌دردنخور نمی‌شوند. ضرب‌المثل یونانی "μηδέν ἄγαν" به معنی «مختصر و مفید» در مورد کامنت‌های ایدئال صدق می‌کند. علاوه بر آن، سبک و مدل کامنت‌گذاری روی کد نیز بهتر است که بین اعضای تیم منسجم و مورد توافق باشد. *اینکه / این سبک و سیاق کامنت‌گذاری چطور باشد خیلی اهمیت ندارد، ولی مهم است که در تیم منسجم و هماهنگ باشد.*^۱ برای سبک و سیاق کدنویسی و کامنت‌گذاری تیم یک راهنما تدوین کنید و دلایل اینکه چرا این سبک را انتخاب کرده‌اید شرح دهید. به عنوان مثال، به مقدمه راهنمای نحوه کدنویسی C++ در گوگل توجه کنید^۲:

زبان برنامه‌نویسی اصلی در بسیاری از پروژه‌های متن‌باز در گوگل، C++ است. همان طور که هر متخصص C++ می‌داند، این زبان، قابلیت‌های قدرتمند بسیار زیادی دارد. ولی این قدرت بالا، پیچیدگی‌های زیادی با خود به همراه دارد که در عمل می‌تواند باعث شود که کد شما نسبت به وجود باگ‌های متفاوت آسیب‌پذیرتر باشد و نگهداری طولانی‌مدت آن دشوار شود.

۱. قسمت کامنت‌گذاری در کتاب "The Art of Readable Code" نوشته Dustin Boswll در این باره بسیار مفید است.

2. <https://github.com/google/styleguide>

هدف این راهنمای کدنویسی این است که با بیان بایدها و نبایدهای کدنویسی، همین پیچیدگی C++ را مدیریت کند. قواعد و قوانین مطرح شده در این راهنما کمک می‌کند که ضمن کاهش پیچیدگی‌های کد نوشته شده، برنامه‌نویسان همچنان بتوانند از همه قابلیت‌های زبان استفاده کنند. سبک و استایل کدنویسی، قواعد و قوانینی هستند که باعث خوانایی هرچه بیشتر کد می‌شوند. «سبک و سیاق» شاید عنوان گمراه‌کننده‌ای برای این مجموعه قوانین باشد، چرا که این قوانین موضوعاتی فراتر از صرفاً نحوه کدنویسی را شامل می‌شوند.

یکی از راه‌های مدیریت بهتر کد، تأکید بر انسجام و همگون‌سازی است. اینکه هر برنامه‌نویسی بتواند با نگاه کردن به کد دیگری ایده کلی و هدف برنامه را به سرعت متوجه شود، بسیار حائز اهمیت است. پیروی از قاعده همگانی به ما کمک می‌کند که الگوهای متداول در کدهای نوشته شده در تیم را یاد بگیریم و با سرعت بیشتری کار را جلو ببریم. الگوهای مشترک و منسجم در تیم، درک و فهم برنامه را سریع‌تر و راحت‌تر می‌کند. اگرچه گاهی اوقات دلایل خوبی برای تغییر این قواعد و قوانین وجود دارند، ولی بهتر است که سعی کنیم آن‌ها را در طول مدت زمان یک پروژه ثابت نگه داریم.

موضوع دیگری که در راهنمای کدنویسی C++ مطرح شده است، لیست خصوصیات و قابلیت‌هایی است که استفاده از آن‌ها ممنوع است. زبان C++ بسیار وسیع و پیچیده است و در بعضی از موارد، یک سری قابلیت‌های پیشرفته این زبان را محدود یا حتی ممنوع می‌کنیم. این کار به ما کمک می‌کند که کد را ساده‌تر و خواناتر نگه داشته و جلوی اشکالات آینده را بگیریم. این راهنما، دلایل محدودیت‌ها و یا ممنوعیت‌ها را به شکل دقیق و مفصل توضیح می‌دهد. این راهنما، زبان C++ را آموزش نمی‌دهد. فرض بر این است که خواننده با این زبان برنامه‌نویسی آشنایی کامل دارد.

توجه کنید که این راهنما، ادعایی در مورد روش‌های بهینه استفاده از زبان C++ و بهترین و سریع‌ترین راه توسعه نرم‌افزار ندارد. بلکه صرفاً قواعد و قوانینی را مطرح می‌کند تا کدنویسی بین اعضای تیم منسجم و هماهنگ شود.

برچسب‌گذاری نامتان روی کار

هر کسی دوست دارد اعتبار کاری را که انجام می‌دهد به دست بیاورد. از هنرمندی که پایین نقاشی خودش را امضا می‌کند تا نویسنده‌ای که نام خود را روی کتابی که نوشته قرار می‌دهد. این ماهیت انسان است که دوست دارد کاری را که انجام داده به دیگران معرفی کند و آن را به همه بشناساند. ولی به عقیده ما، قرار دادن نام

برنامه‌نویس در ابتدای فایل‌های کد، باعث دردسر خواهد بود. همه ما چنین برچسب‌هایی ابتدای فایل‌های کد برنامه دیده‌ایم:

```
# -----
# Created: October 1998 by Brian W. Fitzpatrick <fitz@red-bean.com>
# -----
```

قرار دادن نام برنامه‌نویس در ابتدای فایل، یک رسم قدیمی است. راستش را بخواهید، ما هر دو در گذشته چنین اشتباهی را مرتکب شده‌ایم. شاید این کار در زمان‌های پیش که فایل‌های برنامه را معمولاً یک شخص تکمیل می‌کرد تا یک تیم، توجیه منطقی‌تری داشت. امروزه اما تعداد برنامه‌نویسان زیادی روی یک فایل کدنویسی می‌کنند و قرار دادن نام برنامه‌نویس در ابتدای فایل، موجب سوء تفاهم، اتلاف وقت، بحث و جدل‌های بیخود و ناراحتی افراد می‌شود. در نتیجه، ما عمیقاً مخالف قرار دادن نام برنامه‌نویسان روی فایل‌ها هستیم. شاید در یک سری شرایط خاص بتوان نام یک نفر به عنوان بهترین کاندید برای پاسخگویی به سؤالات احتمالی را روی یک فایل قرار داد، ولی به طور کلی از این کار پرهیز کنید.



فرض کنید به عنوان مثال، شما یک فایل جدید در پروژه تیمی خودتان درست کردید. چند صد خط کدنویسی کرده و نام خودتان به همراه علامت کی‌رایت را به ابتدای فایل چسبانده و برای بازبینی هم‌تیمی‌های خودتان

فرستادید. نهایتاً کدتان را بدون هیچ مشکل و گرفتاری‌ای ثبت می‌کنید. تا اینجای کار همه چیز خوب به نظر می‌رسد. چند روز بعد همکار شما، آدرین تغییراتی در این فایل اعمال می‌کند. چه زمانی او می‌تواند اسم خودش را بالای این فایل قرار دهد؟ آیا باید یک باگ را برطرف کند؟ ۵ تا باگ کافی است؟ آیا باید یک تابع اضافه کند؟ دو تابع؟ چند خط کد او را لایق اضافه کردن اسمش می‌کند؟ فرض کنیم که شخص سومی مدتی بعد تابعی را که آدرین نوشته بود بازنویسی می‌کند. آیا اکنون او باید نام خود را جایگزین نام آدرین کند؟

نرم‌افزار، بر خلاف سایر محصولات خلاقانه مثل نمایش‌نامه، فیلم یا رمان هیچ‌وقت «تمام» نمی‌شود. خیلی راحت در انتهای یک فیلم می‌توان اسامی همه دست‌اندرکاران را لیست کرد، در حالی که به‌روز نگه داشتن لیست افرادی که در توسعه یک نرم‌افزار نقش دارند، کاری است لایتناهی. شکی نیست که شما می‌توانید یک تکه کد را به تنهایی بنویسید، تمام حالات خاص و نیازهای آینده را پیش‌بینی کنید، دائماً و به تنهایی مشکلات کد را مکتوب کنید و به تمام سؤالات در قالب یک نوشته راهنما پاسخ دهید. ولی تمام این کارها از شما وقت زیادی را خواهد گرفت، وقتی که می‌تواند صرف کدنویسی شود. به همین دلیل است که ما توصیه می‌کنیم اسامی دست‌اندرکاران یک نرم‌افزار در سطح پروژه لیست شود، نه در سطح تک‌تک فایل‌های کد. بیشتر پروژه‌هایی که ما دیده‌ایم یک فایل برای نویسندگان^۱ یا دست‌اندرکاران^۲ دارند که شامل اسامی افرادی می‌شود که در پروژه دخیل بوده‌اند. اگر جزئیات بیشتری در مورد هر فایل می‌خواهید، سیستم مدیریت کد پروژه می‌تواند پاسخ دقیق‌تری به شما بدهد. طبیعتاً اگر از سیستم‌های مدیریت نسخه‌های کد استفاده نمی‌کنید، تمام تاریخچه پروژه شما مثل اشک‌هایی که زیر باران ریخته شوند، از بین می‌روند.

مرور و نقد کد را در تیم اجباری کنید

برای آنکه مطمئن شوید استانداردهای کدنویسی در تیمتان رعایت می‌شود، باید کدها را قبل از اینکه وارد محصول نهایی شوند، واری و مرور کنید. فرقی ندارد که در چه مرحله‌ای از کدنویسی این کار انجام می‌شود، ولی باید فرایندی تعریف کنید که ایجاب کند هر قطعه کدی که وارد پروژه می‌شود حداقل یک شخص ثانوی آن را مرور و تأیید کند تا مطمئن شویم که از اشتباهات پیش‌پاافتاده احتمالی جلوگیری کرده‌ایم. تغییرات در کد را در قالب قسمت‌های کوتاه کوتاه اعمال کنید تا هم‌تیمی‌های شما آن‌ها را راحت‌تر و سریع‌تر مرور کنند. این کار نه تنها باعث بالاتر رفتن کیفیت محصول نهایی می‌شود، بلکه غرور تیمی را نیز بین اعضای پروژه تقویت می‌کند؛ چرا که همه در تمام تغییرات پروژه حس مشارکت خواهند داشت.

1. Authors
2. Contributors

تست‌های واقعی و فرایند انتشار نرم‌افزار

صرف‌نظر از اینکه از فرایند توسعه نرم‌افزار آزمون‌محور^۱ استفاده می‌کنید یا صرفاً چند تست خودکار برای ارزیابی کلی سیستم خود دارید، لازم است که مراحل و فرایند دقیق و منسجمی برای انتشار نرم‌افزار خود تعریف کنید. هرچه تعداد تست‌های شما بیشتر باشد، با اطمینان بیشتری می‌توانید قابلیت‌های جدید را اضافه کنید یا مشکلات نرم‌افزار خود را برطرف کنید. تست‌های خودکار باید جزئی از مراحل اضافه شدن و مرور کد جدید در پروژه باشد. فرایند انتشار نرم‌افزار شما باید به قدری سبک و سریع باشد که شما بتوانید مکرراً (مثلاً هر هفته) یک نسخه جدید ارائه بدهید. ولی به همان اندازه باید کامل و دقیق باشد تا ایرادات احتمالی قبل از اینکه به کاربران شما برسند، پیدا شوند.

به هر حال، محصول نهایی مهم است!

تمام عادت‌ها و آداب و رسومی که در باب فرهنگ و اصول ارتباطات تیمی مطرح کردیم بر خلاف آنچه که به نظر می‌رسد، معمولاً بین تیم‌ها مشترک است. تجربه به ما نشان داده است که ساختن یک فرهنگ خوب تیمی که در آن اعضا به نحوه ارتباطشان با یکدیگر اهمیت می‌دهند، باعث می‌شود که وقت مفید بیشتری صرف نوشتن کد شود تا در راستای جروب‌های بی‌نتیجه تلف شود.

تیم‌های قدرتمند، تصادفی به وجود نمی‌آیند. رهبران و بنیان‌گذاران باتجربه، با دقت بذریک تیم قوی را می‌کارند و از آن نگهداری می‌کنند، چون با دردسرهای توسعه نرم‌افزاری که یک تیم به‌هم‌ریخته آن را انجام داده است آشنا هستند. چنین تلاشی از همان نقطه آغاز، باعث می‌شود تیمی تشکیل شود که خود به خود اشخاصی با فرهنگ و آداب مشابه را جذب می‌کند. از فواید جانبی این کار این است که فرایندهای کارآمد ارتباطی درون تیمی، دردسرهای شروع به کار را برای اعضای جدید تیم کاهش می‌دهد. بدون چنین عناصری، اعضای جدید وقت زیادی را در راستای یادگیری روش کار تیم تلف می‌کنند یا اگر موفق به یادگیری نشوند سعی می‌کنند که روش‌های خودشان را از تیم‌هایی که قبلاً عضو آن‌ها بودند، به تیم شما تحمیل کنند (که ممکن است خوب یا بد باشد).

اگرچه پیدا کردن اعضای مناسب جدید و تعریف کردن ارزش‌های درست برای تیم اهمیت بالایی دارند، ولی هیچ‌چیز به اندازه نحوه ارتباطات بین اعضا، در فرهنگ تیم شما تأثیر نمی‌گذارد. شرح مأموریت، فهرست ایمیل‌ها، کامنت‌های روی کد، مستندات مکتوب و حتی روش‌های تصمیم‌گیری در تیم، همگی در اصل نحوه ارتباطات تیم شما را تعریف می‌کنند. بارها آدم‌ها از دیدن میزان اهمیت آداب معاشرت تیمی چه از لحاظ روانی و چه از لحاظ زحمت و زمان در ساخت یک تیم قدرتمند، شگفت‌زده می‌شوند، ولی واقعیت همین است؛ نهایتاً محصول شما نیز با کاربران ارتباط خواهد داشت.

صرف‌نظر از آداب و رسوم ارتباطی و فرهنگ تیمی شما، هر گروه کارآمدی، در عمل یک رهبر دارد. در فصل بعد، به این نکته می‌پردازیم که ویژگی‌های یک رهبر تأثیرگذار چیست و چرا نقش او در تیم احتمالاً با چیزی که فکر می‌کنید متفاوت است. در مورد این موضوع نیز بحث می‌کنیم که چرا تمام اعضای تیم باید با اصول اولیه رهبری تیم آشنا شوند.

فصل سوم

هر قایقی یک کاپیتان لازم دارد

حتی اگر به جان مادرتان قسم خورده باشید که هیچ وقت نمی‌خواهید یک «مدیر» شوید، بالاخره در یک برهه از زندگی کاری‌تان تصادفاً مجبور خواهید شد که پروژه‌ای را مدیریت کنید. این فصل به شما یاد می‌دهد که در چنین حالتی چگونه عمل کنید.

ده‌ها کتاب برای مدیران با موضوع مدیریت نوشته شده است؛ اما این فصل برای آن دسته از اعضای تیم صحبت می‌کند که به طور اتفاقی خودشان را در جایگاه رهبری و مدیریت یک پروژه می‌بینند. بیشتر آدم‌ها به دلایل مختلف از مدیر بودن واهمه دارند، ولی هیچ تیمی نمی‌تواند بدون یک رهبر کارهایش را پیش ببرد. هدف ما این نیست که شما را متقاعد کنیم مدیر بشوید (هرچند که ما هر دو اکنون مدیر هستیم!)، بلکه ما به دنبال این هستیم تا به شما نشان دهیم چرا تیم‌ها به مدیریت نیاز دارند، چرا عده‌ای از رهبری تیمشان گریزان‌اند و چرا بهترین مدیران، تیمشان را بر پایه و اساس اصول تواضع، احترام و اطمینان مدیریت می‌کنند. علاوه بر آن، درباره‌ی الگوهای درست و غلط مدیریت نیز صحبت خواهیم کرد.

یادگیری زیر و بم مدیریت به شما کمک خواهد کرد که کنترل کار خودتان را به دست بگیرید. اگر می‌خواهید صرفاً یک مسافر نباشید و سکان قایق تیمتان را هدایت کنید، لازم است که اصول آن را یاد بگیرید تا قایق را در گل فرو نکنید.

طبیعت از خلأ بیزار است!

کشتی بدون کاپیتان، مثل یک اتاق انتظار شناور روی آب است. اگر کسی سکان را به دست نگیرد و موتور را روشن نکند، این کشتی سرگردان می‌شود و با حرکت موج دریا بی‌هدف به این سو و آن سو حرکت می‌کند. یک پروژه مهندسی نیز مانند یک کشتی است که اگر کسی فرمان را به دست نگیرد، صرفاً یک گروه از افراد را خواهید داشت که دور هم جمع شده و منتظرند اتفاق خاصی بیفتد.



صرف نظر از اینکه شخصی به عنوان مدیر منصوب شده باشد یا نه، بالاخره یک نفر باید هدایت پروژه را به دست بگیرد. در غیر این صورت پروژه به جایی نمی‌رسد. اگر شما شخص بانگیزه، ولی بی‌تاب و بی‌طاقتی باشید، ناگهان می‌بینید که فقط مشغول حل اختلاف بین اعضای تیم شده‌اید، تصمیم‌گیری‌های مهم را انجام می‌دهید و

۱. به فرضیه‌ای در علم فیزیک اشاره دارد (نسبت داده شده به ارسطو) که می‌گوید طبیعت به طور ذاتی، فضای تهی را پر می‌کند. Nature

(م.) abhors a vacuum.

هماهنگی‌های لازم را صورت می‌دهید. این اتفاق معمولاً به شکل تصادفی رخ می‌دهد. شما هیچ‌وقت تصمیم نمی‌گیرید که رهبر تیم باشید، ولی به هر حال این اتفاق می‌افتد. بعضی افراد به این مصیبت، بیماری "manageritis" می‌گویند.

مدیر، کلمه‌ای ناسزاگونه

مفهومی که امروزه به عنوان مدیریت می‌شناسیم، ریشه در سلسله‌مراتب نظامی و بعدها/انقلاب صنعتی^۱ قبل از قرن اخیر دارد. کارخانه‌های بی‌شمار در هر جایی که فکرش را بکنید ظاهر شدند و به کارگرانی (معمولاً با مهارت کمتر) نیاز داشتند تا خطوط تولید را در حرکت نگه دارند. برای اداره کارگران، نیاز به سرپرست بود و از آنجا که در بازار کار راکد آن زمان، جایگزین کردن یک کارگر با شخص جویای کار دیگری بسیار راحت بود، سرپرستان انگیزه‌ای برای رفتار مناسب با کارگران یا بهبود وضعیت کار آن‌ها نداشتند. صرف‌نظر از اینکه این کار آن‌ها انسانی بود یا نه، کارفرمایان به راحتی و برای سال‌های طولانی با این روش در استخدام کارگران برای کارهای ساده و تکراری موفق عمل کردند.

مدیران با کارگرانشان همان طور رفتار می‌کردند که ارباب‌سواران با اسب و الاغشان! انگیزه آن‌ها را با نشان دادن وعده هویج زیاد می‌کردند و وقتی این کار جواب نمی‌داد با چوب به جانشان می‌افتادند. این فلسفه مدیریتی «چوب و هویج» از عصر کارخانه‌ها^۲ به دنیای اداره‌های مدرن نیز راه پیدا کرد. کلیشه کلی از مدیران در اواسط قرن بیستم، یک انسان خشک و خشن توصیف می‌کرد که از کارمندانش بیگاری می‌کشید. کارمندان نیز یک شغل ثابت و کار تکراری را سال‌ها تکرار می‌کردند (چرا که حقوق بازنشستگی آن‌ها نیز به سنوات کارشان بستگی داشت).

هنوز هم در بعضی صنایع، حتی مکان‌هایی که در آن‌ها کار خلاقانه اهمیت دارد و با وجود آنکه تحقیقات و گزارشات زیادی^۳ در ردّ کارایی و مؤثر بودن روش «چوب و هویج» منتشر شده‌اند، این نوع از مدیریت منسوخ نشده و به بهره‌وری انسان‌های مبتکر و خلاق صدمات جدی می‌زند. اگرچه کارگران عصرهای گذشته می‌توانستند به راحتی جایگزین شده و صرف چند روز آموزش ببینند، امروزه شاید ماه‌ها زمان لازم باشد تا شخص جدیدی بتواند در یک تیم دانش‌محور مؤثر واقع شود. بر خلاف کارگران کارخانه‌ها، این کارمندان کار خلاقانه نیاز به فضا و زمان و محیطی دارند که بتوانند خوب فکر کنند و محصولات جدید بسازند.

۱. انقلاب صنعتی اروپا در قرن هجدهم و انقلاب صنعتی آمریکا در قرن نوزدهم.

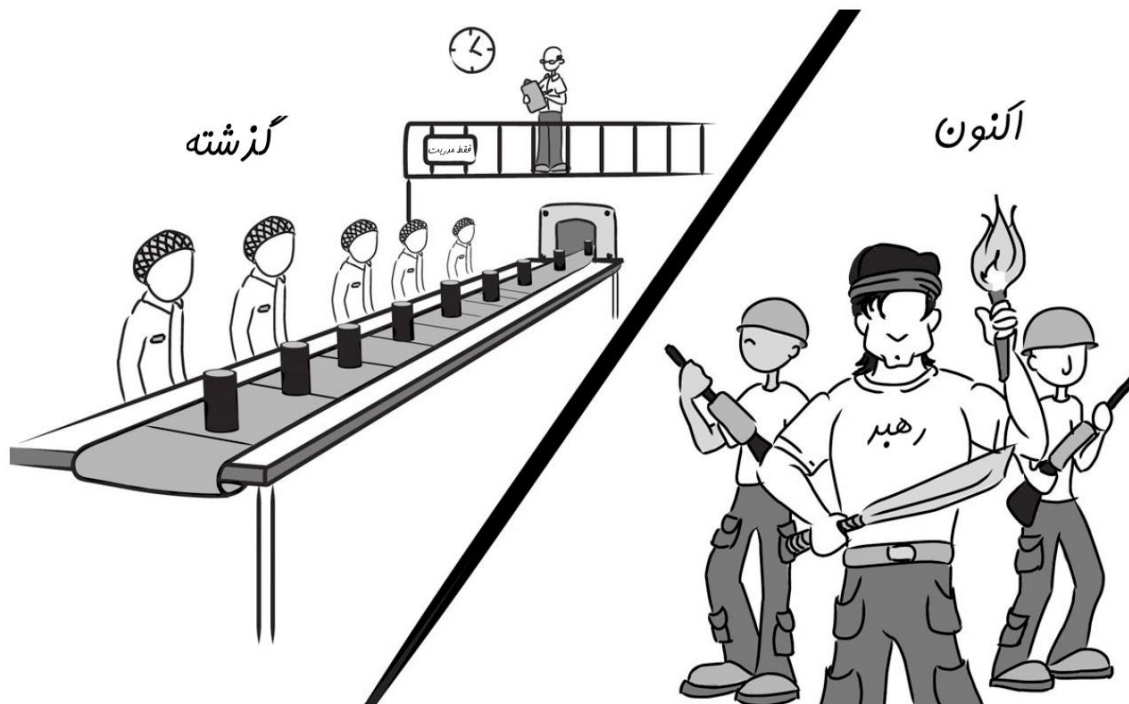
۲. کتاب Scientific Management or Taylorism به خوبی مبحث مدیریت کارمندان در تاریخ را مرور می‌کند.

۳. سخنرانی https://www.ted.com/talks/dan_pink_the_puzzle_of_motivation

«رهبر» به جای «مدیر»

ما معتقدیم که عنوان «مدیر» باید منسوخ شود و به جای آن از کلمه «رهبر» استفاده شود. درست است که هنوز جنبش «نه به کلمه مدیر» را راه نینداخته‌ایم، ولی معتقدیم که صرف کلمه مدیر با خودش بار معنایی رئیس و زبردست را به همراه دارد. مدیران معمولاً مثل پدر و مادر در قبال فرزندانشان رفتار می‌کنند^۱ و در نتیجه اعضای تیمشان نیز مثل کودکان عمل می‌کنند. از دیدگاه اصول HRT به قضیه نگاه کنیم: اگر مدیری به یک عضو از تیم خودش اعتماد داشته باشد، عضو تیم هم احساس می‌کند باید با رفتار مناسب پاسخ این اعتماد را بدهد؛ به همین سادگی. یک رهبر خوب، راه را برای تیمش باز می‌کند، به امنیت و آسایش اعضای تیم اهمیت می‌دهد و دائماً تلاش می‌کند که نیازهای آن‌ها را برطرف کند. اگر یک نکته را از این فصل برای همیشه به خاطر می‌سپارید، بگذارید این باشد:

مدیریان سنتی نگران این موضوع هستند که کارها چطور انجام می‌شوند. رهبران واقعی به اینکه چه کارهایی انجام می‌شوند، توجه می‌کنند... (و به تیم خود اعتماد دارند که راه مناسب را برای انجام دادن کار پیدا می‌کنند).



۱. اگر فرزندی داشته باشید، به خوبی با این لحظه آشنا هستید: بعد از آنکه چیزی به فرزندتان گفتید، با خود فکر می‌کنید: «ای وای منم

دقیقاً مثل مادرم شد.»

چند سال پیش، فیتز یک عضو جدید به نام جری در تیمش استخدام کرد. جری در شرکتی که قبلاً در آن کار می‌کرد، مدیری داشت که معتقد بود همه باید از رأس ساعت ۹ صبح تا ۵ عصر پشت میزشان باشند و اگر نه فرض بر این بود که آن‌ها به اندازه کافی کار نمی‌کنند (البته که فرض مسخره‌ای بود). روز اول کار، جری ساعت ۴:۴۰ نزد فیتز آمد و با شرمندگی خواهش کرد که اجازه دهد ۱۵ دقیقه زودتر محل کار را ترک کند تا به قراری که نتوانسته ساعت آن را تغییر دهد برسد. فیتز لبخندی زد و گفت: «ببین، تا زمانی که کارهایت را انجام بدهی، هیچ اهمیتی برایم ندارد که چه ساعتی کار را ترک می‌کنی.» جری لحظه‌ای درنگ کرد و با تعجب به فیتز نگاه کرد. سپس راهش را گرفت و رفت. فیتز با جری مثل یک انسان بزرگسال رفتار کرد. جری همیشه کارهایش را به موقع انجام داد و دیگر لازم نبود فیتز نگران این باشد که آیا جری مشغول به کار است یا خیر. جری نیاز به یک پرستار بچه نداشت.

رهبر یک تیم بودن، لزوماً به این معنی نیست که شما مسئولیت نهایی همه چیز را به عهده دارید. رهبری، انواع مختلفی دارد. ممکن است فنی باشد یا شخصی. در دنیای مهندسی و توسعه نرم افزار، دو عنوان مجزا و مشخص برای رهبران تیم‌ها وجود دارد: یکی رهبر فنی (Tech Lead) یا به اختصار TL، و دیگری مدیر رهبری فنی (Tech Lead Manager) یا به اختصار TLM. یک TL مسئول مسیر و جهت جزئیات کارهای فنی یک بخش یا تمام یک پروژه است. در حالی که TLM علاوه بر رهبری فنی پروژه، مسئولیت مدیریت منابع انسانی در تیم را نیز به عهده دارد. این تقسیم‌بندی این امکان را به افراد می‌دهد که اگر به رهبری افراد و اشخاص تیم علاقه ندارند، بتوانند رهبری فنی پروژه را به عهده بگیرند.

تنها عنصر ترسناک در مدیریت... امم، همه چیز مدیریته!

غیر از بار منفی کلمه «مدیر»، بیشتر آدم‌ها دلایل مختلف دیگری برای عدم علاقه به مدیریت دارند. زمان کمتر برای کدنویسی، مکررترین دلیلی است که مهندسين برای مدیر نشدن، چه مدیر فنی و چه رهبر تیمی می‌آورند. کمی جلوتر در مورد این موضوع صحبت خواهیم کرد؛ ولی ابتدا درباره چند دلیل دیگر که بعضی از ما در راستای دوری از مدیریت داریم صحبت می‌کنیم.

اگر بیشتر دوران زندگی کاری خودتان را به نوشتن کد و برنامه‌نویسی گذرانده باشید، احتمالاً عادت دارید که روزهایتان را با تمام کردن چیزی که بتوان آن را دید و به آن اشاره کرد، به انتها رسانده باشید. آن چیز می‌تواند یک قطعه کد یا یک طرح نوشتاری نرم‌افزار باشد یا تعدادی ایراد نرم‌افزاری که همه را حل کرده باشید. می‌توانید به آن اشاره کنید و بگویید: «این، کاری‌ست که امروز تمام کرده‌ام.» اگر معیار کارایی را بر مبنای موارد تمام‌شده قابل اشاره فرض کنیم، مدیران در پایان روز چون چیزی برای اشاره و نشان دادن ندارند، احتمالاً با خود می‌گویند: «ای بابا، امروز هم که من کاری نکردم.» فرض کنید چند سال کار شما هر روز چیدن سیب از درخت بوده باشد و یک روز کار شما چیدن موز شود. وقتی کارتان را به چیدن موز تغییر دهید، در انتهای روز نمی‌توانید بدون توجه به موزهایی که چیده‌اید، بگویید: «ای بابا، امروز من هیچ سیبی نچیده‌ام.» کمیت دستاوردهای کار مدیریت را به راحتی کار برنامه‌نویسی نمی‌توان مشخص کرد. به عنوان رهبر تیم، لزومی ندارد که موفقیت‌های تیمتان را به اسم خودتان تمام کنید. موفقیت رهبر تیم، در حس رضایت و بهره‌وری اعضای تیم تعریف می‌شود. وقتی کار شما چیدن موز است، موفقیت خودتان را در تعداد سیب‌هایی که چیده‌اید، تعریف نکنید.



یک دلیل بزرگ دیگر برای مدیر نشدن، ریشه در «اصل پیتر»^۱ دارد. این اصل می‌گوید در سلسله‌مراتب اداری، کارمندان تا زمانی ارتقا می‌یابند که وارد نقشی شوند که در آن به مرز بی‌کفایتی می‌رسند. بیشتر آدم‌ها در زندگی کاری‌شان مدیری داشته‌اند که یا در ایفای نقشش ناتوان بوده یا به طور کلی مدیر بد و بی‌کفایتی بوده است.^۲ ما حتی آدم‌هایی را می‌شناسیم که همه مدیران زندگی‌شان، آدم‌های بی‌صلاحیتی بوده‌اند. وقتی مدیران اطراف شما همگی بی‌کفایت و بی‌عرضه بوده باشند، آیا هیچ علاقه‌ای به مدیریت پیدا خواهید کرد؟ اصولاً چرا یک نفر دوست دارد به سمتی ارتقا یابد که در آن هیچ صلاحیتی نداشته باشد؟

در آن سوی سکه، دلایل زیادی برای مدیر شدن نیز وجود دارد. اولاً مدیریت به شما کمک می‌کند که دامنه اثرگذاری خودتان را گسترده‌تر کنید. حتی اگر یک برنامه‌نویس خارق‌العاده باشید، با مدیریت تیمی از برنامه‌نویس‌ها، شما در انتشار مجموعه کدهای بیشتر و وسیع‌تری سهیم خواهید بود. ثانیاً ممکن است شما واقعاً توانایی مدیریت بالایی داشته باشید. خیلی از افرادی که به واسطه عدم وجود رهبر در تیم، به ناچار خودشان را درگیر مسئولیت‌های رهبری و مدیریتی تیم می‌یابند، متوجه می‌شوند که اتفاقاً استعداد خوبی در این زمینه دارند.

1. Peter Principal

۲. و این موضوع دلیل دیگری است برای اینکه شرکت‌ها نباید مسیر ارتقای شغلی را طوری تعریف کنند که مهندسين به سمت مدیریت سوق داده شوند. اگر مهندس نرم‌افزاری در کدنویسی قابلیت و استعداد بالایی داشته باشد و هیچ علاقه‌ای در راستای مدیریت نداشته باشد، با تشویق او به احتساب پست مدیریتی برای ارتقای شغلی، یک برنامه‌نویس عالی را با یک مدیر بی‌کفایت جایگزین می‌کنید.

رهبر خدمتگزار

به نظر می‌رسد که نوعی بیماری خطرناک به مدیران جدید سرایت می‌کند که فراموش می‌کنند مدیران گذشته خودشان چه بدرفتاری‌هایی با آن‌ها داشته‌اند. مدیران جدید از روی ناآگاهی اشتباهات مدیران سابق خودشان را تکرار می‌کنند. این اشتباهات شامل موارد پیش‌رو است، ولی به آن‌ها محدود نمی‌شود: مدیریت ذره‌بینی و کنترل‌گرانه، غفلت از کم‌کاری بعضی افراد و استخدام افراد بله‌قربان‌گو. اگر این بیماری در همان ابتدا و به سرعت درمان نشود، می‌تواند کل تیم را به کشتن دهد.

وقتی که برای اولین بار به پست مدیریت ارتقا پیدا کردیم، بهترین توصیه را از استیو وینتر یکی از مدیران بلندپایه بخش مهندسی در گوگل دریافت کردیم: «مهم‌ترین نکته در مدیریت این است که باید در مقابل وسوسه کنترل کردن ایستاد.» یکی از وسوسه‌های هر مدیر جدیدی این است که دائماً اعضای تیمش را سرپرستی و «مدیریت» کند. به هر حال مدیریت همین است دیگر، مگر نه؟ نتیجه این کار معمولاً مصیبت‌بار خواهد بود.

درمان این بیماری در مدیریت، استعمال مقدار قابل توجهی از دارویی است به نام «خدمتگزاری در مدیریت.» این دارو نام مؤدبانه‌ای است برای اینکه بگوییم مهم‌ترین کاری که یک مدیر می‌تواند انجام دهد این است که مثل یک پیشکار، در خدمت سلامت روحی و جسمی اهل خانه باشد. شما به عنوان یک رهبر خدمتگزار، باید محیطی بر پایه تواضع، احترام و اعتماد - اصول HRT در تیمتان - ایجاد کنید. این کار به این معنی است که موانع کاغذبازی‌های اداری را در مقابل یک عضو تیم بردارید، کمک کنید تا تیم شما به توافق جمعی برسد یا حتی وقتی اعضای تیم شما مجبورند تا دیروقت کار کنند برایشان شام تهیه کنید. یک مدیر خدمتگزار، چاله‌های سر راه تیم را پر می‌کند تا مسیر برای مهندسین هموار شود و از اینکه دست خود را به کار آلوده کند هراسی ندارد. تنها «مدیریتی» که یک رهبر خدمتگزار انجام می‌دهد، مدیریت سلامتی فنی پروژه و فردی اعضای تیم است. هر چند که مدیریت فنی ممکن است ساده‌تر و در نتیجه وسوسه‌انگیزتر به نظر برسد، مدیریت سلامت روحی و فردی اعضای تیم نیز به همان اندازه اهمیت دارد، هر چند که معمولاً بی‌نهایت سخت‌تر است.

الگوهای اشتباه در مدیریت

قبل از اینکه خصوصیات رهبران موفق را تحلیل کنیم، لازم است مجموعه الگوهایی را که نباید در مدیریت از آن‌ها پیروی کرد لیست کنیم. ما این الگوهای مخرب را در تعدادی از مدیران بی‌کفایتی که در طول زندگی کاری‌مان با آن‌ها ارتباط داشته‌ایم، شناسایی کرده‌ایم. اعتراف می‌کنیم که خود ما نیز در مورد عمل به بعضی از آن‌ها اشتباه رفتار کرده‌ایم.

الگوی اشتباه: استخدام افراد بله‌قربان‌گو

اگر شما به عنوان یک مدیر، به هر دلیلی اعتماد به نفس کافی در نقش خودتان را نداشته باشید، یکی از راه‌هایی که می‌تواند به شما کمک کند تا کسی اختیار و قدرت شما را زیر سؤال نبرد این است که آدم‌هایی را در تیمتان استخدام کنید که همیشه از شما اطاعت امر کنند. به راحتی می‌توانید آدم‌هایی را استخدام کنید که یا به اندازه شما باهوش و یا جاه‌طلب نیستند یا حتی اعتماد به نفس کمتری نسبت به شما دارند. این کار جایگاه شما را به عنوان رئیس و تصمیم‌گیرنده اصلی در تیم مستحکم می‌کند. ولی به این معنی نیز خواهد بود که کار و مسئولیت شما بی‌نهایت بیشتر می‌شود. اعضای تیم شما نمی‌توانند حرکتی انجام بدهند مگر آنکه شما مثل سگی که افسارش را به دست گرفته باشید آن‌ها را حرکت دهند. اگر شما گروهی از افراد بله‌قربان‌گو را دور خود جمع کنید، یک روز هم نمی‌توانید به مرخصی بروید. به محض اینکه از اتاق بیرون بروید تیم شما فلج می‌شود؛ ولی خوب این هزینه خیلی کمی است برای اینکه شما بتوانید در جایگاهتان احساس قدرت کنید، نه؟

برعکس شما باید همیشه تلاش کنید تا افرادی را استخدام کنید که از شما باهوش‌تر هستند و بتوانند جایگزین شما باشند. کار سختی است چون چنین افرادی دائماً شما را به چالش می‌کشاند و اشتباهات و اشکالات شما را به شما گوشزد می‌کنند. ولی همین نوع اعضای تیم هستند که شما را با کارهای خارق‌العاده شگفت‌زده می‌کنند. آن‌ها می‌توانند خودشان را به سمت موفقیت هدایت کنند و گاهی بعضی از آن‌ها می‌توانند به رهبری تیم نیز کمک کنند. شما نباید با این دید به قضیه نگاه کنید که هدف چنین افرادی غارت جایگاه شماست، بلکه چنین افرادی به شما این فرصت را می‌دهند که بتوانید حتی تیم‌های بیشتری را تحت رهبری خودتان دریاورید، پروژه‌های هیجان‌انگیزتری را قبول کنید و یا حتی چند روز بدون دغدغه و نگرانی به مرخصی و سفر بروید.

الگوی اشتباه: غفلت از کم کاری

فیتز، آن اوایل وقتی که به تازگی رهبر یک تیم در گوگل شده بود، اولین بار وظیفه تحویل نامه‌های پاداش اعضای تیمش را به عهده گرفته بود. برای این کار بسیار هیجان‌زده بود و به مدیر خودش گفت: «من عاشق مدیریت هستم!» مدیر فیتز بی‌درنگ به او پاسخ داد: «گاهی اوقات فرشته دندان خواهی بود و گاهی اوقات دندان‌پزشک!»

کشیدن دندان هیچ‌وقت لذت‌بخش نیست! رهبران زیادی را دیده‌ایم که با وجود تمام کارها و تصمیم‌های درست در مورد تیمشان، فقط به دلیل کم کاری صرفاً یک یا دو نفر از اعضای تیم، به موفقیت نرسیده‌اند. مدیریت منابع انسانی، سخت‌ترین کار در رهبری یک تیم است و سخت‌ترین کار در بُعد مدیریت منابع انسانی، برخورد با آن دسته از اعضای تیم است که انتظارات را برآورده نمی‌کنند. گاهی علت کم کاری افراد می‌تواند این باشد که به اندازه کافی برای وظایفشان وقت نمی‌گذارند؛ ولی دشوارترین حالت زمانی رخ می‌دهد که با فردی مواجه می‌شوید که هر چقدر تلاش و کوشش می‌کند و روی کار خود وقت می‌گذارد، باز هم نمی‌تواند مسئولیت‌های خود را به خوبی انجام دهد.

در گوگل تیمی وجود دارد که مسئولیت در دسترس بودن تمام سرویس‌ها را به عهده دارد. شعار تیم این است: «امید، سیاست ما نیست!» هیچ کجا غیر از وقتی که با انسان‌های کم کار برخورد می‌کنید، به این میزان از «امید» به عنوان یک استراتژی سوءاستفاده نشده است. بیشتر رهبران و مدیران در مواجهه با آن دسته از اعضای تیم که انتظارات را برآورده نمی‌کنند، نگاهشان را برمی‌گردانند و فقط به اینکه فرد کم کار، معجزه‌وار انگیزه و توانایی کار را به دست بیاورد یا خودش با پای خودش از تیم برود، چشم امید می‌دوزند. طبیعی است که این سیاست، خیلی به ندرت جوابگو است.

هم‌زمان با امید واهی مدیر تیم در راستای بهبود وضعیت شخص کم کار، سایر افراد تیم مجبورند وقت گران‌بهای خود را صرف جبران اشتباهات و کم کاری‌های او کنند. این کار علاوه بر اتلاف وقت، باعث کاهش روحیه تیمی نیز می‌شود. اینکه شما چشمتان را روی کم کاری یک فرد می‌بندید، به این معنی نیست که سایرین نیز او را نمی‌بینند. اتفاقاً مطمئن باشید که سایر اعضای تیم شما از حضور فرد کم کار به خوبی آگاه هستند؛ چرا که آن‌ها دارند جور او را می‌کشند.

بی‌اعتنایی به کم کاری در تیم مانع جذب نیروهای کارآمد و مفید نیز می‌شود و به سایر اعضای تیم نیز این پیام را می‌دهد که اگر خیلی هم در وظایف خود تلاش و تمرکز نکنید، اهمیت زیادی برای شما ندارد. بدین ترتیب شما افراد سخت‌کوش را هم از دست داده و به مرور یک تیم متشکل از افراد تنبل و بی‌مسئولیت خواهید داشت. شما با نگهداری افراد کم کار در تیمتان هیچ لطفی در حق آن‌ها نمی‌کنید. معمولاً کسی که مسئولیت‌های تیم

شما را نمی‌تواند به خوبی انجام دهد، می‌تواند تیم دیگری را پیدا کند که در آنجا تأثیرگذاری بیشتر و مفیدتری داشته باشد.

بهتر است که در اسرع وقت به شخص کم‌کار در تیمتان رسیدگی کنید تا فرصت کمک به او را داشته باشید. کمی تشویق و دلگرمی یا راهنمایی برای نشان دادن مسیر می‌تواند خیلی زود مشکل کم‌کاری یک نفر را حل کند. اما اگر برای رسیدگی به این کار درنگ کنید، به مرور رابطه شخص کم‌کار با سایر افراد تیم شکرآب شده و کار برای حل و فصل مسئله و کمک به او دشوارتر می‌شود.

چطور می‌توان یک فرد با بهره‌وری پایین را هدایت کرد؟ متأسفانه ما هر دو در این راستا تجربه‌های متنوعی کسب کرده‌ایم و درس‌های زیادی از طریق آزمون و خطاها آموخته‌ایم که گاهی خیلی دردناک بوده‌اند. نزدیک‌ترین قیاسی که می‌توان مطرح کرد این است که بخواهیم به انسانی که لنگان‌لنگان راه می‌رود، ابتدا راه رفتن، سپس قدم زدن‌های طولانی‌تر و نهایتاً دویدن را بیاموزیم. این کار معمولاً به مدیریت و کنترل کوچک‌ترین کارهای آن شخص، هر چند به صورت موقت نیاز دارد. در این مسیر تمام اصول HRT مخصوصاً احترام، اهمیت بسیار بالایی دارند.

یک بازه زمانی محدود ولی کافی را مشخص کنید، مثلاً دو یا سه ماه. انتظاراتی را که از شخص مورد نظر در این مدت دارید دقیق تعریف کنید. اهداف کوچک و پله‌پله‌ای تعیین کنید تا پیروزی‌های کوچک و قابل اندازه‌گیری برای او میسر شوند. هر هفته با او دیدار کنید تا پیشرفت کارش را پیگیری کنید و برایش اهداف مرحله‌بندی را به وضوح تعریف کنید. بدین شکل اندازه‌گیری پیشرفت یا پسرفت راحت‌تر می‌شود. اگر فرد کم‌کار نتواند پابه‌پای اهداف شما پیش بیاید، همان ابتدای کار به وضوح مشخص می‌شود. اینجاست که معمولاً خود فرد متوجه می‌شود که نمی‌تواند در این جایگاه فرد تأثیرگذار و مفیدی باشد و از تیم بیرون می‌رود. بعضی اوقات نیز فرد کم‌کار به خود آمده و با تلاش و پشتکار بیشتر کاستی‌ها را جبران می‌کند و به یک فرد کارآمد در تیم شما تبدیل می‌شود. هر نتیجه‌ای که حاصل شود، دخالت شما در این امر باعث سرعت‌بخشی به انجام تغییرات مورد نیاز می‌شود.

الگوی اشتباه: غفلت از مشکلات انسانی

همان طور که پیش‌تر اشاره کردیم، رهبری یک تیم دارای یک بعد فنی و یک بعد انسانی است. از آنجا که بیشتر رهبران در تیم‌های مهندسی نرم‌افزار قبلاً مسئولیت‌های فنی داشته‌اند، به راحتی بُعد انسانی مدیریت را فراموش می‌کنند. تمرکز روی مشکلات و اهداف فنی یک تیم برای کسی که قبلاً خودش در توسعه فنی پروژه‌ها شریک بوده است ساده‌تر است. حتی در زمان دانشجویی نیز تمام آموزش‌های شما حول محور فنی و مهندسی می‌چرخید. اکنون که مدیر یک تیم شده‌اید دست و پنجه نرم کردن با مدیریت منابع انسانی تیم شما دشوار شده است.

چندین سال پیش نخستین فرزند یکی از دوستان نزدیک فیتز (اسم این دوست را بگذاریم جیک) به دنیا آمد. جیک و فیتز سال‌های زیادی هم به صورت دورکاری و هم در یک دفتر کار مشترک با یکدیگر کار کرده بودند. جیک هفته‌های ابتدایی پس از تولد فرزندش را از راه دور در خانه کار کرد تا برای کمک به همسرش در دسترس باشد. فیتز که تجربه کار از راه دور با جیک را داشت، هیچ ممانعتی با این قضیه نکرد. هر دو به خوبی مشغول کار مشترک بودند تا وقتی که مدیر آن‌ها، پابلو، باخبر شد که جیک بیشتر روزهای هفته را از داخل خانه کار می‌کند. با وجود آنکه جیک و فیتز هیچ مشکل ارتباطی با یکدیگر نداشتند و کارشان را همیشه به خوبی ادامه می‌دادند، پابلو از دورکاری جیک ناراضی بود. هرچه جیک برای پابلو توضیح می‌داد که عملکرد و کارایی او با کار از منزل تغییری نکرده است و به این طریق می‌تواند حامی همسرش هم باشد و در نگهداری از نوزادشان به او کمک کند، هیچ فایده‌ای نداشت. پاسخ پابلو این بود: «بابا همه مردم بچه‌دار می‌شوند! کار باید در دفتر و به صورت حضوری انجام شود.» لازم نیست بگوییم که جیک (که معمولاً انسان آرام و خوش‌برخوردی بود) چقدر از این برخورد ناراحت شد و احترامش را نسبت به مدیرش از دست داد.

پابلو راه‌های بهتری برای برخورد با این موضوع داشت. می‌توانست وضعیت شخصی جیک را درک کند و بپذیرد که او نیاز دارد تا در کنار همسرش باشد. وقتی عملکرد تیمش تحت تأثیر قرار نگرفته است، چرا باید روی حضور فیزیکی جیک در دفتر کار اصرار کند؟ او حتی می‌توانست با جیک این‌طور مذاکره کند که او یک روز در هفته را برای حضور در دفتر کار اختصاص دهد تا نوزاد او کمی بزرگ‌تر شده و کارها برای او و خانواده‌اش راحت‌تر شود. اندکی همدلی از جانب پابلو می‌توانست کمک کند تا جیک احساس رضایت بیشتری از کار در تیم داشته باشد.

الگوی اشتباه: دوستی صمیمانه با همه

وقتی شما به سمت مدیریت تیم خودتان ارتقا می‌یابید، با این چالش مواجه می‌شوید که چطور دوستی خودتان را با همکارانتان که حالا شما رهبرشان هستید حفظ کنید. بعضی از رهبران با ترس از بین رفتن رفاقت‌های قدیمی‌شان مجبور می‌شوند که تلاش و کوشش مضاعفی در امر سرپرستی تیم داشته باشند. این کار برعکس ممکن است به یک فاجعه و از دست رفتن دوستی‌ها منجر شود. دوستی را با مدیریت ساده‌لوحانه اشتباه نگیرید. وقتی مسئولیت زندگی کاری یک نفر با شماست، ممکن است که او به رفتارهای دوستانه شما به طور تصنعی و از روی اجبار پاسخ دهد.

یادتان باشد که شما می‌توانید رهبر موفق یک تیم باشید بدون آنکه با تک‌تک اعضا رفاقت صمیمانه‌ای داشته باشید. همچنین می‌توانید در رهبری تیمتان اقتدار داشته باشید بدون آنکه نیاز باشد تا دوستی‌های قدیمی‌تان را به خطر بیندازید. یکی از کارهایی که ما برای برقراری ارتباط صمیمی ولی حرفه‌ای با اعضای تیم پیدا کرده‌ایم، صرف ناهار به صورت تیمی است. این کار به شما کمک می‌کند که با اعضای تیمتان در یک محیط غیررسمی‌تر در مورد موضوعات غیر کاری صحبت کنید.

گاهی ارتقا به مدیریت روی تیمی که یکی از اعضای آن از دوستان صمیمی و نزدیک شما باشد، محیط بسیار دشوار و پر چالشی را ایجاد می‌کند. مخصوصاً اگر دوست صمیمی شما خیلی عضو کارآمدی نباشد و دائماً نیاز به کنترل و مدیریت داشته باشد. توصیه ما این است که از چنین موقعیت‌هایی تا جایی که می‌توانید دوری کنید.

الگوی اشتباه: معیارهای پایین برای استخدام

استیو جابز معتقد بود: «آدم‌های درجه یک، آدم‌های درجه یک و آدم‌های درجه دو، آدم‌های درجه سه استخدام می‌کنند.» به راحتی می‌توان قربانی این اشتباه شد، مخصوصاً وقتی که برای استخدام افراد جدید عجله داشته باشید. روند متداول استخدام آن‌طور که ما مشاهده کرده‌ایم، این‌گونه است که وقتی تیم‌ها نیاز دارند تا مثلاً پنج نفر نیروی جدید جذب کنند، با چهل تا پنجاه نفر متقاضی مصاحبه می‌کنند و پنج نفر از بهترین‌ها را استخدام می‌کنند. دیگر اهمیتی ندارد که این پنج نفر، تمام معیارها و استانداردهای شما برای استخدام را دارند یا نه. این روش سریع‌ترین راه برای درست کردن یک تیم معمولی و متوسط است.

هزینه پیدا کردن شخص مناسب، از هزینه‌های تبلیغات تا حقوق افراد منابع انسانی، در مقابل هزینه سروکله زدن با شخص اشتباهی که هیچ‌وقت نباید استخدام می‌شد، بسیار ناچیز است. هزینه این نوع سروکله زدن در از دست رفتن کارایی تیم و زمانی که در این راستا تلف می‌شود، نهفته است و با این فرض آن را می‌پذیریم تا از آسیب‌هایی که جدایی شخص اشتباه به تیم می‌زند صرف‌نظر کنیم. اگر اختیار عمل در نحوه استخدام یا اخراج اعضای تیم با شما نباشد، باید برای جذب بهترین مهندسين با چنگ و دندان مبارزه کنید. اگر اعضای تیمی که به شما داده می‌شود علی‌رغم تمام تلاش‌های شما، از کیفیت بالایی برخوردار نباشند، شاید بهتر باشد که به فکر پیدا کردن جای دیگری برای کار باشید. بدون مواد خام اولیه برای تشکیل یک تیم خوب، شما محکوم به شکست خواهید بود.

الگوی اشتباه: با اعضای تیم مثل بچه‌ها رفتار کنید

رفتار بچه‌گانه با اعضای تیم، بهترین راه برای نشان دادن عدم اعتماد شما به آنهاست. مردم همان طوری با شما رفتار می‌کنند که شما با آنها رفتار می‌کنید. اگر نحوه برخورد شما با اعضای تیمتان مانند بچه‌ها یا مثل یک زندانی باشد، از باز خورد آنها شگفت‌زده نشوید. رفتار بچه‌گانه با اعضای تیم در عمل این‌طور است که تک‌تک کارها و تصمیم‌های آنها را تحت کنترل خود داشته باشید و در به عهده گرفتن هیچ مسئولیتی به آنها اعتماد نکنید. بدین صورت تیم خود را به سمت شکست رهبری می‌کنید. البته می‌توان گفت اگر هدف شما پرستاری از کودکان باشد، در رسیدن به این هدف موفق خواهید شد. اگر شما افراد متعهد و قابل اعتمادی را استخدام کرده باشید و مسئولیت‌های مختلف را با اطمینان به عهده آنها بگذارید، آنها نیز از عهده کار به خوبی بر می‌آیند. توجه کنید که این کار منوط به استخدام افراد مسئولیت‌پذیر است، همان‌طور که در بخش قبل توضیح داده شد.

فیتز مسئول برگزاری همایشی در شیکاگوست که قبلاً در مکانی کرایه‌ای انجام می‌شد. اولین باری که او می‌خواست کلید سالن همایش را تحویل بگیرد، مدیر سالن بعد از نشان دادن امکانات سالن و توضیح اینکه هر چیزی در کجا قرار دارد، کلید سالن را به فیتز تحویل داد و رفت. هیچ لیستی از دستورات برای بایدها و نبایدها وجود نداشت و مدیریت سالن به فیتز اعتماد کاملی را نشان داده بود. در نتیجه فیتز و تیمش احساس مسئولیت بیشتری در راستای مراقبت از مکان داشتند و در طول همایش با سالن طوری رفتار کردند که انگار خود صاحب آنجا هستند.

چنین سطح از اعتمادی می‌تواند در قالب یک سالن همایش باشد و یا حتی دفتر و وسایل کار. به عنوان یک مثال دیگر، گوگل به کارمندان خود امکان دسترسی به کابینت‌هایی را می‌دهد که پر از لوازم تحریر نظیر دفتر و خودکار و ... هستند. همه آزادند هر تعدادی که نیاز دارند از این وسایل استفاده کنند. بخش فناوری اطلاعات شرکت نیز ایستگاه‌هایی به اسم Tech Stop دارد که در اصل شبیه یک دستگاه خودپرداز برای قطعات الکترونیکی کوچک است. اعضای شرکت هر زمانی که نیاز داشته باشند می‌توانند وسایل مختلف الکترونیکی از قبیل کابل‌های برق، کیبورد، هدفون و ... را به راحتی بردارند. کارمندان گوگل وقتی این حس اعتماد از طرف شرکت را می‌بینند در قبال نحوه برخورد با این امکانات احساس مسئولیت بیشتری در راستای انجام کار درست دارند. بسیاری از افرادی که در شرکت‌های دیگر کار می‌کنند با شنیدن چنین امکاناتی وحشت‌زده می‌شوند و با خود می‌گویند که حتماً گوگل هزینه زیادی بابت «دزدی»‌هایی که کارمندان می‌کنند می‌پردازد. قطعاً چنین چیزی امکان‌پذیر است، ولی در مقابل، یک شرکت با نیروی کاری که مانند بچه‌ها نیاز به مراقبت داشته باشد، هزینه بیشتری را متقبل نمی‌شود؟ مطمئناً از تعدادی خودکار و رابط USB گران‌تر خواهد شد.

الگوهای صحیح در رهبری تیم

در این بخش الگوهایی را مطرح خواهیم کرد که بر اساس تجربه شخصی خودمان، آنچه در رفتار رهبران موفق دیده‌ایم و همچنین توصیه‌هایی که مربی‌های ما در طول زندگی کاری به ما داشته‌اند، به شما برای رهبری بهتر و موفق‌تر کمک خواهند کرد. الگوهای مطرح شده نه تنها به ما در امر رهبری کمک کرده‌اند، بلکه بهترین رفتارهایی بوده‌اند که ما در سایر رهبرانی که دنبال می‌کنیم دیده‌ایم.

غرور را بشکن

در فصل اول زمانی که اصول HRT را مطرح کردیم، درباره‌ی کنار گذاشتن غرور نیز صحبت کردیم. شکستن غرور به طور خاص وقتی شما می‌خواهید یک رهبر خدمتگزار باشید اهمیت ویژه‌ای دارد. عده‌ای به اشتباه این توصیه را این‌طور قلمداد می‌کنند که رهبران باید با هر ساز اعضای تیمشان برقصند و اجازه دهند که همه از سروکله‌ی آن‌ها بالا بروند. ولی این صحیح نیست. البته اعتراف می‌کنیم که تواضع و فروتنی مرز بسیار باریکی دارد با اینکه اجازه دهید از شما سوءاستفاده شود. ولی فروتنی به معنی عدم اعتماد به نفس نیست. شما می‌توانید بدون تکبر و خودبزرگ‌بینی با اعتماد به نفس نظرات و عقاید خودتان را داشته باشید. غرور شخصی، مخصوصاً اگر از جانب رهبر تیم باشد، کار را برای تیم سخت می‌کند. در عوض شما باید به دنبال تقویت روحیه‌ی احساس غرور و افتخار تیمی بین اعضای گروه باشید.

بخشی از شکستن غرور و تکبر، معلول مبحثی است که در بخش پیش صحبت کردیم: اعتماد به اعضای تیم. به این معنی که باید به دستاوردها و توانایی‌های اعضای تیم احترام بگذارید، حتی اگر که به تازگی به تیمتان ملحق شده‌اند.

اگر در جزئیات کار و تصمیم‌های افراد تیم دخالت بی‌جا نمی‌کنید، مطمئن باشید که آن‌ها دانش بیشتری درباره‌ی کاری که انجام می‌دهند دارند. اگرچه شما مسئول اجماع کلی تیم و هدایت آن‌ها در مسیر درست هستید، اعضای تیم بیشتر از شما از زیر و بم جزئیات کار اطلاع دارند و بهتر می‌توانند برای رسیدن به اهداف، تصمیم‌گیری کنند. این کار به اعضا، احساس مالکیت بیشتر و نیز احساس مسئولیت بالاتری در راستای موفقیت (یا حتی شکست!) تیم می‌دهد. اگر یک تیم خوب جُفت‌وجور کنید و به آن‌ها اجازه دهید که سطح کیفی کار را مشخص کنند، کارهای بیشتری را به سرانجام خواهید رساند.

بیشتر افرادی که به تازگی به پست رهبری یک تیم رسیده‌اند، احساس می‌کنند که باید سر از همه‌چیز درآورند و برای هر مشکلی پاسخ صحیحی داشته باشند. به شما اطمینان می‌دهیم که هیچ‌وقت جواب همه‌چیز را پیدا نخواهید کرد و نمی‌توانید همه‌ی مشکلات را حل کنید. اگر وانمود کنید جواب همه‌ی سؤال‌ها را می‌دانید، خیلی زود

احترام اعضای تیمتان را از دست خواهید داد. بخش عمده‌ای از این رفتار به میزان اعتماد به نفس شما باز می‌گردد. از اینکه در مورد عملکرد شما سؤال شود، استقبال کنید. یادتان باشد که وقتی یک نفر از شما دربارهٔ تصمیم یا صحبتی که بیان کردید سؤال می‌کند، معمولاً به دنبال این است که شما را بهتر بشناسد. اگر از به چالش کشیده شدن استقبال کنید، راحت‌تر و سریع‌تر انتقادات سازنده را می‌شنوید و در نتیجه رهبر بهتر و کارآمدتری خواهید شد. پیدا کردن افرادی که از شما صادقانه انتقاد کنند، بسیار دشوار است، مخصوصاً اگر این افراد عضو تیم شما باشند و برای شما کار کنند. به صورت مسئله کلی و هدف نهایی تیم فکر کنید تا راحت‌تر و با ذهن بازتری انتقاد و پیشنهادهای را بپذیرید.

آخرین مرحله از شکستن غرور اگرچه خیلی ساده است اما باید بگوییم که بسیاری از مهندسين حاضرند در روغن داغ سرخ شوند، ولی انجامش ندهند: وقتی اشتباهی مرتکب می‌شوید عذرخواهی کنید. منظورمان این نیست که دائماً در صحبتتان مثل نمک روی پاپ‌کورن، کلمه‌های بی‌معنی «ببخشید... ببخشید» را تکرار کنید. باید از ته دل به عذرخواهی خودتان اعتقاد داشته باشید. شک نکنید که قطعاً روزی در کارتان اشتباه خواهید کرد. صرف‌نظر از اینکه به اشتباه خودتان اعتراف کنید یا خیر، اعضای تیم متوجه اشتباه شما خواهند شد. چه دربارهٔ اشتباهتان صحبت کنید یا نه، اعضای تیم اشتباه شما را می‌بینند (و قطعاً دربارهٔ آن پشت سر شما با همدیگر صحبت خواهند کرد). عذرخواهی هزینه‌ای ندارد. انسان‌ها احترام بسیار زیادی برای رهبری که به خاطر اشتباهش عذرخواهی می‌کند قائل هستند. برخلاف باور معمول، عذرخواهی هیچ‌چیز از شما کم نمی‌کند. برعکس معمولاً احترام بیشتری را بعد از عذرخواهی دریافت می‌کنید؛ چرا که با این کار شما نشان می‌دهید که یک رهبر منطقی و واقع‌بین و مطابق با اصول HRT، متواضع و فروتن هستید.

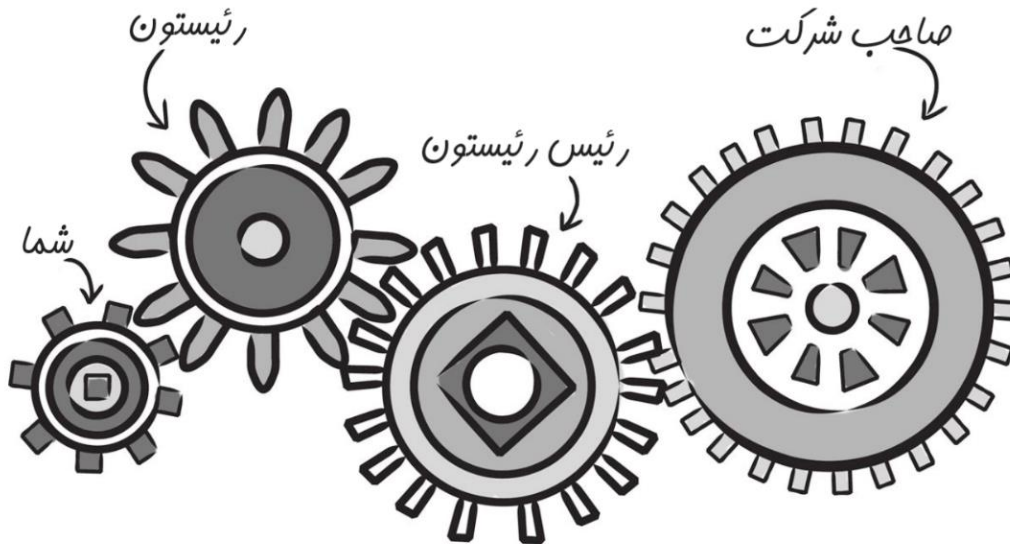
آرام و متین باشید

معمولاً به عنوان یک مهندس، حس بدبینی و شکاکی در شما تقویت می‌شود. ولی وقتی در مقام رهبر یک تیم قرار می‌گیرید، این شک و تردید برای شما ایجاد دردسر خواهد کرد. منظور ما این نیست که ساده‌لوح و خوش‌خیال باشید. بلکه توصیه ما این است که شک و تردید را در گفتار خود بروز ندهید، ولی در عین حال به تیم خود نشان دهید که در جریان پیچیدگی‌ها و موانع کارتان هستید. متانت در رفتار و آرامش عمل در امر رهبری تیم اهمیت بسیار بالایی دارد؛ چرا که اعضای تیم شما (به طور خودآگاه یا ناخودآگاه) از رفتار شما در مواجهه با اتفاقات اطراف، الگو می‌گیرند.



یک راه ساده برای نشان دادن اثر این قضیه این است که به سلسله‌مراتب سازمانی شرکت به چشم تعدادی چرخ و دنده وصل شده به هم نگاه کنید. اعضای تیم چرخ‌های کوچکی هستند که دنده‌های آن به رهبر تیم متصل است. رهبر تیم به نسبت چرخ بزرگ‌تری است که خود به مدیر میانی وصل است. نهایتاً بزرگ‌ترین چرخ متعلق به مدیرعامل شرکت است. به این معنی که وقتی رهبر تیم یک دور می‌چرخد، اعضای تیم که چرخ‌های کوچک‌تری

هستند دو یا سه دور خواهند زد. و به همین ترتیب یک حرکت کوچک از مدیرعامل شرکت ممکن است باعث شود که اعضای تیم‌ها هر کدام شش یا هفت دور تند بزنند. هرچه که شما در سلسله‌مراتب شرکت بالاتر می‌روید، حرکات شما باعث می‌شود که زیردستان شما دورهای سریع‌تری بزنند.



به این موضوع با یک دید دیگر نیز می‌توان نگاه کرد: رهبر تیم همیشه در معرض دید همه است. تمام حرکات و رفتارهای شما همیشه تحت نظر اعضای تیم خواهد بود. نه تنها وقتی که در حال صحبت کردن یا در یک جلسه هستید، بلکه حتی زمانی که پشت میز خود به تنهایی نشستید و مشغول جواب دادن به ایمیل‌هایتان هستید. تیم شما دائماً دنبال نشانه‌هایی در نحوه رفتار، کلام و زبان بدن شما در موقعیت‌های گوناگون، مثل واکنش شما به یک حرف یا حتی رفتار شما هنگام صرف ناهار هستند. آیا در رفتارتان اعتمادبه‌نفس می‌بینند یا ترس و نگرانی؟ وظیفه شما به عنوان رهبر، الهام‌بخشی به اعضای تیمتان است. ولی این کار را باید در ۲۴ ساعت روز و ۷ روز هفته انجام دهید. طرز برخورد شما در موقعیت‌های مختلف حتی در وضعیت‌های بی‌اهمیت، به طور ناخودآگاه روی تیم شما تأثیر می‌گذارد.

فیتز، مدیری به نام بیل^۱ داشت که حقیقتاً استاد حفظ آرامش و خونسردی در تمام حالات بود. بیل صرف‌نظر از مشکلات و اتفاق‌های ناگوار و غیرمنتظره، هیچ‌وقت اضطراب و نگرانی‌ای بروز نمی‌داد. هنگام یک مشکل معمولاً یک دستش را روی سینه می‌گذاشت و با دست دیگرش چانه‌اش را می‌گرفت و از اعضای تیم و مهندسی که مضطرب و نگران بودند به آرامی سؤالاتش را می‌پرسید. این کار او باعث آرامش بیشتر و کاهش اضطراب تیم می‌شد تا آن‌ها مثل یک مرغ پرنده دور اتاق ندوند. فیتز به شوخی درباره او می‌گفت که اگر روزی به بیل بگویند

۱. بیل نام واقعی اوست.

موجودات فرازمینی نوزده عدد از دفاتر شرکت را نابود کرده‌اند، او حتماً با خونسردی خواهد پرسید: «می‌دانید چرا به دفتر بیستم نرسیدند؟»

این مثال ما را به ترفند دیگری در حفظ آرامش می‌رساند: «سؤال پرسیدن.» وقتی کسی برای مشورت نزد شما می‌آید، معمولاً برای ارائه پیشنهاد و راه‌حل بسیار هیجان‌زده می‌شوید، چرا که بالاخره این فرصت را پیدا کرده‌اید که مشکلی را حل کنید. به هر حال تمام زندگی کاری شما قبل از رسیدن به مدیریت، حل مشکلات مختلف مهندسی بوده است. بنابراین خیلی سریع وارد حالت حل‌کننده مشکلات می‌شوید. اما این کار صحیحی نیست. کسی که برای مشورت پیش شما می‌آید، اصولاً به دنبال این نیست که شما مشکلتان را حل کنید، بلکه از شما می‌خواهد که کمکش کنید تا خودش مشکلتان را حل کند و بهترین روش برای این کار این است که از او سؤال بپرسید. با رعایت اصول HRT به او کمک کنید تا بتواند مسئله‌اش را تحلیل کند، از زوایای مختلف به آن نگاه کند و خودش به راه‌حلی که برایش بهتر است برسد.^۱ اعضای تیم با این روش معمولاً به جواب می‌رسند. و این جواب متعلق به خودشان خواهد بود که نشان می‌دهد این روش مطابق با همان اصول اعتماد به اعضای تیم برای قبول مسئولیت است که پیش‌تر درباره‌اش صحبت کردیم. صرف‌نظر از اینکه شما برای مشکل افراد، راه‌حلی داشته باشید یا نه، با این روش معمولاً آن‌ها به یک راه‌حل می‌رسند و احساس می‌کنند که شما از اول راه‌حل را می‌دانستید. مکارانه به نظر می‌رسد، نه؟ سقراط به شما افتخار خواهد کرد.

۱. مثل تکنیک Rubber duck debugging در برنامه‌نویسی که در آن برنامه‌نویس مشکل کدش را برای یک اردک پلاستیکی توضیح می‌دهد تا در روال توضیح دادن، مسئله را بهتر درک کرده و به راه‌حل برسد.

کاتالیزور باشید

کاتالیزور در علم شیمی، ماده‌ای است که سرعت یک واکنش را بیشتر می‌کند، ولی خود در واکنش شرکت نمی‌کند. یکی از روش‌های کاتالیزورها (مثل آنزیم‌ها) این است که عناصر شرکت‌کننده در واکنش را به همدیگر نزدیک می‌کنند تا در محیط یک محلول پخش و دور از هم نباشند. وقتی که کاتالیزور عناصر را گرد هم می‌آورد، احتمال رخداد واکنش‌های شیمیایی افزایش پیدا می‌کند. این همان نقشی است که شما باید به عنوان رهبر برای اعضای تیمتان ایفا کنید. راه‌های زیادی برای انجام این کار وجود دارند.

یکی از وظایف رایج هر رهبری این است که کمک کند تا اعضای تیم بر سر موضوعات مختلف به توافق برسند. برای این کار گاهی لازم است که شما کنترل بحث‌ها را از ابتدا تا انتها به دست بگیرید و گاهی لازم است که فقط بحث را کمی در جهت مناسب هدایت کنید. مدیرانی که خود را در جایگاه رهبری یک تیم می‌بینند ولی به طور رسمی وظیفه رهبری را به عهده نگرفته‌اند، معمولاً از این مهارت خیلی استفاده می‌کنند؛ چرا که با این روش آن‌ها بدون آنکه مستقیماً اختیار عمل را به دست بگیرند می‌توانند جهت‌گیری تیم را رهبری کنند. اگر اختیار عمل به طور رسمی با شما باشد، راحت‌تر می‌توانید مسیر بهتر را به تیمتان دیکته کنید. ولی این‌طور امر و نهی کردن نمی‌تواند به اندازه رسیدن به یک اجماع تیمی مفید و مؤثر باشد. اگر اعضای تیم شما به دنبال این باشند که با سرعت بیشتری پیش بروند، ممکن است داوطلبانه اختیاری بعضی از تصمیم‌گیری‌ها را به عهده رهبر فنی تیم بگذارند. اگرچه ممکن است ظاهراً دیکتاتوری به نظر بیاید، ولی اگر داوطلبانه باشد این هم خود نوعی توافق و اجماع تیمی است.

محل نشانه‌گذاری با گچ را بدانید

داستانی درباره یک مکانیک بازنشسته وجود دارد که در کار خود بسیار ماهر و زبردست بود. یکی از شرکت‌های قبلی او دچار مشکلی در یکی از سیستم‌هایشان شده بود که هیچ‌کدام از کارمندان نتوانسته بودند مسئله را حل کنند. بنابراین با استاد مکانیک بازنشسته تماس گرفتند تا شاید او بتواند مشکل را پیدا کند. استاد آمد و دستگاه را به دقت بررسی کرد، به صداهایی که از دستگاه بیرون می‌آمد گوش داد و در نهایت یک تکه گچ از روی زمین برداشت و یک ضربدر را روی قسمتی از دستگاه نشانه‌گذاری کرد. سپس به کادر فنی گفت که دقیقاً در همین محل یک قطعه سیم پاره شده و نیاز به تعمیر دارد. آن‌ها همان ناحیه از دستگاه را باز کردند، قطعه سیم را تعمیر کردند و مشکل دستگاه را حل کردند. چند وقت بعد مدیرعامل شرکت، رسید هزینه کار را که شامل مبلغ ۱۰,۰۰۰ دلار بود دریافت کرد و با تعجب از استاد خواست که ریز جزئیات هزینه‌ها را لیست کند تا ببیند که چرا یک ضربدر گچی این قدر گران تمام شده است. استاد مکانیک بازنشسته در جواب گفت: ۱ دلار هزینه گچ، ۹,۹۹۹ دلار هزینه اینکه فهمیدم کجا را باید علامت بزنم. این داستان برای ما نشان‌دهنده اهمیت دانش و ذکاوت است. اینکه یک تغییر هر چند کوچک ولی ظریف و با دقت می‌تواند تأثیرات بسیار بزرگی را به همراه داشته باشد. بن، در نحوه مدیریتش از این تکنیک خیلی استفاده می‌کند. او اعضای تیمش را سوار بر یک بالن بسیار بزرگ و در حال پرواز

تصور می‌کند. به جای اینکه تک‌تک حرکات و تصمیم‌گیری‌های افراد را مدیریت کند، بیشتر مدت زمان هفته را با دقت تیمش را زیر نظر می‌گیرد. سپس در پایان هفته با یک تکه گچ یک ضربدر کوچک و دقیق روی نقطه‌ای از بالن می‌گذارد و با یک ضربه آرام ولی حیاتی به آن نقطه، مسیر حرکت بالن را تنظیم می‌کند.

گاهی اوقات تیم شما بر سر آنچه که باید انجام دهند توافق و اجماع کلی دارند، ولی در طول مسیر با موانعی برخورد می‌کنند و گیر می‌افتند. این موانع می‌توانند فنی باشند یا مربوط به یک شرکت و سازمان. یکی دیگر از مهارت‌های معمول در رهبری تیم‌ها این است که بتوانید این موانع را از سر راه تیمتان بردارید. بعضی اوقات برای اعضای تیم شما غیرممکن است که بتوانند دست‌اندازها را بردارند، در حالی که شما به راحتی می‌توانید مشکلشان را حل کنید. تلاش کنید تا تیم شما همیشه بداند که با کمال میل توانایی‌های خودتان را به کار می‌گیرید تا موانع سر راهشان را از بین ببرید.

یک بار اعضای تیم فیتز چندین هفته درگیر سروکله زدن با واحد حقوقی شرکت بودند و وقتی احساس کردند کاملاً به بن‌بست رسیده‌اند، نزد فیتز آمدند. از آنجا که فیتز دقیقاً می‌دانست چه شخصی در شرکت مسئول این کار است، مشکل تیم را در کمتر از دو ساعت حل کرد. در یک مثال دیگر، تیم بن به دنبال مقاداری ظرفیت سخت‌افزاری بودند که از هیچ‌جا نمی‌توانستند آن را تهیه کنند. خوشبختانه بن با یک تیم دیگر در شرکت تماس گرفت و توانست همان بعدازظهر ظرفیت سخت‌افزاری را برای تیمش فراهم کند. بار دیگر یکی از مهندسين فیتز در فهم یک قطعه کد جاوا دچار مشکل شده بود. فیتز اگرچه خود متخصص جاوا نبود، ولی ارتباطی بین عضو تیمش و یک مهندس متخصص جاوا از تیمی دیگر برقرار کرد تا مشکل حل شود. لزومی ندارد که شما جواب همه سؤال‌ها را بدانید. معمولاً فقط کافی‌ست افرادی را بشناسید که در جواب سؤال‌ها بتوانند به شما کمک کنند. بیشتر اوقات شناختن شخص مناسب از دانستن جواب درست باارزش‌تر و مفیدتر است.

شکست، نیز یک گزینه است

یک راه دیگر برای اینکه کاتالیزور تیم باشید این است که فضا را طوری فراهم کنید تا اعضای تیم برای ریسک و خطر کردن احساس امنیت و آرامش کنند. ریسک و خطر بسیار فریبنده هستند. بیشتر آدم‌ها در سنجش میزان خطر بسیار بد عمل می‌کنند و شرکت‌ها نیز معمولاً تمام تلاششان را می‌کنند تا آنجا که ممکن است از خطر دوری کنند. در نتیجه همه با تصمیم‌گیری‌های محافظه‌کارانه به دنبال موفقیت‌های کوچک ولی مطمئن هستند؛ هر چند که شاید با تصمیم‌های جاه‌طلبانه و خطر کردن بتوانند به موفقیت‌های بسیار بزرگ‌تری دست یابند. یک باور معمول در شرکت گوگل این است که اگر به دنبال دستیابی به اهداف غیرممکن باشید، به احتمال قوی به هدفتان نخواهید رسید، ولی در طول این مسیر به موفقیت‌هایی می‌رسید که در تلاش برای اهداف معمولی، به دست نمی‌آمدند. یک راه خوب برای ساختن فرهنگ تیمی که در آن اعضا حوصله ریسک و خطر کردن داشته باشند این است که به تیم خود یادآوری کنید شکست خوردن هیچ اشکالی ندارد و همیشه می‌تواند یک گزینه مورد قبول باشد.

بنابراین بپذیریم که شکست هیچ اشکالی ندارد. در واقع بهتر است به شکست به چشم فرصتی برای یادگیری سریع و درس‌های بیشتر نگاه کنیم. البته اگر در یک موضوع تکراری مرتباً شکست نخورده باشید. موضوع حائز اهمیت دیگر این است که موقع شکست به دنبال پیدا کردن مقصر نباشیم و فقط روی موضوعاتی که یاد می‌گیریم تمرکز کنیم. شکست‌های تند و سریع خیلی خوب هستند،^۱ چون هزینه کمتری دارند. شکست‌هایی که روند کندتری دارند، اگرچه دروس زیادی به ما می‌آموزند، ولی درد و رنج بیشتری دارند؛ چرا که برای ما هزینه بالاتری (معمولاً در قالب زمان برای مهندسی) را ایجاد می‌کنند. شکست‌هایی که قدری بزرگ هستند و روی نحوه سرویس‌دهی ما به مشتریان نیز تأثیر بگذارند، نامطلوب‌ترین نوع ناکامی هستند. بنابراین بیشترین تلاش برای ایجاد ساختار و پروتکل مناسب برای یادگیری از این نوع شکست‌ها باید انجام شود. در گوگل همان طور که پیش‌تر نیز اشاره کردیم، بعد از هر اشتباهی که منجر به یک مشکل برای مشتریان شود جلسه‌ای تحت عنوان «کالبد شکافی» انجام می‌شود. این کار روشی است تا مجموعه اتفاقاتی را که منجر به اتفاق حادثه شد مکتوب کنیم و سپس مجموعه‌ای از تصمیم‌ها و مسئولیت‌ها را تعریف کنیم تا جلوی رخداد مجدد این حادثه در آینده گرفته شود. این جلسه نه مکانی برای پیدا کردن مقصر است و نه محلی برای کاغذبازی‌های اداری. هدف فقط و فقط این است که هسته مسئله شکافته شود تا مشکل یک بار و برای همیشه حل و فصل شود. کار بسیار سخت، ولی تأثیرگذار است.

موفقیت‌ها و شکست‌های شخصی کمی متفاوت هستند. تمجید از موفقیت‌های شخصی اعضای تیم جایز است، ولی مقصر شمردن یک شخص زمانی که مشکلی به وجود می‌آید صرفاً یک راه برای از هم پاشی اتحاد تیمی

۱. به سخنرانی آلبرتو ساوویا به نام The Pretotyping Manifesto گوش کنید. <https://www.albertosavoia.com/>

خواهد بود. با این کار جرئت ریسک را نیز از تیمتان سلب می‌کنید. شکست هیچ اشکالی ندارد، ولی عدم موفقیت‌ها را به نام تیمتان بنویسید و نه اشخاص. اگر فردی به موفقیت رسید، او را در برابر تمام تیم تشویق کنید. ولی اگر فردی قصور یا اشتباهی مرتکب شد با او به صورت خصوصی و در قالب انتقاد سازنده صحبت کنید.^۱ از چنین فرصتی در راستای عمل به اصول HRT استفاده کنید و به تیمتان کمک کنید تا از اشتباهات و شکست‌ها درس‌های جدید یاد بگیرند.

۱. انتقاد از اشخاص در ملاعام تقریباً هیچ‌وقت ارزشی ندارد و معمولاً ظالمانه و سنگدلانه است. مطمئن باشید وقتی کسی اشتباه می‌کند تمام تیم متوجه خواهند شد. یادآوری این قضیه برای همه هیچ کمکی نمی‌کند.

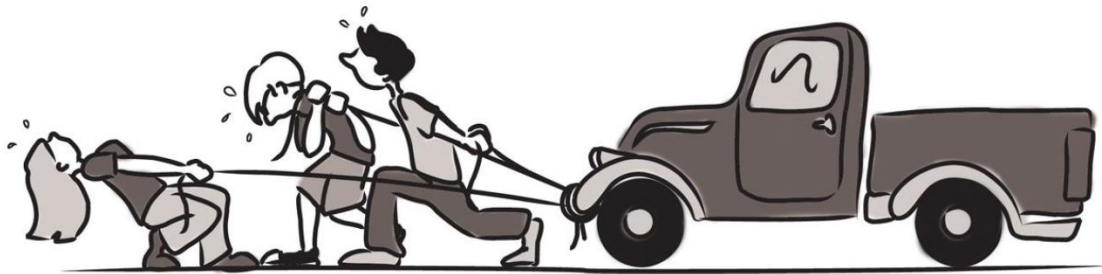
معلم و مربی تیم خود باشید

یکی از سخت‌ترین کارها به عنوان رهبر تیم این است که ببینید عضو کم‌تجربه‌ای در تیمتان ساعت‌ها در حال دست‌وپنجه نرم کردن با کاری‌ست که می‌دانید شما خود می‌توانید در کمتر از بیست دقیقه به انجام برسانید. فرصت دادن به افراد برای اینکه خودشان بتوانند کار را یاد بگیرند، معمولاً در اوایل کار مدیریت می‌تواند خیلی دشوار باشد. ولی بخش بسیار مهمی از عملکرد یک رهبر تأثیرگذار است. این کار مخصوصاً در راستای رهبری افرادی که به تازگی استخدام شده‌اند اهمیت دارد. چرا که افراد جدید علاوه بر اینکه باید تکنولوژی و کدهای پروژه شما را یاد بگیرند، باید با فرهنگ تیمی و مسئولیت‌های جدیدشان نیز آشنا شوند.

همان‌طور که معمولاً برای مدیر شدن داوطلب نمی‌شوید، بیشتر آدم‌ها برای اینکه نقش مربی را در تیمشان ایفا کنند درخواستی ثبت نمی‌کنند. معمولاً نقش مربی را وقتی پیدا می‌کنید که عضوی کم‌تجربه‌تر در تیمتان نیاز به کمک و هدایت داشته باشد. برای مربیگری نیازی به تحصیلات عالی و رسمی نیست، بلکه فقط سه چیز لازم است: تجربه روند کاری در تیمتان، توانایی توضیح دادن مسائل برای یک کارآموز و نهایتاً قدرت سنجش نیاز کارآموز به کمک. آخرین مورد احتمالاً مهم‌ترین نکته نیز هست. وظیفه شما دادن اطلاعات به کارآموزتان به میزان مورد نیاز است. اگر بیش از نیاز موضوعات مختلف را توضیح دهید و از این در و آن در حرف‌های بی‌هدف بزنید توجه کارآموزتان را از دست خواهید داد.

اهداف دقیق، مشخص کنید

این الگوی مدیریت، با آنکه خیلی بدیهی به نظر می‌آید، ولی معمولاً بسیاری از رهبران آن را فراموش می‌کنند. اگر می‌خواهید تیم شما به سمت و سوی مشخصی با سرعت حرکت کند، باید اطمینان حاصل کنید که تمام اعضای تیم بر سر مسیر و هدف نهایی با شما توافق نظر دارند. فرض کنید محصول شما یک وانت بزرگ است و هر یک از افراد در تیم شما یک تکه طناب به دست دارد که به جلوی این وانت متصل شده است. با کار کردن روی پروژ، هر کس وانت را به وسیله این طناب به سمت خودش می‌کشد. اگر هدف شما این باشد که وانت را به سمت شمال ببرید، نمی‌توانید اجازه دهید که هر یک از اعضای تیم وانت را به سمتی که خودش خواست بکشد. همه باید وانت را به سمت شمال هدایت کنند.



راحت‌ترین راه برای اینکه یک هدف دقیق برای تیمتان مشخص کنید تا تمام اعضای تیم شما محصول را به سمت درست هدایت کنند این است که یک شرح مأموریت تیمی تعریف کنید. در فصل دوم درباره شرح مأموریت تیم و نحوه نگارش آن صحبت کردیم. زمانی که مأموریت تیم را مشخص کردید و مطمئن شدید که تمام افراد حاضر در تیم شما با مسیر کلی تیم موافق هستند، می‌توانید یک قدم از تیم دور شوید تا آن‌ها استقلال و اختیار عمل را به دست بگیرند. فقط کافیست که هر از گاهی به تیم سر بزنید تا مطمئن شوید که همچنان از مسیر خارج نشده‌اند. بدین صورت، نه تنها فرصت بیشتری برای سایر وظایف خود به دست می‌آورید، بلکه کارایی تیم خودتان را نیز به شدت افزایش می‌دهید. تیم‌ها البته بدون یک هدف مشخص نیز می‌توانند به موفقیت برسند، ولی انرژی و وقت زیادی را به این دلیل که بعضی از اعضای تیم محصول را به سمت هدف اشتباهی حرکت می‌دهند، تلف می‌کنند. این حالت باعث استیصال شما و کاهش سرعت تیم می‌شود و شما را مجبور می‌کند تا دائماً برای تصحیح خط مشی تیم در کارهای تیم دخالت کنید.

صادق باشید

منظور ما این نیست که شما به تیمتان دروغ می‌گویید، ولی گاهی اوقات شما در موقعیتی قرار می‌گیرید که مجبور خواهید شد اطلاعاتی را با تیمتان به اشتراک نگذارید یا حتی بدتر مجبور می‌شوید چیزی را با آن‌ها در میان بگذارید که شنیدن آن خوشایند نباشد. یکی از مدیران سابق فیتز همیشه به اعضای جدید تیم می‌گفت: «من هیچ‌وقت به شما دروغ نمی‌گویم، ولی گاهی خواهم گفت که مطلبی را نمی‌توانم با شما به اشتراک بگذارم یا از آن اطلاع ندارم.»

اگر یکی از اعضای تیم شما از شما در مورد مطلبی سؤال کند که نمی‌توانید آن را با او به اشتراک بگذارید، هیچ اشکالی ندارد به او بگویید که شما جواب سؤال را می‌دانید ولی نمی‌توانید به او بگویید. حتی وقتی از شما سؤال می‌پرسند که جوابش را نمی‌دانید نیز خیلی راحت می‌توانید بگویید که جواب را نمی‌دانید. این هم یک مثال دیگر از الگوهای مدیریت است که ظاهراً خیلی بدیهی است، ولی بسیاری از مدیران در عمل نمی‌توانند آن را رعایت کنند. فکر می‌کنند عدم اطلاعشان نشانه‌ای از ضعف و ناتوانی آن‌ها است، در حالی که صرفاً نشانی از این خواهد بود که آن‌ها نیز یک انسان هستند.

انتقاد از یک عضو تیم... هم... خیلی دشوار است. اولین باری که می‌خواهید به کسی بگویید در کارش اشتباهی انجام داده یا صرفاً از عملکردش راضی نیستید، اصلاً راحت نخواهد بود. بیشتر کتاب‌هایی که در مورد مدیریت نوشته شده‌اند به شما پیشنهاد می‌کنند که انتقادات خود را درون یک ساندویچ تعریف و تمجید قرار دهید. انتقاد درون یک ساندویچ تمجید چیزی شبیه این خواهد بود:

«تو یک عضو قابل اطمینان و یکی از باهوش‌ترین مهندسين تیم ما هستی. ولی گدی که می‌نویسی خیلی پیچیده است و تقریباً برای هیچ‌کس دیگر در تیم قابل فهم نیست. ولی باید گفت که قابلیت‌های زیادی داری و ریشتم هم خیلی باحاله!»

بله، بدین صورت از شدت انتقاد شما کم می‌شود؛ ولی وقتی شنونده از این جلسه بیرون می‌آید به تنها چیزی که فکر می‌کند این است که چه ریش باحالی دارد! ما به شدت توصیه می‌کنیم که از ساندویچ تمجید، دوری کنید. منظور ما این نیست که هنگام انتقاد، بی‌دلیل سنگدل و ظالم باشید، بلکه می‌گوییم که قرار دادن یک نقد درون ساندویچ تمجید باعث می‌شود که شنونده پیام کلیدی شما در راستای درخواست برای تغییر را دریافت نکند. بهترین کار این است که اصول HRT را در اینجا به کار بگیرید: مهربانانه و از روی همدلی، انتقاد سازنده خود را مطرح کنید و آن را درون ساندویچ تمجید قرار ندهید. مهربانی و همدلی اهمیت بسیار بالایی دارد تا شنونده در مقابل شما آرایش دفاعی نگیرد.



چندین سال پیش، فیتز یک عضو جدید به اسم تیم را به گروهش اضافه کرد و مدیر سابقش تأکید داشت کار کردن با او غیرممکن است. او معتقد بود که تیم به هیچ یک از انتقادات و ایرادهایی که مطرح می‌شود توجه نمی‌کند و هیچ تلاشی برای بهتر شدن انجام نمی‌دهد. فیتز در یکی از جلسات مشترک آن مدیر و تیم شرکت کرد و متوجه شد که این مدیر تمام انتقادات خود را دائماً درون ساندویچ تمجید قرار می‌دهد تا مبدا احساسات تیم جریحه‌دار شود. وقتی فیتز، تیم را به گروه خودش اضافه کرد، در یک جلسه مشترک با او خیلی دقیق و مستقیم برایش توضیح داد که باید یک سری از رفتارهایش را تغییر دهد تا بتواند به خوبی کار کند. فیتز نه صحبتش را لابه‌لای تعریف و تمجیدهای نابه‌جا مخفی کرد و نه ظالمانه و بی‌ادبانه حرفش را بیان کرد، بلکه واقعیات را دقیق و بر اساس شواهد رفتاری و عملکرد سابق تیم مطرح کرد. پس از چند هفته (و تعدادی جلسه دوفره دیگر) عملکرد تیم شدیداً پیشرفت کرد. او فقط یک انتقاد سازنده و مستقیم نیاز داشت.

نحوه صحبت و طرز بیان شما، هنگام انتقاد سازنده از یکی از اعضای تیم، مهم‌ترین نکته است تا پیام شما به خوبی شنیده شود. اگر طوری نقد خود را بیان کنید که شنونده حالت تدافعی به خود بگیرد، او به اینکه چطور تغییر کند فکر نخواهد کرد، بلکه روی این متمرکز می‌شود که چطور به انتقاد پاسخ دهد و با شما بحث کند. بن مهندسی به نام دین در تیمش داشت. دین در بحث‌های تیمی روی نظراتش پافشاری بسیار زیادی داشت و تقریباً سر هر موضوعی با هم‌تیمی‌هایش بحث و جدل می‌کرد. این مشاجره‌ها می‌توانست در مورد موضوعات بزرگی مثل شرح مأموریت تیمی یا حتی درباره موارد به نسبت کم‌اهمیت‌تر مثل مکان قرار دادن یک قطعه کد روی یک وب‌سایت باشد. دین با اعتقاد راسخ و بسیار شدید همیشه سعی داشت تا حرف و نظر خودش را به کرسی بنشاند و از قبول نقطه‌نظرات دیگران خودداری می‌کرد. بعد از چند ماه که این رفتار ادامه پیدا کرد، بن تصمیم گرفت تا

با دین صحبت کند. اگر فرضاً بن به او می‌گفت: «دین... این قدر کله‌شق نباش!» مطمئن باشید که دین هیچ توجهی به این حرف نمی‌کرد. بن در مورد اینکه چطور با دین صحبت کند خیلی فکر کرد و نهایتاً به استعاره زیر رسید:

«هر زمانی که یک تصمیم گرفته می‌شود، مثل این است که یک قطار وارد شهر می‌شود. وقتی جلوی قطار بپری تا آن را متوقف کنی، سرعت این قطار را کم می‌کنی و مهندسی را که رانندگی قطار را به عهده دارد اذیت می‌کنی. هر ۱۵ دقیقه یک قطار جدید وارد شهر می‌شود و اگر جلوی هر قطار بپری، نه تنها وقت بسیار زیادی را فقط صرف جلوگیری از حرکت قطارها می‌کنی، بلکه نهایتاً یکی از مهندسین راننده یکی از قطارها را آن قدر عصبی و ناراحت می‌کنی که قید همه چیز را بزند و از روی تو رد شود. بنابراین با وجود اینکه متوقف کردن قطارها هیچ اشکالی ندارد، ولی فقط بعضی از قطارهایی را که برایت اهمیت بیشتری دارند برای بحث و توقف انتخاب کن.»

این تمثیل و حکایت، نه تنها قدری شوخی را در صحبت دخیل کرد، بلکه کار را برای بن و دین ساده‌تر کرد تا در مورد تأثیر عادت توقف قطار دین روی تیم و نیز زمانی که دین صرف این کار می‌کند، راحت‌تر صحبت کنند.

رضایت تیم را اندازه‌گیری کنید

یکی از روش‌هایی که شما به عنوان رهبر تیم برای بالا بردن بهره‌وری تیم می‌توانید به کار بگیرید این است که دائماً میزان شادی و رضایت کاری در اعضای تیم را ارزیابی کنید. بهترین رهبرانی که می‌شناسیم، اندکی روان‌شناسی بلد هستند. آن‌ها هر از گاهی به آسایش و رفاه عمومی اعضای تیم توجه می‌کنند، پیروزی‌های اعضای تیم را به رسمیت می‌شناسند و تلاش می‌کنند تا همه در کاری که انجام می‌دهند راضی و خوشحال باشند. رهبری را می‌شناسیم که همیشه لیستی از کارهای ناخوشایند و بی‌هوده‌ای را که به ناچار باید انجام می‌شدند تهیه می‌کرد تا مطمئن شود چنین کارهایی بین تمام اعضای تیم تقسیم می‌شود. و یا یک رهبر دیگر را نیز می‌شناسیم که ساعت‌های اضافه‌کاری کارمندان را یادداشت می‌کرد تا در قبال آن‌ها یا به اعضای تیمش مرخصی بیشتری بدهد یا برنامه‌های تفریحی تیمی تدارک ببیند تا از خستگی مفرط و فرسودگی در تیمش جلوگیری کند. یک مثال دیگر از رهبران خوبی که با آن‌ها کار کردیم شخصی است که در جلسات دوفره‌ای که با اعضای تیمش داشت سعی می‌کرد بعد از صحبت‌های کاری و فنی اولیه در مورد وضعیت روحی و رضایت فردی آن شخص نیز صحبت کند تا مطمئن شود که او از کارش لذت می‌برد.

یک ابزار باارزش برای سنجش میزان شادی تیم این است که در پایان هر جلسه دوفره پرسید: «چه چیزی لازم داری؟» این سؤال ساده نه تنها یک روش خوب برای جمع‌بندی جلسه است، بلکه به شما کمک می‌کند تا مطمئن شوید اعضای تیم شما به تمام آنچه که برای کار کردن و لذت بردن از شغلشان نیاز است، دسترسی دارند. البته گاهی برای دریافت جواب دقیق نیاز به اندکی سؤال و جواب و تحقیق و تفحص نیز هست. اگر به طور منظم این سؤال را در انتهای جلسات دوفره پرسید، به مرور متوجه می‌شوید که اعضای تیم شما با آمادگی قبلی بیشتری برای پاسخ دادن به جلسه می‌آیند و نهایتاً روزی یک لیست بلندبالا از موارد مورد نیازشان به شما تحویل می‌دهند.

سؤال غیرمنتظره

روزهای اول کار فیتز در گوگل و در انتهای نخستین جلسه‌ای که با مدیرعامل گوگل اریک اشمیت داشت، اریک از فیتز پرسید: «چیزی هست که نیاز داشته باشی؟» فیتز که با آمادگی کامل برای دفاع از عملکردش و جواب دادن به میلیون‌ها بازخواست احتمالی به جلسه آمده بود، در مقابل این سؤال غیرمنتظره پاسخی نداشت و مات و مبهوت خیره شد. ولی از جلسات بعدی او همیشه آماده این سؤال بود و لیست موارد مورد نیازش را حاضر می‌کرد.

گاهی اهمیت دادن به شادی و رضایت فردی اعضای تیمتان خارج از محیط کاری نیز حائز اهمیت است. تصور نکنید که افراد خارج از کارشان زندگی شخصی ندارند. با انتظارات بی‌جا از افراد بابت زمانی که می‌توانند برای کار بگذارند، احترامتان را از دست خواهید داد. به هیچ عنوان نمی‌گوییم که در زندگی شخصی اعضای تیمتان دخالت

کنید، ولی اندکی حساسیت به خرج دادن نسبت به مشکلات غیر کاری افراد باعث می‌شود که دید واقع‌بینانه‌تری نسبت به دلیل عملکرد آن‌ها در کار به دست بیاورید. وقتی انتظارات خود را از کسی که درگیر یک سری مشکلات شخصی است پایین بیاورید، خواهید دید که وقتی مشکلتش برطرف شد برای کار شما دو چندان وقت خواهد گذاشت.

بخش مهمی از سنجش میزان رضایت شغلی افراد این است که دربارهٔ مسیر شغلی و آینده حرفه‌ای آن‌ها گفت‌وگو کنید. بیشتر اوقات وقتی از کسی می‌پرسید که ۵ سال دیگر خود را در چه جایگاهی می‌بینی، جواب خاص و دقیقی نمی‌شنوید. ولی آدم‌ها معمولاً یک سری اهداف مشخص برای ۵ سال آینده خود دارند. مثلاً می‌خواهند که در کار خود ارتقا بگیرند، حرفهٔ جدیدی یاد بگیرند، محصول تازه‌ای تولید کنند، یا با افراد خاصی همکاری شوند. حتی اگر دربارهٔ این اهداف صحبت نکنند، ولی ته دل به آن‌ها فکر می‌کنند. اگر می‌خواهید رهبر تأثیرگذاری باشید، باید کمک کنید تا این اهداف برای آن‌ها محقق شوند و هم اینکه آن‌ها بدانند شما نیز به فکر اهداف آن‌ها هستید. مهم‌ترین نکته در این راستا این است که اهداف گنگ، مبهم و کلی را به صورت صریح، دقیق و واضح تعریف کنید تا به راحتی قابل اندازه‌گیری و ارزیابی باشند.

سنجش رضایت شغلی در نظارت از کار افراد خلاصه نمی‌شود. بلکه لازم است به اعضای تیم فرصت پیشرفت، یادگیری، شناخته شدن و نیز اندکی تفریح و شادی در کار را بدهید.

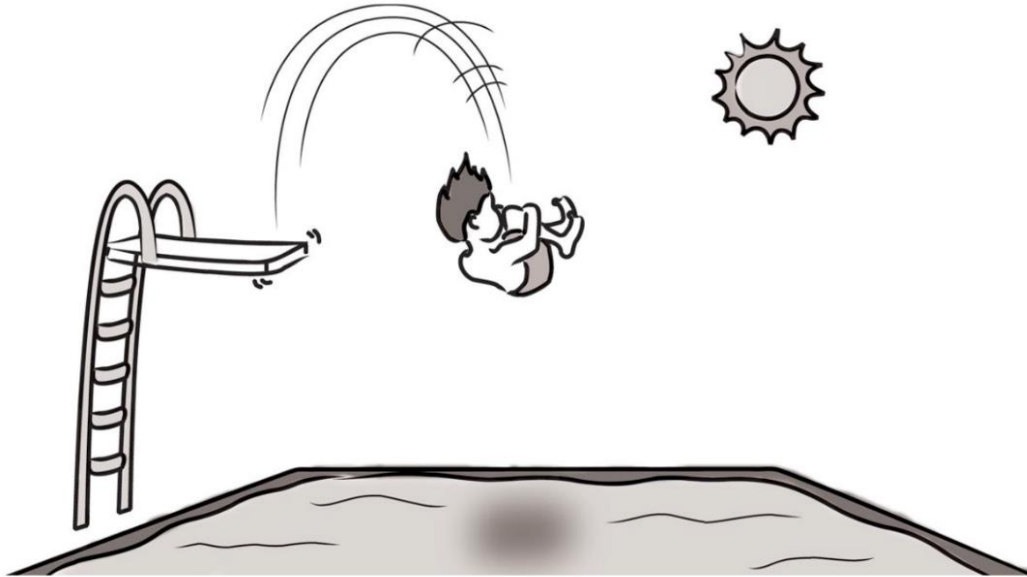
سایر ترفندهای رهبری

کارها را به دیگران محول کنید، ولی تلاش کنید که خود نیز در کار دخیل باشید. وقتی عنوان شما از مهندس به رهبر تغییر می‌کند، ابتدا یافتن تعادل بین انجام کارها و اداره تیم بسیار دشوار خواهد بود. روزهای اول بسیار تمایل دارید که خودتان کارها را به دست بگیرید و بعد از مدتی طولانی که به رهبری ادامه می‌دهید، می‌بینید کم‌کم تمام کارها را به عهده دیگران می‌گذارید و خودتان هیچ دخالتی در امور ندارید. اگر به تازگی به سمت رهبری تیم رسیدید، احتمالاً باید تلاش کنید تا کارها را به دیگران محول کنید، حتی اگر اعضای تیم شما برای انجام آن زمان بیشتری نیاز داشته باشند. این کار نه تنها باعث می‌شود که سلامت روانی خود را حفظ کنید، بلکه کمک می‌کند تا تیم شما کارها را بهتر یاد بگیرند. ولی اگر مدت زمان زیادی است که رهبر یک تیم هستید یا رهبری یک تیم جدید دیگر به عهده شما گذاشته شده است، یکی از سریع‌ترین راه‌ها برای کسب احترام اعضای تیم این است که خود را در کارها دخیل کنید. مخصوصاً اگر مسئولیت کارهای ناخوشایندی را که هیچ‌کس علاقه‌ای به انجام دادنشان ندارد به عهده بگیرید. ممکن است یک رزومه طولانی از افتخارات و توانایی‌هایتان داشته باشید، ولی هیچ‌چیز به اندازه انجام یک سری کارهای سخت به تیمتان مهارت، پشتکار و تواضع شما را نشان نمی‌دهد.

به دنبال جایگزین برای خود باشید. همیشه سعی کنید افرادی را پیدا کنید تا با انجام دادن کار شما بتوانند جای شما را بگیرند، مگر آنکه علاقه داشته باشید که یک کار را تا ابد تکرار کنید. همان طور که پیش‌تر اشاره کردیم، این کار با استخدام افراد تأثیرگذاری شروع می‌شود که توانایی جایگزینی شما را داشته باشند. به طور خلاصه: «شما باید افرادی را استخدام کنید که از شما باهوش‌ترند.» وقتی اشخاصی را پیدا کردید که می‌توانند از عهده کار شما برآیند، به آن‌ها مسئولیت‌های بیشتری محول کنید تا این فرصت را پیدا کنند که گهگاهی تیم را رهبری کنند. بدین صورت، به مرور آن دسته از اعضای تیم را که شایستگی و تمایل مسئولیت بیشتر دارند شناسایی می‌کنید. به یاد داشته باشید که عده‌ای دوست دارند صرفاً یک مهندس برجسته در تیم باشند و علاقه‌ای به رهبری نداشته باشند. البته که این هیچ اشکالی ندارد. شرکت‌هایی که بهترین مهندسين خود را به اجبار وارد پست‌های مدیریتی می‌کنند، همیشه ما را متعجب کرده‌اند. نتیجه این کارشان معمولاً کم شدن یک مهندس تأثیرگذار و اضافه شدن یک مدیر بی‌کفایت است.

در هنگام نیاز، تغییر ایجاد کنید. بی‌شک بارها در طول زندگی کاری شما موقعیت‌های دشواری پیش خواهد آمد که تک‌تک اجزای بدنتان شما را وسوسه می‌کنند تا توجهی به مسئله نکنید. این مشکل مثلاً می‌تواند یک مهندس در تیم شما باشد که توانایی فنی بالایی نداشته باشد، یا شخصی در تیم شما که خود را در برابر هر قطاری می‌اندازد، یا عضوی از تیم شما که انگیزه کار کردن را از دست داده است. به خود می‌گویید: «یک کم دیگه صبر کن، حتماً وضع بهتر می‌شه!» در این دام گرفتار نشوید. گاهی وضعیت جوری است که شما اتفاقاً باید بیشترین تغییرات را ایجاد کنید. این‌طور مشکلات به ندرت به خودی خود حل و فصل می‌شوند. هرچه که بیشتر آن‌ها را

نادیده بگیرید، بیشتر خواب را از چشمانتان می‌گیرند. صبر باعث می‌شود که مشکل کم‌کم بزرگ‌تر شده و در طول مسیر نیز آسیب‌های بیشتری به تیم شما بزند. بنابراین خیلی فوری در برابر آن‌ها کاری انجام دهید.



سپر دفاعی تیمتان در برابر شلوغی و هرج و مرج باشید. وقتی برای اولین بار رهبر تیمتان می‌شوید، متوجه خواهید شد که دنیای خارج از تیم، یک فضای شلوغ و پُرهرج و مرج، مملو از بی‌ثباتی و گاهی حتی دیوانگی است که هیچ‌گاه به عنوان یک عضو تیم از آن باخبر نبودید. در دههٔ ۹۰ میلادی، وقتی فیتز برای اولین بار رهبری تیمش را به عهده گرفت، از دیدن بلا تکلیفی و آشفتگی سازمانی در شرکت خیلی تعجب کرد. از یکی از مدیران شرکت پرسید که چرا این قدر همه چیز به هم ریخته شده است. کار در شرکت ما که همیشه منظم و پر از آرامش بود. همکار او از سادگی فیتز از خنده به قهقهه افتاد و پاسخ داد: «اینجا همیشه همین طور بوده است. فقط رهبر تیمت، تو و سایر اعضای تیم را از این فضا و محافظت می‌کرده.»

محافظت تیمتان باشید. درست است که باید تیم خود را در جریان تصمیم‌ها و اتفاقاتی که در مقاطع بالای شرکت رخ می‌دهند قرار دهید، ولی به همان اندازه نیز مهم است که از آن‌ها در برابر بلا تکلیفی‌ها و درخواست‌های بی‌پهوده که ممکن است از خارج تیم شما را تحت تأثیر قرار دهند، محافظت کنید. تا جایی که می‌توانید اخبار و اطلاعات را با تیمتان در میان بگذارید، ولی اجازه ندهید که دیوانگی سازمانی تمرکز تیم شما را به هم بزند.

وقتی تیمتان خوب عمل می‌کند، آن‌ها را تشویق کنید. بسیاری از رهبران معمولاً به اندازه‌ای روی بهبود وضعیت و تصحیح کارها تمرکز می‌کنند که فراموش می‌کنند از عملکرد مناسب تیم خود قدردانی و تشویق کنند.

همان طور که موقع اشتباه افراد تیمتان، از آن‌ها انتقاد می‌کنید، در زمان کار خوب و نتیجه مطلوب نیز آن‌ها را تشویق کنید. وقتی کاری را به نحو احسن به پایان می‌رسانند نشان دهید که به کارشان توجه کرده‌اید.

و آخر از همه آنکه بهترین رهبرها از تصمیم‌های جاه‌طلبانه آن دسته از اعضای تیم که بی‌پروا قصد انجام کار جدیدی دارند حمایت می‌کنند. وقتی چیزی را بتوان به راحتی به حالت اول برگرداند، ریسک کردن ساده می‌شود. وقتی یکی از مهندسين شما دوست دارد تا یک ابزار یا تکنولوژی جدید را تست کند که به وسیله آن شاید بتوان کارایی تیم را بیشتر کرد (و ضرب‌الأجلی برای پروژه ندارید)، به راحتی می‌توان به او چراغ سبز نشان داد. البته اگر ایده او بزرگ‌تر باشد، مثلاً انتشار محصول جدیدی که سال‌ها نیاز به پشتیبانی داشته است، شاید بهتر باشد که بیشتر درباره‌اش فکر کنید. رهبران خوب معمولاً حس ششم قوی‌ای در این زمینه دارند.

پدیده خودویرانگری^۱

درباره سندرم یا پدیده خودویرانگری مطالب بسیار زیادی نوشته شده است.^۲ ویکی‌پدیا این سندرم را این‌طور تعریف می‌کند: «یک پدیده روان‌شناسی که در آن انسان‌ها پیروزی‌ها و دستاوردهایشان را باور نمی‌کنند. با وجود آنکه شواهد و دلایل بی‌شماری توانایی و شایستگی آن‌ها را نشان می‌دهند، کاملاً متقاعد می‌شوند که لایق جایگاهی که دارند و موفقیت‌هایی که کسب کرده‌اند، نیستند.»

ما عنوان «پدیده» را برای این موضوع مناسب‌تر می‌دانیم تا «سندرم». هر چند که این پدیده به شما حس تقلبی بودن و ترس از برملا شدن را می‌دهد، ولی معمولاً باعث می‌شود که تلاش و کار بیشتری انجام دهید و به اهدافی دست یابید که به طور معمولی به آن‌ها نمی‌رسیدید.

این مشکل بین مدیران نوپا بسیار معمول است، مخصوصاً افرادی که به ناچار و از روی نیاز به مرتبه مدیریت رسیده‌اند. تقریباً همیشه چند نفر بعد از سخنرانی‌های ما از ما می‌پرسند: «من هیچ از کارم سر در نمی‌آرم. احساس می‌کنم که اشتباهی به اینجا رسیده‌ام. چه کار کنم؟» جواب معمول ما به این نوع سؤال‌ها این است که هر کس بالاخره در برهه‌ای از زندگی کاری احساس گیجی و سردرگم بودن را خواهد داشت. شاید حتی این عدم اعتمادبه‌نفس به ما انگیزه‌ای برای تلاش بیشتر و دستیابی به موفقیت‌های بالاتری را بدهد.

داستان ازدواج پدر و مادر بن بسیار جالب است. یک شب مانده به جشن عروسی هر دوی آن‌ها دچار اضطراب شده و برای ازدواج مرد شدند. هر دو اعتراف کردند که احساس می‌کنند اشتباه بزرگی مرتکب شده‌اند و برای زندگی مشترک آماده نیستند. ولی از آنجا که برای به هم زدن مراسم عروسی خیلی دیر بود، با هم تصمیم گرفتند که نقش بازی کنند و در طول جشن وانمود کنند عروس و داماد مصمم و مشتاقی هستند تا چند روز بعد از عروسی از هم جدا شوند. چند هفته بعد از عروسی دوباره با هم صحبت کردند و تصمیم گرفتند که به زندگی مشترکشان یک ماه دیگر نیز فرصت بدهند. و این داستان ماه بعد و ماه‌های بعد نیز تکرار شد تا نهایتاً تبدیل به یک شوخی دونفره شد. هر سال در روز سالگرد ازدواجشان به هم می‌گویند: «بیا این دوره آزمایشی زندگی مشترکمون رو یک سال دیگه هم ادامه بدیم. باشه؟»

در هر مقام مدیریتی که باشید، شگرد «آن قدر وانمود کن تا به واقعیت تبدیل شود»، معمولاً خیلی خوب جواب می‌دهد. اولین باری که بن رهبری یک تیم را به عهده گرفت، دقیقاً از همین روش استفاده کرد. «شما می‌خواهین که من مسئولیت این پروژه رو داشته باشم؟! بسیار خوب، فکر کنم باید مدتی وانمود کنم که رهبر تیم هستم.» و

1. Imposter Phenomenon

۲. نخستین بار دکتر کِلنس این پدیده را مطرح کرد.

هر سال زمان بررسی عملکردش می‌گفت: «فکر کنم یک سال دیگه هم بتونم وانمود کنم که رهبر تیم هستم...»
به نظر می‌آد تا الان خوب پیش رفته.»

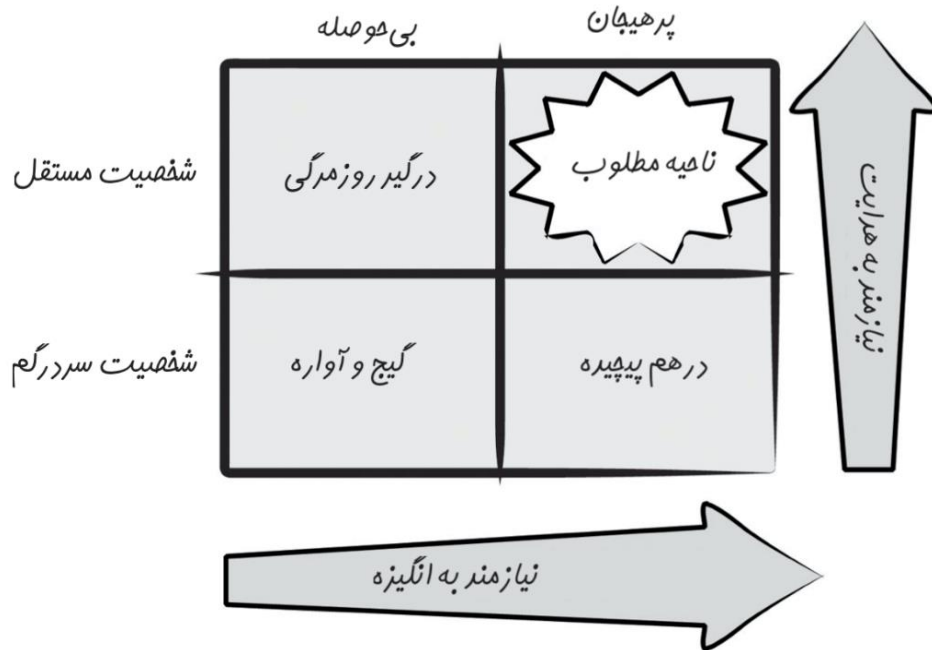
مردم مانند گل و گیاه هستند

همسر فیتز، کوچک‌ترین خواهر بین ۶ فرزند است. مادر او به سختی و تنهایی فرزندانش را بزرگ کرد. فیتز از مادرخانمش پرسید که چطور این کار را «مدیریت» می‌کرد و او این‌طور پاسخ داد: «بچه‌ها مثل گل و گیاه هستند، بعضی شبیه کاکتوس هستند که آفتاب زیاد و آب کم می‌خواهند. عده‌ای شبیه بنفشه آفریقایی هستند که به نور غیرمستقیم و خاک نمدار نیاز دارند. و بعضی دیگر مثل بوته گوجه‌فرنگی برای رشد نیازمند مقداری کود هستند. اگر برای همه ۶ فرزند خود به یک اندازه آب و نور فراهم کنید، ممکن است مساوات را رعایت کرده باشید، ولی به احتمال قوی نیاز هیچ یک از آن‌ها را برآورده نخواهید کرد.»



اعضای تیم شما هم مثل گل و گیاه هستند که بعضی نور و بعضی آب بیشتر نیاز دارند. عده‌ای از آن‌ها نیز... کود بیشتری می‌خواهند! وظیفه شما به عنوان رهبر تیم این است که تشخیص دهید هر یک از افراد تیم به چه چیزی و در چه زمانی نیاز بیشتری دارند.

به این جدول توجه کنید:



برای اینکه تمام تیمتان را در ناحیه مطلوب نگه دارید، باید به کسانی که درگیر کارهای یکنواخت و تکراری شده‌اند انگیزه بیشتری بدهید و افرادی که تمرکزشان را به دلیل هیجان برای کارهای گوناگون از دست می‌دهند، به سمت هدف تیم هدایت کنید و البته آن دسته از اعضای تیم که گیج و سردرگم هستند هم نیاز به انگیزه مضاعف و رهبری دقیق‌تر دارند. بنابراین به جای نور و آب، برای تک‌تک اعضای تیمتان باید ترکیبی مناسب از انگیزه و هدایت را فراهم کنید. فراموش نکنید که انگیزه یا رهبری بیش از اندازه وقتی کسی به آن‌ها نیازی نداشته باشد، در عمل باعث اذیت و آزار آن عضو تیم شما می‌شود.

راهنمایی افراد به سمت هدف مطلوب خیلی سخت نیست. فقط کمی مهارت اداری و اندکی آگاهی از مراحل کار را لازم دارد تا بتوان هدف را به شکل مراحل کوچک و قابل مدیریت درآورد. با همین ابزار ساده می‌توان یک مهندس سردرگم را به مسیر درست هدایت کرد (قبول، موضوع به این سادگی هم نیست. ولی در فصل‌های پیش درباره این موضوع مفصل صحبت کردیم). ولی کمک برای ایجاد انگیزه بسیار پیچیده‌تر و مفصل‌تر است.

انگیزه‌های درونی و بیرونی

انگیزه در افراد دو نوع دارد: درونی و بیرونی. انگیزه‌های بیرونی نیروهایی هستند که از دنیای اطراف می‌آیند، مثلاً حقوق و مزایای خوب. در مقابل ولی انگیزه‌های درونی، نیروهایی هستند که در باطن، انسان را به جلو تشویق می‌کنند. دن پینک در کتاب عامل محرکه^۱ توضیح می‌دهد که برای داشتن شادترین و بانگیزه‌ترین تیم، نیروهای خارجی (مثل پرتاب مقدار زیادی پول سمت آن‌ها) هیچ کمکی نمی‌کنند، بلکه باید انگیزه‌های درونی را در آن‌ها افزایش داد. دن معتقد است که سه چیز می‌تواند در راستای انگیزه درونی به آدم‌ها کمک کند: *استقلال عمل*، *خبرگی و هدف*^۲.

استقلال عمل یعنی اینکه فرد این اختیار را دارد که مسئولیت و کار خودش را پیش ببرد، بدون آنکه مدیرش در ریزترین تصمیم‌ها دخالت کند.^۳ برای یک مهندس مستقل فقط کافی است که هدف کلی را توضیح دهید و اجازه دهید که خود مسیر رسیدن به هدف را انتخاب و اجرا کنند. این کار باعث افزایش انگیزه در افراد می‌شود، چرا که رابطه آن‌ها با محصول نهایی نزدیک‌تر شده و حس مالکیت نسبت به نتیجه کار در آن‌ها تقویت می‌شود. هرچه سهم افراد در اثر نهایی بیشتر باشد، علاقه آن‌ها برای موفقیت کار بیشتر می‌شود.

ایجاد خبرگی در اصل یعنی اینکه به اعضای تیمتان فرصت یادگیری مهارت‌های جدید را بدهید. این کار نه تنها باعث افزایش انگیزه و پیشرفت کاری افراد می‌شود، بلکه کمک می‌کند که تیم قوی‌تری داشته باشید.^۴ مهارت کارمندان مثل تیزی یک چاقو می‌ماند. ممکن است از یک چاقوی برنده سال‌ها استفاده کنید، ولی اگر مرتباً آن را تیز نکنید خیلی زود ارزشش را از دست می‌دهد. فرصت کافی برای یادگیری کمک می‌کند تا اعضای تیم شما مهارت‌هایشان را «تیز» کنند تا کارآمدتر و مؤثرتر باشند.

تمام استقلال و خبرگی در دنیا برای شخصی که دلیل مشخصی برای کار کردن نداشته باشد هیچ سودی ندارد. افراد زیادی هستند که روی محصولات و پروژه‌های بسیار مهم و تأثیرگذار کار می‌کنند، ولی هیچ‌گاه در جریان نحوه استفاده مشتریان و اثرگذاری کار نهایی در دنیا قرار نمی‌گیرند. حتی اگر پروژه تیم شما خیلی بزرگ نباشد، باز هم می‌توانید که با توضیح دلایل تولید محصول، نحوه تأثیرگذاری پروژه را برایشان مشخص کنید. دیدن هدف نهایی و مشخص شدن علت کار کمک بسیار زیادی می‌کند تا انگیزه کار در تیم شما افزایش یابد. به عنوان مثال مدیری را می‌شناسیم که به ایمیل‌های مشتری‌ها توجه می‌کند. وقتی یکی از مشتریان ایمیلی ارسال می‌کند و از

1. Drive

۲. البته فرض این است که افراد به اندازه کافی حقوق می‌گیرند و اضطراب مسائل اقتصادی را ندارند.

۳. البته که خود شخص هم باید توانایی و شایستگی اعتماد برای استقلال عمل را نیز داشته باشد.

۴. البته به این معنی نیز خواهد بود که کارمندان شما فرصت‌های شغلی بیشتری خواهند داشت و راحت‌تر می‌توانند تیم شما را ترک کنند، اگر رضایت شغلی نداشته باشند.

نرم‌افزار آن‌ها تعریف و تمجید می‌کند، ایمیل را برای اعضای تیمش نیز ارسال می‌کند تا آن‌ها نیز در جریان تأثیرگذاری مثبت پروژه‌ای باشند که روی آن کار می‌کنند. این کار نه تنها باعث ایجاد انگیزه کار می‌شود، بلکه اعضای تیم را تشویق می‌کند تا راه‌هایی را پیدا کنند که نرم‌افزارشان حتی بهتر نیز بشود.

افکار نهایی

صرف نظر از اینکه رهبری یک تیم را به عهده دارید یا خیر، امیدواریم که این فصل به شما دید بهتری نسبت به یک رهبر خوب داده باشد تا افسانه‌هایی را که دربارهٔ مدیران تیم‌ها گفته می‌شوند فراموش کنید. حتی اگر مصمم هستید که هیچ‌وقت رهبری هیچ تیمی را به عهده نگیرید نیز مطالب این فصل به شما کمک می‌کند تا تصمیم‌ها و رفتارهای مدیران خودتان را چه درست و چه غلط بهتر درک کنید. به تیم خود نگاه کنید و ببینید رهبر تیم شما کدام یک از الگوهای صحیح یا اشتباه مدیریتی را اعمال می‌کند و این الگوها چه تأثیری روی موفقیت (یا عدم موفقیت) تیم شما دارند.

درک رهبر تیم و اعضای گروهی که روزانه با آن‌ها ارتباط دارید، فقط بخشی از جنبهٔ کار کردن با سایر افراد است. گاهی ممکن است مجبور شوید با افرادی سروکله بزنید که لزوماً عضو تیم شما نیستند. مخصوصاً اگر چنین افرادی در تلاش باشند تا تیم شما را تخریب کنند. چنین آدم‌هایی را «افراد سمی» می‌نامیم و در فصل بعد نحوهٔ برخورد با آن‌ها را توضیح می‌دهیم.

فصل چهارم

سروکله زدن با انسان‌های سمی

همان طور که اولین جمله کتاب هم با این عبارت شروع شد، دشوارترین قسمت کار مهندسی، انسان‌ها هستند.

تا اینجای کتاب، تمرکز ما روی رفتارهای شخصی بود. کتاب را با بررسی اصول HRT شروع کردیم و توضیح دادیم که چطور می‌توانیم تواضع، احترام و اعتماد را در کنش و واکنش‌های شخصی خودمان اعمال کنیم. سپس به روابط بین افراد یک گروه و فرهنگ تیمی بر پایه همین اصول پرداختیم و در فصل بعد ویژگی‌های رفتاری یک رهبر تأثیرگذار را شرح داده و توضیح دادیم که چطور می‌توان هنگام نیاز، خود را در قالب چنین رهبری قرار داد.

در نیمه دوم کتاب نگاهمان را تغییر می‌دهیم و روی عوامل خارجی تمرکز می‌کنیم. تیم شما در مورد ارتباط با افراد خارج از گروه چگونه رفتار می‌کند؟ چنین افرادی معمولاً یا به دنبال عضویت در تیم شما هستند یا می‌خواهند که درباره موضوعی با شما همکاری کنند. همچنین درباره روابط سازمانی و مشکلاتی که حاصل سیاست‌ورزی‌های شرکت‌های بزرگ هستند، صحبت می‌کنیم و البته به مهم‌ترین نوع از افراد بیگانه در تیم، یعنی کاربران نرم‌افزار شما نیز می‌پردازیم.

در این فصل درباره اهمیت برخورد با بیگانگانی که قصد تخریب فرهنگ تیمی شما را دارند بحث خواهیم کرد. همچنین در مورد آن دسته از افراد سمی که عضوی از تیم شما نیز هستند صحبت می‌کنیم. شاید سروکله زدن با آن‌ها از همه مهم‌تر باشند.

تعریف شخص سمی

پیش از این، اهمیت ساخت یک فرهنگ مناسب برای تیم را شرح دادیم. بیشتر وقتمان را برای توضیح خصوصیات چنین فرهنگ تیمی مؤثری صرف کردیم: ویژگی‌هایی از قبیل اجماع تیمی در تصمیم‌گیری، کیفیت کدنویسی، مرور کدها و یا محیطی که افراد برای اینکه شکست بخورند و یا چیزهای جدید را امتحان کنند، راحت باشند.

اینکه فرهنگ تیم شما چه خصوصیتی را نباید داشته باشد نیز به همین میزان حائز اهمیت است. برای ساخت یک تیم کارآمد و سریع لازم است به چیزهایی که نمی‌خواهید در تیم وجود داشته باشد نیز توجه کنید. اگرچه مهندسين با استعداد می‌توانند کارها را سریع و مفید جلو ببرند، بعضی از الگوهای غلط رفتاری در تیم می‌توانند جلوی پیشرفت و حرکت تیم شما را بگیرند و شرکت شما را به محیطی نه چندان راحت و آرام‌بخش برای کار تبدیل کنند. چنین رفتارهایی نهایتاً زنجیره‌های ارتباطی تیم شما را پاره می‌کنند و باعث شکست و نابودی پروژه می‌شوند.

نخستین بار که درباره‌ی چالش‌های اجتماعی تولید نرم‌افزار مطلبی برای سخنرانی آماده کردیم، عنوان ارائه‌مان را گذاشتیم: «با تخم‌مرغ‌های فاسد چه کنیم؟» یکی از مسئولان برگزاری کنفرانس از ما خواست که عنوان را به «پروژه‌ها چطور از دست انسان‌های سمی نجات می‌یابند» تغییر دهیم تا بلکه شاید افراد بیشتری علاقه‌مند شوند تا در ارائه‌ی ما شرکت کنند. حق با او بود و ما این ارائه را بارها و بارها در کنفرانس‌های مختلف تکرار کردیم و همیشه سالن‌ها پر بود و حتی عده‌ای دور سالن می‌ایستادند و به ما گوش می‌کردند. نه فقط بار منفی کلمه «سمی»، بلکه این واقعیت که بیشتر آدم‌ها تجربه‌ی شخصی سروکله زدن با چنین انسان‌های آزاردهنده‌ای را داشتند، باعث جذابیت بیشتر ارائه‌ی ما شد. صحبت‌های ما معمولاً به جلسات مشاوره که هر یک از حضار تجربه‌ی شخصی خودشان را به اشتراک می‌گذاشتند تبدیل می‌شد.

ولی خطری در اینجا نهفته است. به طور کلی، توجه زیاد به موضوعات با بار منفی نمی‌تواند مفید باشد و باعث درگیری‌های بیشتر در تیم شما می‌شود. لقب «افراد سمی»، صفت زشت و زنده‌ای است که خودبه‌خود مرز جداکننده‌ی بین «ما» (آدم خوب‌ها) و «آن‌ها» (آن لعنتی‌های نادان) ایجاد می‌کند. یک راه بهتر برای نگاه به این مسئله وجود دارد. به جای اینکه به تیمتان به چشم افراد نخبه‌ای نگاه کنید که مأموریت دارند تا افراد بدجنس را از گردونه خارج کنند، یک فرهنگ تیمی ایجاد کنید که رفتارهای اشتباه را تحمل نکند. به دنبال تصفیة «رفتارها» باشید نه «اشخاص». تقسیم‌بندی افراد به گروه‌های خوب و بد، حاصل یک نگاه ساده‌لوحانه است. شناسایی و محکوم کردن رفتارهای غیرقابل تحمل به مراتب عملی‌تر و سودمندتر است.

فعالاً برای ساده‌تر کردن توضیحات از لفظ «سمی» برای توصیف افرادی که رفتارهای نامناسب دارند استفاده می‌کنیم. در عمل ولی از این لغت در گفت‌وگوهای روزمره خود استفاده نکنید!

تقویت کردن تیم

مثال پخت نان و مخمر را یادتان است؟ برای پخت بهترین نان، مهم‌ترین چیز مایهٔ شروع‌کنندهٔ شما است. مهم‌ترین عامل تأثیرگذار روی فرهنگ تیمی شما در طولانی‌مدت نیز ترکیب اولیهٔ اعضای تیم شماست. اگر بنیان‌گذاران تیم شما از فرهنگ تیمی به خوبی مراقبت نکنند، عوامل خارجی به مرور روی کیفیت تیم تأثیر می‌گذارند. ولی اگر از همان ابتدا و با قدرت رفتارهای قابل قبول و غیرقابل قبول دقیق تعریف شوند، توانایی تیم پایدارتر خواهد بود.

هر دوی ما وقت زیادی را روی پروژه‌های متن‌باز صرف کردیم و تجربهٔ ما نیز کاملاً مطابق با همین ایده است.

پروژه‌ای که وقت زیادی را برایش صرف کردیم، نرم‌افزار ساب‌ورژن، ابتدا از یک گروه کوچک شروع شد. آن‌ها بی‌نهایت انسان‌های فروتنی بودند که برای هم اعتماد و احترام زیادی قائل بودند. بعد از گذر بیش از پانزده سال، حداقل ۳ تا ۴ نسل از برنامه‌نویسان در این پروژه دخیل شده‌اند (بیشتر بنیان‌گذاران پروژه دیگر در تیم نیستند). با وجود این، مشاهده می‌کنیم که فرهنگ تیمی، پایدار باقی مانده و همهٔ مهربانان و محترمانه رفتار می‌کنند و از سایرین نیز انتظار اخلاقی مشابه دارند. علت جاودانگی فرهنگ تیمی صرفاً تعریف دقیق معیارهای درست و غلط نیست، بلکه یک فرهنگ تیمی کارآمد به طور خودکار افراد مناسب را به خود جذب می‌کند. انسان‌های خوش‌برخورد به طور طبیعی به سمت گروه‌های مهربان و خوش‌رفتار حرکت می‌کنند.

قابلیت‌گزینشی فرهنگ‌های تیمی در جهت عکس نیز عمل می‌کند. بدین صورت که وقتی تیمی با گروهی از افراد عصبانی و بداخلاق شروع به کار می‌کند، معمولاً آدم‌های مشابهی نیز جذب آن‌ها می‌شوند. پروژه‌هایی که بهتر است نام برده نشوند، مثل گروه لینوکس کرنل^۱، مثال‌های خوبی در این زمینه هستند؛ سرشار از جروب‌های بی‌ثمر و ناشی از غرور و تعصب. بله، ممکن است تیم به دستاوردهای زیادی برسد، ولی کارایی کلی تیم لزوماً خیلی بالا نخواهد بود. اگر وقت و انرژی برنامه‌نویسان برای حملات شخصی و دعوای بی‌خود هدر نمی‌رفت، چقدر کار بیشتری می‌توانست انجام شود؟ چه میزان همکاری مفید از دست رفته است، فقط به این خاطر که آدم‌های محترم و خوش‌اخلاق از تیم دوری کردند؟



مجدداً این موضوع را بیان می‌کنیم تا متوجه اهمیت مسئله شوید: انسان‌های سمی تهدید مستقیمی برای کارایی تیم شما هستند. اگر به رفتارهای بد اجازه دهید که خود را بروز دهند، نه تنها عملکرد تیمتان افت می‌کند، بلکه فرهنگ تیمی شما نیز به مرور پسرفت می‌کند. بهترین دفاع، تقویت فرهنگ تیمتان با تعریف دقیق مجموعه‌ای از الگوهای مناسب رفتاری و اخلاقی است. در فصل دوم جزئیات چنین خصوصیت‌هایی را توصیف کردیم، ولی برای یادآوری، خلاصه‌ای از آن‌ها را اینجا می‌آوریم:

- شرح مأموریت تیم را تدوین کنید تا اهداف مجاز و غیرمجاز برای همه مشخص شوند.
- آداب و رسومی برای مکالمات ایمیلی تیم مکتوب کنید. از افراد جدید در تیم بخواهید که آن‌ها را خوب مطالعه کنند و از اینکه همراه عده‌ای محدود دائماً مکالمات دیگران را کنترل کنید، بپرهیزید.
- تمام گذشته کار را مکتوب کنید. نه تنها تاریخچه کدهای پروژه، بلکه شرح تصمیم‌ها، طرح نرم‌افزار مشکلات کد و اشتباهات گذشته را به دقت آرشیو کنید.
- در کار با یکدیگر همکاری کنید. همیشه از سیستم‌های کنترل نسخه در کدنویسی استفاده کنید و تغییرات را در مراحل کوچک که به سادگی قابل مرور و بازدید هستند اعمال کنید. تلاش کنید تا «ضرب اتوبوسی» تیم شما افزایش پیدا کند تا دانش در تیم پراکنده شود.
- برای رفع ایرادات کد، تست پروژه و انتشار نرم‌افزار مراحل و مقررات را دقیق تعریف کنید.

- راه را برای ورود افراد جدید به تیم هموار کنید.
- در تصمیم‌گیری‌ها به دنبال اجماع تیمی باشید، ولی برای وقتی که تیم بر سر موضوعی به توافق نمی‌رسد نیز قانونی را از پیش مشخص کنید.

نکته نهایی این است که هرچه این قواعد با حوصله بیشتری رعایت شوند، تحمل تیم شما برای رفتارهای ناهنجار کمتر خواهد شد. وقتی که فرد مزاحم به تیم شما نزدیک می‌شود، شما کاملاً آماده خواهید بود.

شناسایی تهدید

اولین قدم برای دفاع در مقابل انسان‌های سمی این است که ویژگی‌های یک خطر یا تهدید را بشناسید تا بدانید چه زمانی وقت نگرانی است.

به طور خاص چیزی که در معرض تهدید قرار می‌گیرد، دقت و تمرکز تیم شماست.

دقت و تمرکز کمیاب‌ترین سرمایه برای تیم شماست. هرچه تیم شما بزرگ‌تر باشد، ظرفیت بالاتری برای تمرکز روی توسعه محصول و حل مسائل جذاب دارد. ولی این سرمایه همیشه محدود است. اگر دائماً از آن محافظت نکنید، افراد سمی به راحتی می‌توانند روند تیم شما را به هم بزنند. تیم شما درگیر جروبحث‌های بی‌دلیل می‌شود، تمرکزش را از دست داده و بی‌روحیه می‌شود. در نتیجه اعضای تیم شما دقت و تمرکزشان را روی موضوعاتی غیر از تولید یک محصول فوق‌العاده قرار می‌دهند.



حال سؤالی که پیش می‌آید این است که انسان‌های سمی چه خصوصیتی دارند؟ برای دفاع از خود باید با ویژگی‌های چنین افرادی آشنا باشید.

تجربه به ما نشان داده است که انسان‌های ذاتاً خبیث خیلی به ندرت پیدا می‌شوند. یعنی آدم‌های خیلی کمی را می‌توان پیدا کرد که به عمد و با آگاهی قصد تخریب و ویرانگری داشته باشند. ما به چنین افرادی، گول‌بیابانی یا ترول troll می‌گوییم و معمولاً به آن‌ها محل نمی‌گذاریم. اما بیشتر افرادی که رفتارهای نامناسب دارند، عموماً یا متوجه کار اشتباهشان نیستند یا نسبت به نتیجه اعمالشان بی‌تفاوت هستند. علت اصلی این نوع رفتار بدجنسی یا قصد سوء نیست، بلکه عموماً جهل و نادانی و یا فقدان توجه و علاقه است. بیشتر رفتارهای مخرب ریشه در عدم توجه به اصول HRT دارند.

در اینجا بعضی از نشانه‌ها و الگوهای رفتاری مخرب را نام می‌بریم. وقتی این علائم را مدام در اخلاق و رفتار یک شخص می‌بینیم، خود به خود او را یک فرد سمی قلمداد کرده و در مواجهه با او با دقت و احتیاط بیشتری عمل می‌کنیم.

عدم توجه به وقت دیگران

افرادی وجود دارند که نمی‌توانند خود را در جریان کارهای پروژه قرار دهند و در نتیجه باعث اتلاف وقت سایر اعضای تیم می‌شوند. به جای اینکه مستندات پروژه، طرح‌های نرم‌افزار، شرح مأموریت تیم و بحث‌های ایمیل‌های گذشته را مطالعه کنند، دائماً حواس تیم را با سؤالاتی که جوابشان به سادگی در دسترس است، پرت می‌کنند.

در پروژه ساب‌ورژن، شرکت‌کننده‌ای داشتیم به نام چارلی که از انجمن آنلاین مباحث تیمی، به عنوان مکانی برای یادگیری و آگاهی‌یابی روزانه شخصی خود استفاده می‌کرد. در حالی که در اصل کدنویسی هیچ کمکی نداشت، هر دو یا سه ساعت یک بار خیال‌پردازی‌ها و بارش افکارش را برای همه ارسال می‌کرد. سایر اعضای تیم به ناچار مجبور به پاسخ‌گویی بودند و توضیح می‌دادند که چرا ایده‌های او اشتباه‌اند، غیرممکن هستند، پیش از این مورد بررسی قرار گرفته‌اند، مکتوب شده‌اند و یا کسی قبلاً روی آن‌ها کار کرده است. حتی بدتر، چارلی گاهی خود را در سؤالات مشتری‌های پروژه دخیل می‌کرد و جواب‌های اشتباه می‌داد. برنامه‌نویسان اصلی پروژه دائماً مجبور بودند تا به سرعت پاسخ‌های او را تصحیح کنند. مدتی طول کشید تا ما متوجه شدیم که این فرد پرشور و هیجان در اصل یک انسان سمی در پروژه ما بود که تلاش جمعی تیم را ضایع می‌کرد. کمی جلوتر در این فصل درباره نحوه برخورد ما با چارلی صحبت خواهیم کرد.

غرور و خودپسندی

شاید استفاده از کلمه «غرور» در اینجا درست نباشد، ولی منظور ما توصیف افرادی است که نتوانند تصمیم جمعی تیم را قبول کنند، برای نقطه نظرات دیگران احترامی قائل نیستند و روی ایده‌های خود به شدت پافشاری می‌کنند. چنین افرادی معمولاً بحث‌هایی را که در گذشته به نتیجه رسیده و مکتوب شده‌اند مجدداً باز می‌کنند، چرا که او در تصمیم‌گیری شرکت نکرده بوده است. برای صحبت‌های گذشته ارزشی قائل نیست و از تیم می‌خواهد که بحث‌ها را فقط به خاطر او تکرار کنند. او معمولاً درباره آینده پروژه ادعاهای بی‌اساسی دارد و معتقد است همه چیز منجر به شکست می‌شود، مگر آنکه ایده‌ها و نظرات او اجرایی شوند.

پروژه ساب‌ورژن تجربه‌ای در این راستا نیز دارد. یک مهندس باهوش در تیم یک روز سرزده اعلام کرد که به نظرش تمام طرح اولیه نرم‌افزار کاملاً اشتباه است. معتقد بود دقیقاً می‌داند کارها چطور باید پیش بروند و اصرار داشت که کل پروژه باید از ابتدا شروع شود. او حتی داوطلب شد که کارهای بازنویسی پروژه را رهبری کند، چرا که ادعا داشت بدون رهبری او، شکست و نابودی پروژه حتمی است.

یک هفته تمام صرف این شد تا بنیان‌گذاران پروژه سعی کنند این شخص را در مورد چرایی تصمیم‌های اولیه پروژه متقاعد کنند. دقت و تمرکز بسیار زیادی از تیم در این راستا از بین رفت. برای همه کاملاً واضح و مبرهن

شد که این شخص قرار نیست در مورد ایده‌هایش مدارایی بکند و حاضر نیست که نظراتش را در قالب وضعیت کنونی پروژه پیاده‌سازی کند. پیاده‌سازی کل پروژه که در مرحلهٔ بتا بود و کاربرانی در حال استفاده داشت نیز اصولاً امکان‌پذیر نبود. بالاخره کار به جایی رسید که مجبور شدیم این فرد را به حال خود بگذاریم و روی پیشرفت کار و برنامه‌های از قبل مشخص شده تمرکز کنیم. اتفاقاً سال‌ها بعد، بخش زیادی از پیش‌بینی‌های این شخص درست از آب درآمد؛ ولی به‌هیچ‌وجه مانع موفقیت‌های چشمگیر ساب‌ورژن — حداقل در دنیای سازمان‌های بزرگ — نشد.

نکتهٔ ماجرا این نیست که چه کسی درست می‌گوید و چه کسی غلط. بلکه این است که آیا امکان حل یک اختلاف‌نظر وجود دارد یا خیر و آیا یک بحث ارزش دارد تا آن را ادامه دهیم یا نه؟ این سؤالات را دائماً از خود بپرسید. جایی از داستان بالاخره باید تصمیم بگیرید که جلوی ضرر و زیان را بگیرید و روی ادامهٔ کار تمرکز کنید.

انسان حق به جانب

هر وقت با کسی برخورد داشتید که طلبکارانه از شما انجام کاری را درخواست می‌کند، حواستان را خوب جمع کنید. نمی‌توان به شخصی که انرژی خود را صرف مطالبه از بقیه و شکایت کردن می‌کند، در حالی که خود تمایلی برای کمک به حل مشکل ندارد اعتماد کرد. گاهی فردی با چنین حس حق به جانبی تبدیل به یک انسان سمی برای تیم شما می‌شود. وقتی تیم پروجکت هاستینگ را در گوگل هدایت می‌کردیم، یکی از مدیران پروژه از ما خواست تا نگذاریم یکی از کاربران به دلیل رفتار زشت و خارج از اخلاقش، از نرم‌افزار استفاده کند. یک شبیه‌ساز بازی کامپیوتری رایگان و متن‌باز روی بازی مورد علاقه این کاربر خوب عمل نمی‌کرد. او با ثبت یک گزارش و ایراد در پروژه، سعی کرد با بی‌ادبی تمام مشککش را حل کند. برنامه‌نویسان پروژه مؤدبانه توضیح دادند که چرا شبیه‌ساز برای بازی او هنوز آماده نبود و چرا حل این مشکل به زودی امکان‌پذیر نخواهد بود. چنین پاسخی برای این کاربر به هیچ‌وجه قابل قبول نبود. هر روز روش جدیدی برای آزار و اذیت برنامه‌نویسان پیدا می‌کرد. گزارش ایرادهای متعدد و تکراری ثبت می‌کرد و روی گزارش ایرادهای سایرین یادداشت‌های بی‌ادبانه می‌گذاشت، در باب اینکه برنامه‌نویس‌های «احمق» شرکت حاضر نیستند مشکل او را حل کنند. طرز بیان او روز به روز زشت‌تر و حتی مستهجن‌تر می‌شد تا جایی که برنامه‌نویسان و سرپرستان پروژه مدام به او تذکر می‌دادند. علی‌رغم تمام تلاش‌های ما برای جلوگیری از این رفتار، او به کار خود ادامه می‌داد و تلاشی برای تغییر رویکردش نداشت. بنابراین ما نهایتاً مجبور شدیم، به عنوان آخرین راه‌حل ممکن، کاری کنیم تا او نتواند از نرم‌افزار شرکت استفاده کند.

ارتباطات بچگانه یا گیج کننده

بعضی اوقات افرادی را می بینیم که از اسم اصلی خود برای نام کاربری استفاده نمی کنند و به جای آن اسامی بچگانه نظیر "SuperCamel"، "jubjub89"، یا "SirHacksalot" را به کار می گیرند. حتی بدتر گاهی از نام های متفاوت در مکان های مختلف استفاده می کنند. یک نام برای ایمیل، یکی برای نرم افزارهای پیام رسان و شاید نام دیگری برای ارسال کد. در بعضی از شرایط غیرمعارف نیز اشخاصی را می بینیم که از کلمات عجیب مثل lol، عدد به جای حروف، حروف تماماً بزرگ لاتین و یا نشانه گذاری های بیش از حد استفاده می کنند!!!!!!؟!

بدگمانی و دشمن پنداری

همان طور که پیش تر در مثال های قبل دیدیم، گاهی حس حق به جانبی و خودخواهی افراد باعث می شود که نسبت به پروژه خصومت های شخصی پیدا کنند. به دفعات دیده شده که چنین حسی به مرور تبدیل به خیال پردازی می شود و بدگمانی زیاد منجر به دشمن پنداری پروژه و اعضای تیم برای فرد سمی می شود. چنین فردی وقتی که بر سر موضوعی با تیم به توافق نمی رسد، به سرعت خیال پردازی می کند و نظریه توطئه می بافت. گاهی به شکلی خنده دار، فرض می کند اعضای تیم به قدری به او اهمیت می دهند که تمام وقت خود را صرف این می کنند علیه او دسیسه و توطئه بچینند و نقشه بکشند و اگر محیط تیم شما مبتنی بر اصول ارتباطی باز و شفاف (مطابق آنچه در فصل دوم توضیح دادیم) باشد، تمام مکالمات و بحث ها برای دسترسی همه مکتوب شده و نظریه وجود توطئه را خنده دارتر هم می کند. توصیه ما این است که به چنین ادعاهایی اهمیت ندهید و حتی به آن ها پاسخی هم ندهید. وقتی شخص سمی این قدر نسبت به تیم و پروژه بدبین شده است، هر حرفی که شما بزنید اعتماد او به شما کمتر و کمتر می شود. پس چرا به خود زحمت بدهید و چیزی را برایش توضیح دهید؟ بهترین کار این است که روی توسعه نرم افزار و تولید محصول تمرکز کنید.

کمال گرایی

در نگاه اول، انسان‌های کمال‌گرا به هیچ‌وجه خطرناک به نظر نمی‌آیند. بلکه گاهی ممکن است رفتارهای به شدت وسواسی از خود نشان دهند، ولی عموماً انسان‌های متواضع و محترمی هستند که به خوبی به صحبت‌های شما گوش می‌دهند. همه چیز مطابق اصول HRT به نظر می‌آید، پس مشکل چیست؟ مشکل، خطر از کار افتادن پیشرفت پروژه است.

به یکی از افرادی که قبلاً با او همکاری بوده‌ایم نگاه کنید. پاتریک مهندس خارق‌العاده‌ای بود. طرح‌های خیلی خوبی ارائه می‌کرد، کدنویسی با کیفیتی داشت و به راحتی می‌شد با او ارتباطات مؤثری برقرار کرد. ولی متأسفانه وقتی قرار می‌شد معماری یک نرم‌افزار را طراحی کنیم، امکان داشت پاتریک تمام عمر خودش را روی این طراحی بگذارد. هیچ‌وقت از طراحی که انجام می‌داد راضی نمی‌شد و در نتیجه هیچ‌وقت برای شروع کدنویسی کار آماده نبود. با وجود اینکه نقطه نظرات منطقی و شواهد دقیقی در طراحی‌های خود داشت، سایر اعضای تیم به مرور خسته و کلافه شدند. با ادامه این موضوع همه تیم به این نتیجه رسیدند که احتمالاً کار کدنویسی هیچ‌وقت شروع نخواهد شد. تعدادی از ما با هم صحبت کردیم تا برای خروج از این وضعیت تصمیمی بگیریم. پاتریک نیروی بسیار مفیدی در تیم ما بود، ولی از سوی دیگر جلوی پیشرفت کار تیم را گرفته بود. هر زمان که می‌خواستیم کدنویسی را شروع کنیم، او محترمانه مشکلات احتمالی‌ای را مطرح می‌کرد که ممکن بود در آینده دور به وجود بیایند. بدون آنکه خود متوجه باشد، باعث شده بود تا تیم فلج شود. در قسمت بعد درباره این که چطور این مسئله را حل کردیم صحبت خواهیم کرد.

سم زدایی

ما طرفدار اخراج آدم‌ها از گروه صرفاً به خاطر یک رفتار زشت و اشتباه نیستیم. همان طور که قبلاً اشاره شد، نباید افراد را به دو دسته «ما» آدم خوب‌ها و «آن» بی‌ادب‌ها تقسیم کنیم. در مثال‌هایی که آوردیم، هدف ما از میان برداشتن شخص خطاکار نبود، بلکه ترجیح می‌دادیم رفتارهای اشتباه را از بین ببریم. طوری عمل کنید تا رفتارهای سمی جایی در تیم پیدا نکنند. فقط وقتی کسی بعد از اخطارهای متعدد همچنان کارهای اشتباهش را ادامه می‌دهد، می‌توان به این فکر کرد که بهتر است او را اخراج کنیم یا از تیم کنار بگذاریم.

تلاش و تمرکز شما روی از بین بردن رفتارهای سمی، معمولاً مهندسین باهوش تیمتان را (حتی اگر از هوش اجتماعی بالایی برخوردار نباشند)، تبدیل به افراد بسیار کارآمدتری می‌کند. چند سال پیش عضوی در تیم داشتیم که اگرچه از لحاظ فنی بسیار توانمند بود، ولی هر از گاهی با حرف‌های نابه‌جا باعث کدورت خاطر همکارانش

می‌شد. یکی از ما او را کنار کشید و از او پرسید که آیا متوجه هست تا چه حد کلماتش باعث آزار دیگران می‌شوند؟ از شنیدن این سؤال به شدت تعجب کرد و با وجود آنکه کاملاً درک نمی‌کرد چرا طرز صحبت او باعث ناراحتی بعضی از هم‌تیمی‌های او می‌شود، پذیرفت که در انتخاب کلمات و نوع مکالمه تغییراتی اعمال کند تا همکار خوبی در تیم باشد. همه چیز به نحو احسن پیش رفت، در رفتارش تجدیدنظر کرد و مسئله ریشه کن شد. پایان هر داستانی تلخ نیست.

تا اینجا خصوصیات یک عضو سمی در تیم را شناختیم. شاید همین الان کسی را در تیم خود بشناسید که با نشان دادن بعضی از الگوهای رفتاری سمی، باعث شود انرژی تیمی شما هدر رود. چطور می‌توان با این قضیه کنار آمد؟ در این بخش استراتژی‌هایی را برای برخورد با افراد سمی مرور خواهیم کرد.

انرژی افراد کمال‌گرا را به سمت دیگری هدایت کنید

وقتی یک فرد کمال‌گرا در تیم شما موفق شد به یک راه‌حل قابل‌قبول و منطقی برای یک مشکل دست پیدا کند، مسئولیت مشکل جدیدی را به او بسپارید.

پاتریک در پروژه ساب‌ورژن را به یاد دارید؟ نهایتاً یک روز صبر ما تمام شد و به او گفتیم: «بسیار خوب، ما فکر می‌کنیم همین طرحی که تا الان آماده کردی قابلیت اجرایی شدن را دارد. ما کدنویسی را شروع می‌کنیم و امیدواریم که تو هم بتوانی به ما کمک کنی.» با کمال تعجب دیدیم که پاتریک هیچ مخالفتی نداشت، از حرف ما ناراحت نشد و او هم کدنویسی روی پروژه را در کنار ما کار کرد.

یک ضرب‌المثل قدیمی این‌طور می‌گوید: «اجازه ندهید که کار بی‌نقص، دشمن کار خوب باشد.» برای داشتن یک تیم با کارایی بالا، مبارزه با کمال‌گرایی به اندازه مبارزه با سایر رفتارهای سمی که شاید واضح‌تر باشند، اهمیت دارد.

این استراتژی برای انسان‌های حق‌به‌جانبی که به جای کمک کردن و مشارکت در پروژه مدام به کار ایراد می‌گیرند نیز جوابگوست. خیلی راحت می‌توان به چنین افرادی این‌طور پاسخ داد: «ما از کمک تو برای حل مشکل استقبال می‌کنیم.» بدین صورت، خیلی مؤدبانه به او می‌گوییم که یا خودت کار را بهتر کن و مشکل را حل کن یا ساکت باش. این روش به او کمک می‌کند که همچنان ایراد کار خودش را پیدا کند، ولی حداقل این بار به نحوی این کار را انجام دهد که برای ما و پروژه مفید باشد.

هیولا را تغذیه نکنید

این جمله، ضرب‌المثلی است که در مکالمات پروژه سیستم‌های Usenet^۱ متداول بود. این استراتژی به طور خاص در مواجهه با هیولاهایی (trolls) که از عمد قصد تخریب کار و روحیه تیم شما را دارند می‌تواند مفید باشد. هرچه بیشتر به آن‌ها پاسخ دهید بیشتر آن‌ها را تغذیه کرده و وقت خود را تلف می‌کنید. سکوت و بی‌مחلی در مقابل آن‌ها معمولاً راه‌حل ساده و مؤثری است. احتمالاً از ته دل خیلی دوست دارید که با یک جمله کوبنده او را سر جایش بنشانید، ولی در مقابل این وسوسه مقاومت کنید. وقتی متوجه شود هیچ‌کس به او توجهی ندارد، از علاقه او به آزار و اذیت شما کم می‌شود و دست از سر شما بر می‌دارد. کار آسانی نیست و سکوت در مقابل چنین افرادی معمولاً تلاش زیاد و اراده قدرتمندی می‌خواهد. قوی باشید!



۱. البته اصل این ضرب‌المثل احتمالاً به فیلم پیش‌تازان فضا (Star Trek) برمی‌گردد که در آن احساسات منفی باعث تغذیه یک هیولا می‌شدند. آن‌ها موفق شدند با این دستور که احساسات منفی کنار گذاشته شوند تا این هیولا گرسنه بماند، او را از قلمرو خود خارج کنند.

بیش از حد احساسی نباشید

خیلی سخت است که در مقابل کسی که حتی ناخودآگاه رفتارهای سمی از خود بروز می‌دهد، حالت دفاعی و پرخاشگراییانه نگرفت. افرادی که شما را به دروغ متهم به توطئه می‌کنند، تصمیمات فنی شما را بی‌دلیل زیر سؤال می‌برند و یا حتی صرفاً با سوال‌های بی‌شماری که جوابشان در دسترس است وقت شما را می‌گیرند، به راحتی موجب آزرده‌گی خاطر شما می‌شوند. یادتان باشد که کار شما تولید محصولات خارق‌العاده است و وظیفه ندارید هر بازدیدکننده را راضی نگه دارید و همه را توجیه کنید. هرچه بیشتر درگیر احساسات بشوید، وقت بیشتری را برای نوشتن پاسخ‌های پرشور و آتشین برای کسی که احتمالاً لیاقت توجه شما را ندارد، هدر می‌دهید. موضوعاتی را که برایتان ارزش مبارزه دارند، با دقت انتخاب کنید و خود را درگیر هر نبردی نکنید. آرامش خود را حفظ کنید و ببینید که چه کسی ارزش وقت و توجه شما را دارد و چه کسانی را باید به حال خود رها کنید.

به دنبال حقایق مسلم باشید

بهترین نتیجه‌گیری در ادامه بحث کنترل احساسات این است که همیشه روی حقایق و واقعیت‌های مسلم تمرکز کنید. به شکایات افراد با دقت گوش کنید. همیشه با این پیش‌فرض با افراد روبه‌رو شوید که آن‌ها نیت خیر و مثبتی دارند، حتی اگر لحنی عصبانی و کلامی بی‌ادبانه داشته باشند. آیا در صحبت‌های او نکته خوبی نهفته است؟ آیا می‌توان از تجربه او چیز جدیدی را یاد گرفت یا ایده نابی را دنبال کرد؟ بیشتر اوقات جواب مثبت است. معمولاً در میان صحبت‌های نیش‌دار یک فرد سمی نکات مفیدی پنهان است. همیشه موضوع بحث را محدود به مسائلی فنی نگه دارید.^۱

مثالی که در این زمینه خیلی موردعلاقه ما قرار دارد، در مورد ایمیل خشناک یکی از رهبران معروف پروژه متن‌بازی است که به دستمان رسید. موضوع ایمیل، گزارش یک ایراد در نرم‌افزار ما بود، ولی نحوه نگارش بیشتر شبیه یک حمله بی‌رحمانه به هوش و استعداد ما بود. تمام متن پیامش پر بود از تهمت و افترا و مبالغه. به نظر می‌آمد که هدف صرفاً ایجاد التهاب و ناراحتی برای تیم ما بود تا تلاش برای رفع ایراد کد. یکی از اعضای تیم به ایمیل او پاسخ مختصری داد و چند سؤال فنی درباره مشکلی که گزارش شده بود پرسید. فرد گزارش‌کننده سؤالات را جواب داد، ولی پاسخ او همچنان در قالب همان لحن عصبانی و پُر تنش بود. همکار ما همچنان به کلام سمی بی‌توجه بود و به سادگی پاسخ داد: «ممنون بابت گزارش این ایراد. فهمیدیم که چگونه ایراد را در نسخه بعدی برطرف کنیم.»

به همکارمان به خاطر نحوه پاسخ‌گویی او به شدت افتخار کردیم. او با حفظ آرامش و تمرکز روی حقیقت‌های فنی مسئله، طوری رفتار کرد که هرچه مکالمه جلو رفت شخص مقابل بیشتر شبیه انسان‌های مجنون و دیوانه به نظر می‌آمد. در پایان این رهبر معروف دنیای متن‌باز که ایراد ما را گزارش کرده بود، تمام احترام و اعتبار خود را از دست داده بود و کمتر نزد ما می‌آمد.

۱. در مورد این موضوع کتاب *The Retrospective Prime* از Norman Kerth را مطالعه کنید.

هیولاهای را با سلاح مهربانی شکست دهید

در ادامه بحث حفظ آرامش و پیگیری حقایق مسلم، گاهی می‌توان صرفاً با مهربانی بیش از اندازه روبه‌روی انسان‌های سمی ایستاد. به مکالمه‌ای که در اتاق گفت‌وگوی اینترنتی پروژه ساب‌ورژن اتفاق افتاد توجه کنید:

هَری: این ساب‌ورژن واقعاً مزخرفه! اعصاب من رو خُرد کرده.

ساسمن: اگر کمکی لازم داری، بگو.

هَری: می‌خوام با cvs فایل‌های یه نفر رو از روی مخزن کد بردارم، ولی این یارو از یه چیزی به اسم ساب‌ورژن داره استفاده می‌کنه. به جای cvs فقط با svn کار می‌کنه.

ساسمن: خوب از یه نرم‌افزار svn استفاده کن تا فایل‌هاش رو بتونی ببینی.

هَری: خوب حالا گیرم من رفتم و ساب‌ورژن رو هم دانلود کردم. بعدش می‌تونم راحت بقیه مواردش رو پیکربندی و راه‌اندازی و نصب کنم، مثل cvs؟ معلومه که نه! البته به نظر من بیشتر تقصیر این یارو هست که داره از ساب‌ورژن استفاده می‌کنه تا آدم‌های پشت پروژه ساب‌ورژن.

ساسمن: صرفاً به این دلیل که تو بلد نیستی سیستم رو پیکربندی کنی و بعدش نصب و راه‌اندازی کنی، به این معنی نیست که ساب‌ورژن یه ایراد بزرگ داره. مردم هر روز دارن ازش استفاده می‌کنن.

هَری: من نگفتم که ساب‌ورژن کلاً باگ داره.

ساسمن: واقعاً فکر می‌کنی که ما نسخه‌های مختلف رو منتشر می‌کردیم، اگه همچین ایراد بزرگی وجود داشت؟!

هَری: بابا من فقط دارم در مورد این یارو غر می‌زنم. به نظر می‌آد که باید expat یا libxml رو نصب کنم. ای خدا...

ساسمن: اونا معمولاً به صورت پیش‌فرض روی اکثر سیستم‌ها وجود دارن.

ساسمن: این دوست شما از سرور آپاچی استفاده می‌کنه؟ شاید فقط کافی باشه که فایل Binary رو دانلود کنی.

هَری: نمی‌دونم. فقط گفت svn.

ساسمن: از کدوم نسخه سیستم‌عامل استفاده می‌کنی؟

هَری: FreeBSD

ساسمن: فقط کافیه وارد Ports Tree بشی و پُرت مربوط به آن را باز کنی.

هَری: شما دارین غر زدن های من رو خراب می کنید. من اومدم اینجا یه کم دعوا کنیم خالی بشم... شما بیش از حد دارین کمک می کنید.

ساسمن فقط یک استیکر لبخند می گذارد.

هَری: از کی تا حالا آدم وقتی می آد اتاق گفت و گو همه مهربون می شن و می خوان کمک کنن؟! مسخره...

... هَری از اتاق گفت و گو خارج شد.

گاهی باید تسلیم شد

بعضی اوقات می‌بینید که هر چقدر تلاش می‌کنید فایده‌ای ندارد. باید بی‌خیال شد و راه را ادامه داد. گاهی حتی بعد از اینکه وقت و انرژی زیادی را در راستای تلاش برای بهبود یک اخلاق سمی هدر داده‌اید، باید قبول کنید که شکست خورده‌اید.

به داستانمان درباره چارلی بازگردیم. فیلسوف مهربانی که دائماً در حال ارسال ایمیل‌های گروهی بود. عاقبت ما آمار ایمیل‌ها را بررسی کردیم و متوجه شدیم که چارلی ظرف مدت دو ماه به رده سوم از لحاظ تعداد ایمیل‌های ارسالی رسیده بود. نفر اول و دوم هر دو از مشارکت‌کنندگان هسته اصلی پروژه بودند، که البته ۷۰ درصد از ایمیل‌های ارسالی آن‌ها نیز صرفاً پاسخ به صحبت‌های چارلی بود. وقت و انرژی ما کاملاً تلف شده بود، در حالی که چارلی هیچ سوءنیتی هم نداشت. راه‌حل نهایی ما این بود که به چارلی به صورت خصوصی ایمیل بزنیم و مؤدبانه از او خواهش کنیم که ایمیل‌هایش را کمتر کند. مکالمه دشواری بود، مخصوصاً که او نمی‌توانست میزان آسیبی را که به تیم می‌زند درک کند. چند هفته گذشت و تغییری در طرز رفتار چارلی دیده نشد. نهایتاً یکی از ما طی یک گفت‌وگوی تلفنی طولانی و سخت او را قانع کرد که کلاً دست از ارسال پیام به تیم بردارد. چارلی با ناراحتی و سردرگمی به درخواست تیم احترام گذاشت و کنار رفت. همه ما کمی از این اتفاق ناراحت بودیم، ولی می‌دانستیم که کار درستی را انجام دادیم، چرا که چارلی هیچ‌وقت متوجه اشتباهش نشد. موقعیت ظریفی بود که با دشواری و با اتکا به اصول HRT با رعایت ادب و احترام حل شد.

روی اهداف درازمدت تمرکز کنید

مسیر موفقیت یک پروژه لبریز از هزاران عامل حواس‌پرتی است. نقطه اشتراک در تمام تجربه‌های سروکله زدن با افراد سمی این است که در تمام آن‌ها به راحتی می‌توان اسیر حواشی شد. وقتی با یک رفتاری که به ظاهر سمی است، مواجه می‌شوید دو سؤال اساسی را باید از خود بپرسید:

- حتی اگر دقت و توجه تیم شما در کوتاه‌مدت درگیر شود، آیا واقعاً فکر می‌کنید این رفتار در طولانی‌مدت به نفع شما خواهد بود؟
- آیا فکر می‌کنید مسئله به خودی خود حل و فصل می‌شود؟



اگر جواب شما به هر یک از این دو سؤال «نه» باشد، در اسرع وقت باید دست‌به‌کار شوید و با رفتار سمی برخورد کنید. ممکن است این‌طور به نظر بیاید که شاید این رفتار پرخطر موقت است و شاید تحمل آن در کوتاه‌مدت ارزش داشته باشد، ولی معمولاً این‌گونه نیست. مثلاً فردی ممکن است از نظر فنی بسیار قوی باشد، ولی همچنان از خود رفتارهای سمی بروز دهد. ممکن است وسوسه شوید تا از رفتارهای پرخطر او به خاطر مشارکت‌های تکنیکی و فنی او در پروژه چشم‌پوشی کنید. ولی مواظب باشید! یک فرهنگ تیمی قوی بر اساس

اصول HRT غیر قابل جایگزین است، ولی مهندسين باهوش و توانا قطعاً پيدا می‌شوند. به نقل از یکی از همکاران قدیمی:

«دوستان زیادی دارم که او را کم و بیش می‌شناسند. یکی از آنها می‌گفت: فلانی روی یک مرز باریک بین نابغه و بی‌شعوری در حرکت. مشکل اصلی اینه که این روزها نابغه‌ها به قدری زیاد هستند که دیگه نمی‌شه بهشون به چشم یک استعداد نایاب نگاه کرد.»
گرگ هادسن^۱

البته گرگ اینجا در مورد نابغه‌های خاص و نایاب حرف نمی‌زد. منظور او این بود که امروزه برنامه‌نویس‌های شایسته و با استعداد زیادی در دنیا پیدا می‌شوند. اگر شما برنامه‌نویس باهوشی دارید که اخلاق و رفتار مناسبی ندارد و فرهنگ بلندمدت تیم شما را تهدید می‌کند، بهتر است به دنبال شخص دیگری بگردید.

چنین اتفاقی را یک بار در پروژه ساب‌ورژن تجربه کردیم. تیم قانونی وضع کرده بود که تأکید می‌کرد هیچ‌کس اجازه ندارد اسم خود را درون فایل‌های کد قرار دهد (همان قانونی که در فصل دوم درباره‌اش صحبت کردیم). معتقدیم که این کار باعث ایجاد قلمروهای شخصی غیر قابل مدیریت می‌شود. اعضای تیم جرئت نمی‌کنند که به کدی که نام شخص دیگری را دارد دست بزنند و اصولاً ضریب ثابت اتوبوسی تیم را پایین می‌آورد. در عوض با کمک سیستم کنترل نسخه‌های کد به مشارکت‌کنندگان پروژه اعتبار مناسب می‌دهیم و اسامی تمام اعضای تیم را در یک فایل مشخص لیست می‌کنیم.

یک روز یک برنامه‌نویس باهوش وارد پروژه شد و داوطلب شد تا یک قابلیت به نسبت بزرگ و بسیار مورد نیاز را به کد اضافه کند. او کدش را برای بازبینی اعضای پروژه فرستاد. مهم‌ترین درخواست ما این بود که او صرفاً نامش را از ابتدای فایل بردارد و ما نام او را به لیست اعضای پروژه (مانند سایرین) اضافه خواهیم کرد. او قبول نکرد و بحث ما با او نیز به جایی نرسید. نهایتاً مجبور شدیم که کد او را رد کنیم. او و قابلیت جدید پروژه هر دو از تیم کنار رفتند. البته که همه ما ناراحت شدیم، ولی به‌هیچ‌وجه نمی‌خواستیم که قوانین خودمان را نقض و فرهنگ تیم را فقط به خاطر به دست آوردن سریع‌تر یک قابلیت، ضعیف کنیم. چند ماه بعد بالاخره شخص دیگری این قابلیت را به پروژه ما اضافه کرد.

دقیق می‌گوییم: نفع و سود کوتاه‌مدت ارزش لطمه زدن به فرهنگ تیمی را ندارند. مخصوصاً اگر موضوع یک برنامه‌نویس باهوش باشد که به اصول HRT توجه نمی‌کند.

1. Greg Hudson

نتیجه

در این فصل داستان‌ها و شخصیت‌های زیادی را شرح دادیم. پس از شنیدن تمام نکات حق دارید اگر یک حسی از بدبینی و دشمن‌پنداری در ته دل شما شکل گرفته باشد. ولی یادتان باشد که دنیا مملو از آدم‌های بی‌شعور نیست. یکی از دوستان ما می‌گفت: «آره، یه تعداد آدم دیوانه و روانی اون بیرون وجود دارند، فقط اینترنت باعث شده حس کنیم همین بغل دست ما هستند.»

یا به نقل قول از رابرت هانلن:

«رفتارهای اشتباه افراد ناشی از عدم توجه و حماقت آن‌هاست نه بدذاتی و کینه‌ورزی.»

به جای کلمه «حماقت» ما ترجیح می‌دهیم از «جهالت» یا «بی‌خبری» استفاده کنیم، ولی ایده کلی جمله یکی است. همان طور که در ابتدای فصل توضیح دادیم، دسته‌بندی آدم‌ها به یک گروه خوب و یک گروه بد بسیار ساده‌لوحانه است. افرادی که به عمد قصد تخریب فرهنگ تیم شما را داشته باشند به ندرت پیدا می‌شوند. معمولاً چنین رفتاری ناشی از اطلاعات غلط یا گمراهی آدم‌هاست. یا حتی شاید آن‌ها از لحاظ اجتماعی خجالتی باشند و با چنین اخلاقی به دنبال توجه سایرین هستند. هر دلیلی که وجود داشته باشد، وظیفه شما این نیست که کدخدایی کنید و انسان‌های گمراه را از پروژه بیرون کنید، بلکه وظیفه شما این است که روبه‌روی رفتارهای مخرب بایستید و انتظاراتی بر پایه اصول HRT تثبیت کنید تا واکنش‌های سمی جایی در تیم پیدا نکنند. تمییز دادن این دو از هم نیازمند ذکاوت شما و ایفای بی‌نقص این وظیفه نیازمند مهارت شماست.

فصل پنجم

دغل کاری های سازمانی

تاکنون دربارهٔ بُعد انسانی و اخلاقی شخص شما و نیز تیمتان صحبت کردیم. مهارت های بنیادی برای رهبری تیم و همچنین راه های مبارزه با انسان های سمی را مرور کردیم. علاوه بر موضوعات مطرح شده، لازم است راه و روش ادارهٔ امور در سلسله مراتب سازمانی شرکت های خوب و بد (سمی) را نیز یاد بگیرید. بیشتر آدم ها در شرکت های به هم ریخته کار می کنند و درگیر کاغذبازی های بی خود سازمانی هستند. بنابراین برای به ثمر رساندن کارها نیاز است تا تکنیک های خاصی را به کار بگیرند. بعضی نام این مهارت ها را «سیاستمداری» می گذارند و بعضی دیگر آن ها را «مهندسی اجتماعی» تلقی می کنند.

ما این مهارت ها را «دغل کاری های سازمانی» می نامیم.

خوب، بد و استراتژی‌ها

شرکت‌های بزرگ مثل یک معمای پرپیچ‌وخم هستند. برای عبور و راه‌گشایی حتی در بهترین شرکت‌ها، به یک چراغ‌قوه قوی، GPS و یک کامیون بزرگ خُرده نان (برای نشانه‌گذاری مسیر) نیاز است.



ابتدا وضعیت یک تیم را بررسی می‌کنیم که در یک محیط ایدئال در یک شرکت کارآمد مشغول به کار است. سپس راه‌های مختلفی را که شرکت‌های ناب‌سامان سد راه موفقیت تیم‌ها می‌شوند مرور می‌کنیم. پس از آن استراتژی‌هایی را مطرح می‌کنیم که به وسیله آن‌ها یک تیم می‌تواند در هر نوع شرکتی موفق شود. نهایتاً اگر هیچ یک از استراتژی‌ها و مهارت‌های مطرح‌شده موفقیت‌آمیز نبودند، استراتژی جایگزین (Plan B) را به کار می‌گیریم.

جوری که شرایط باید باشند

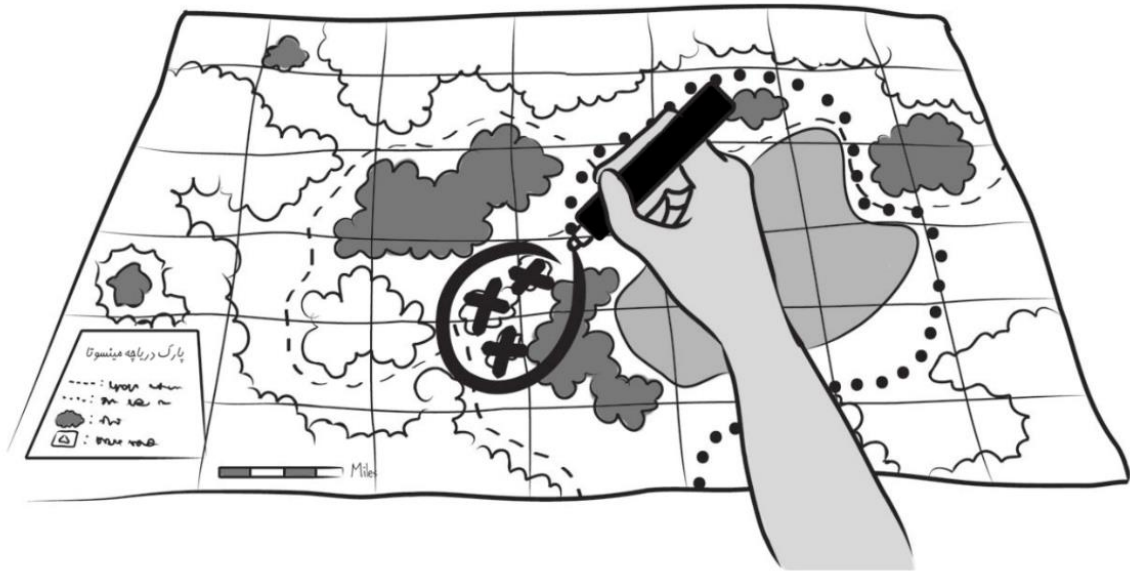
اداره امور یک شرکت از دیدگاه شما در دو سطح انجام می‌شوند. سطح اول: مدیر شما، که بیشتر وقتتان را با او سروکار دارید. و سطح دوم: سلسله‌مراتب اداری پشت مدیر شما که مدیران میانی، هیئت‌رئیس، تیم فروش، و کلا ... را شامل می‌شوند.

تجربه ایدئال یک کارمند

اگر مدیر شما یک رهبر خدمتگزار باشد که با به‌کارگیری اصول HRT عمیقاً برای پیشرفت کاری و حرفه‌ای شما تلاش کند، با به‌کارگیری تکنیک‌های زیر می‌توانید هم کار او را ساده‌تر کنید و هم در زندگی کاری خود به موفقیت‌های بیشتری برسید.



به دنبال مسئولیت‌های اضافه در کنار وظایف اصلی خود باشید. مثال موردعلاقه ما در مورد این موضوع، یک جنگلبان است که شاگرد خود را برای از بین بردن یک درخت پیر و آسیب‌دیده به جنگل می‌فرستد. شاگرد اگر فقط روی وظیفه‌ای که به او محول شده تمرکز کند، درخت مورد نظر را قطع کرده و سربلند و با افتخار باز می‌گردد. اما اگر با دید بازتری به مسئولیتش نگاه کند، درون جنگل می‌رود، درخت مورد نظر را قطع می‌کند، نقشه سایر درخت‌های پیر و آسیب‌دیده‌ای را که در مسیر مشاهده کرده تهیه می‌کند و به همراه یک طرح پیشنهادی برای از بین بردن آن‌ها باز می‌گردد تا با جنگلبان در راستای قطع آن‌ها نیز مشورت کند و اجازه بگیرد. در نتیجه این کار، در آینده هر زمانی که جنگلبان به دنبال شخصی می‌گشت تا کار دقیقی انجام دهد، به سراغ همین شاگرد خود می‌آمد، چرا که نه تنها از انجام کار مطمئن است، بلکه نگرانی و فشار فکری کمتری نیز متحمل می‌شود.



چنین حرکت فعالانه و مسئولانه‌ای، سنگینی بارِ وظایف مدیر شما را کاهش می‌دهد و باعث می‌شود حداقل یک موضوع کمتر برای نگرانی داشته باشد. این کار بیانگر توانایی شما در راستای قبول مسئولیت‌های بیشتر در کنار وظایف اصلی و تمایل برای خروج از حاشیه امن و امتحانِ مهارت‌های جدید نیز هست، به خصوص وقتی که ریسک و خطر کردن در تیم شما تشویق شود و افراد از اشتباه کردن ترسی نداشته نباشند.

خطر کنید و از شکست نترسید. در مورد اهمیت ریسک کردن و پذیرش سریع شکست در فصل‌های سوم و چهارم مفصلاً صحبت کردیم. در کنار یک رهبر روشن فکر، ناکامی و شکست سریع‌ترین راه برای یادگیری، درک محدودیت‌ها و رشد مهارت‌های شخصی است. یکی از دوستان ما، استیو هیمن، که دائماً به سفرهای کاری می‌رود معتقد است: «اگر حداقل یک بار در سال از پروازتان جا بمانید، همیشه زودتر از موعد مقرر در فرودگاه حاضر می‌شوید!» این استعاره به خوبی در مورد تولید هر نوع محصولی صدق می‌کند: اگر حداقل سالی یک بار شکست نخورید، یعنی به اندازه کافی ریسک و خطر نمی‌کنید. همانند موضوع قبول مسئولیت‌های بیشتر، ریسک کردن نیز بیانگر توانایی شما در قبول کارهای بزرگ‌تر و مهم‌تر است.

اگر در کار خود ریسک نکنید، کمتر شکست می‌خورید، ولی موفقیت‌های بزرگ کمتری نیز به دست می‌آورید. یک مدیر خوب دوست دارد تا اعضای تیمش محدودیت‌ها را کنار بزنند و ببینند چه کارهای دیگری را می‌توانند انجام دهند و در این راستا چیزهای جدیدی یاد بگیرند. وقتی در کاری با شکست مواجه می‌شوید، مسئولیت آن را به عهده بگیرید، به دنبال مقصر نشان دادن دیگران نباشید و تمام مراحل کار همراه با درس‌های آموخته شده و راه‌های جلوگیری از تکرار اشتباه را مکتوب کنید. سپس دوباره از نو به دنبال ریسک‌های جدید باشید.

بزرگسالانه رفتار کنید. توصیه‌ای دیگر که شاید واضح و مبرهن به نظر بیاید این است که شما خود مسئول طرز رفتار بقیه با خودتان هستید. مدیران بد با اعضای تیمشان مثل بچه‌ها رفتار می‌کنند. اگر در گذشته با چنین مدیرانی سروکله زده‌اید، احتمالاً به صورت پیش‌فرض رفتار مشابهی از سایر مدیران نیز انتظار خواهید داشت.

موضوعاتی را که به نظرتان مشکوک هستند به چالش بکشید. اگر مدیر شما تصمیمی اتخاذ می‌کند که فکر می‌کنید اشتباه است، از سؤال کردن و به چالش کشیدن او نترسید. هر چند ممکن است مخالفت شما مانع اجرای آن تصمیم نشود، ولی روحیه «بله‌قربان‌گویی» هیچ کمکی به مدیر شما نخواهد کرد تا رهبری بهتری ارائه کند.

مدیر شما علم غیب ندارد. دائماً مدیرتان را در جریان پیشرفت کارهایتان قرار دهید، قبل از آنکه لازم شود تا او از شما درباره وضعیت مسئولیت‌هایتان سؤال کند. وقتی در کارتان با مانعی برخورد می‌کنید یا موفقیتی به دست می‌آورید یا نیاز به کمکی دارید، یک پیام کوتاه برای مدیرتان ارسال کنید. بدین روش همیشه مطمئن خواهید بود که مدیر شما از کارهای شما باخبر است. گاهی بعضی از افراد از این حيله برای سروکله زدن با مدیرانی که در کوچک‌ترین کارها و مسئولیت‌ها دخالت‌های بی‌جا می‌کنند استفاده می‌کنند. اگر مدیر شما مدام از شما در مورد کارتان سؤال می‌کند، می‌توانید با ارسال مرتب گزارش کار، قبل از اینکه او فرصت پیدا کند تا مدام شما را سؤال پیچ کند، خود را از دست او آزاد کنید.

تمام این فنون و مهارت‌ها فقط زمانی که شما در یک شرکت با ساختار سازمانی کارآمد مشغول به کار هستید می‌توانند به خوبی جوابگو باشند، ولی وقتی شرایط شرکت ایدئال نیست، چطور؟

شرایطی که معمولاً وجود دارند

تمام خانواده‌های خوشحال مانند هم هستند، ولی هیچ دو خانواده غمگینی شبیه هم نیستند. لئو تولستوی، از کتاب آناکارینا راه‌های بی‌شماری برای یک سازمان یا شرکت وجود دارند تا جلوی موفقیت شما را بگیرند. این دست‌اندازها معمولاً در یکی از این دو دسته قرار می‌گیرند: «انسان‌های بد، یا سازمان‌های بد.»

مدیر بد

توصیف ویژگی‌های یک مدیر بد، آسان نیست. فیلم‌ها و سریال‌های تلویزیونی مختلفی در راستای نکوهش مدیران بد در دنیا ساخته شده‌اند. بیشتر ما در زندگی کاری‌مان این تجربه را داشته‌ایم تا حداقل با یک مدیر بد سروکار داشته باشیم. یک مدیر نالایق به راحتی می‌تواند زندگی بهترین تیم‌های مهندسی را به جهنم تبدیل کند. در این فصل ویژگی‌هایی از مدیران بد را بررسی می‌کنیم که روی کار مهندس‌های تیم بیشترین تأثیر منفی را دارند.

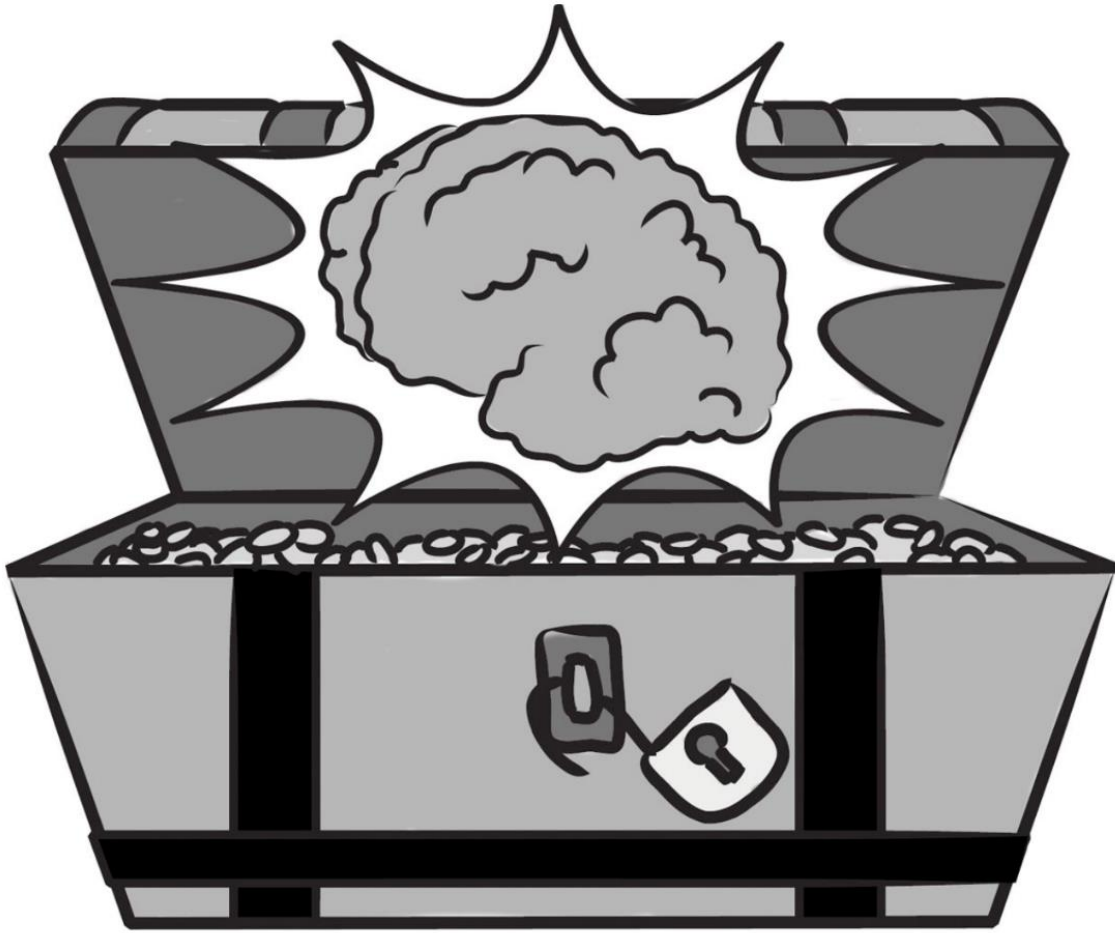
معمول‌ترین خصوصیت مدیران نالایق، ترس از شکست است. این حس ناامنی و عدم اعتمادبه‌نفس باعث می‌شود که در مدیریت بسیار محافظه‌کار باشند. این رفتار معمولاً در تضاد مستقیم با عملکرد یک مهندس و برنامه‌نویس است. اگر مدیر تیم به شما اجازه ریسک کردن ندهد، شما فرصتی برای پیاده‌سازی و یا امتحان کردن ایده‌هایتان نخواهید داشت و نهایتاً محصولی را پیاده‌سازی می‌کنید که شخص دیگری آن را طراحی کرده است.^۱

عدم اعتمادبه‌نفس یک مدیر باعث می‌شود که همیشه خودش را در تمام مکالمات و ارتباطات شما با افراد خارج از تیم دخیل کند. در نتیجه مانع صحبت مستقیم شما با سایرین و خارج از سلسله‌مراتب فرماندهی می‌شود. چنین مدیری، رابطه شما با دیگران و مخصوصاً با سایر مدیران را یک نوع نافرمانی و سرپیچی تلقی می‌کند. از شما توقع دارد که تمام ارتباطات شما از طریق او انجام شود تا اهمیت او برای همه روشن شده و قدرتش تثبیت شود.

حتماً در مدرسه شعار «توانا بود هر که دانا بود» را بارها شنیده‌اید. مدیر بد نیز با این شعار به خوبی آشناست، ولی با یک تفاوت کوچک: «او می‌خواهد تمام دانش را برای خودش نگه دارد و برای دانایی شما و در میان گذاشتن اطلاعات با شما اهمیتی قائل نیست؛ صرف‌نظر از اینکه گاهی اطلاعات بیشتر چقدر ممکن است کیفیت کار شما

۱. البته این هم روش قابل‌قبولی‌ست برای برنامه‌نویسی و توسعه نرم‌افزار. ولی ما فکر می‌کنیم که شاید این راه برای مهندسين خارق‌العاده روش جذابی نباشد.

را بالا ببرد. چنین مدیری اطلاعات را از شما مخفی می‌کند تا مطمئن شود که همیشه در جریان جزئیات کار است و در تمام تصمیم‌گیری‌ها حضور دارد. این اخلاق برای تثبیت قدرت مدیر بد مانع پیشرفت کار کل تیم می‌شود و به قیمت کم شدن سرعت توسعه نرم‌افزار تمام می‌شود.



مدیران نالایق با احتکار اطلاعات آن‌ها را تبدیل به تنها مجرای ارتباطی تیم می‌کنند که در نتیجه به آن‌ها این امکان را می‌دهد تا اعتبار و امتیاز موفقیت‌های شما را به اسم خودشان تمام کنند^۱، در حالی که اشتباهات و شکست‌های شما (و حتی گاهی شکست‌های خودشان) را نیز به پای شما بنویسند. بارها دیده شده که چنین مدیرانی به اعضای تیمشان صرفاً به عنوان ابزار و وسیله‌ای برای ارتقای شغلی خودشان نگاه می‌کنند. نه تنها هیچ اهمیتی برای پیشرفت کار، بلکه برای حس شادی و رضایت شغلی اعضای تیمشان نیز قائل نیستند.

۱. که به شدت کلافه‌کننده است؛ چرا که شما با وجود تمام دخالت‌ها و مزاحمت‌های آن‌ها کار را به موفقیت رسانده‌اید.

یکی از دوستان ما به اسم سوزان، چند سال با یک مدیر بد دست‌وپنجه نرم می‌کرد. مدیر سوزان بدون هیچ اطلاعات کمکی‌ای مسئولیت پروژه‌ها را به او می‌سپرد، هیچ توضیحی در راستای نحوه اجرای کار فراهم نمی‌کرد و هیچ شخص دیگری برای کمک یا اطلاعات بیشتر نیز معرفی نمی‌کرد. حتی اگر سوزان هیچ ایده‌ای از کار نداشت، این مدیر همچنان به این روش ادامه می‌داد. هدف او این بود که سوزان در طول مدت کارش و نیز برای برقراری ارتباط با سایر تیم‌ها به او وابسته باشد. با جلوگیری از هرگونه ارتباطی بین سوزان و افراد خارج از تیم اجازه نمی‌داد کسی متوجه شود که سوزان روی این پروژه کار می‌کند. بارها سوزان با وجود تمام دست‌اندازهای مدیرش موفق می‌شد به نحوی و هر طور شده نیازمندی‌های پروژه را درک کند و کار را به نتیجه برساند، ولی بعدها متوجه می‌شد که مدیرش اعتبار کار سوزان را به اسم خودش تمام کرده است.

در تضاد مستقیم با ویژگی‌های یک رهبر خدمتگزار که در فصل سوم بررسی کردیم، یک مدیر بد همیشه به دنبال این است که ببیند اعضای تیمش برای او چه کرده‌اند. او حتی با کم‌کاری‌های اعضای تیم نیز تا وقتی که تیم را به قهقرا نبرده‌اند، برخوردی نمی‌کند، چرا که این کار زحمت زیاد و امتیاز شخصی کمی دارد. همه چیز در این سؤال خلاصه می‌شود: آیا مدیرتان شما را نجات می‌دهد یا شما مدیرتان را نجات می‌دهید؟

همکار سیاستمدار

هر چند که ما همیشه طرفدار اعتماد به آدم‌ها هستیم، اما اطمینان به یک فرد سیاستمدار می‌تواند دشوار و خطرناک باشد.

یک شخص سیاستمدار به سادگی و در نگاه اول قابل تشخیص نیست، چرا که چنین فردی مهارت بالایی در مدیریت روابط انسانی دارد و شاید ابتدا خیلی دوستانه و صمیمی به نظر بیاید. معمولاً به خوبی می‌تواند از همکاران و اعضای تیمش برای منفعت و پیشرفت شخصی سوءاستفاده کند. همیشه به راحتی تقصیر را گردن دیگران می‌اندازد و به سرعت موفقیت‌ها را به نام خودش تمام می‌کند. به ندرت به طور مستقیم با شما درگیر می‌شود، برعکس معمولاً دوستانه رفتار می‌کند و حرف‌هایی را می‌زند که شما دوست دارید بشنوید. اگر موفق نشود از شما سوءاستفاده کند، یا شما را نادیده می‌گیرد یا به شما به چشم یک تهدید نگاه می‌کند و تلاش می‌کند شما را از سر راه خود بردارد. پس از مدتی همکاری با چنین افرادی، تشخیص آن‌ها راحت‌تر می‌شود. یک فرد سیاستمدار بیشتر ادای کار کردن و تأثیرگذاری را درمی‌آورد تا اینکه واقعاً تأثیرگذار باشد.

توصیه ما این است که تا آنجا که می‌توانید از افراد سیاستمدار دوری کنید، ولی پشت سر آن‌ها پیش سایر همکاران و افراد شرکت بدگویی نکنید، چرا که هیچ‌وقت نمی‌توان مطمئن شد چه کسی او را شناخته و چه کسی فریب او را خورده است. اگر شما فردی هستید که دوست دارد سرش در کار خودش باشد و فقط روی توسعه محصولات خلاقانه و جذاب تمرکز کند، نزد افراد سیاستمدار باید در این استراتژی تجدیدنظر کنید. اگر برای ارتقای کاری تلاش نکنید، چون دوست ندارید که درگیر «بازی‌های شرکت» شوید، ممکن است ببینید که یک فرد با سیاست به جای شما ترفیع گرفته است. در این حالت علاوه بر همکار سیاستمدار با یک مدیر بد نیز روبه‌رو هستید. جلوتر در بخش اداره روابط سازمانی درباره این موضوع صحبت خواهیم کرد.

سازمان بد

با رشد شرکت، کاغذبازی و فرایندهای اداری در تلاش برای مدیریت سود، کاهش ریسک و افزایش قابلیت پیش‌بینی امور، به مرور بیشتر و بیشتر می‌شوند و بار زیادی را روی سازمان می‌گذارند. چنین تشریفات و مقرراتی کم‌کم می‌توانند تا اندازه‌ای زیاد شوند که مانع پیشرفت و موفقیت شرکت شوند. همانند مدیران بد، درباره سازمان‌های بد نیز اطلاعات زیادی نوشته شده است. در این بخش ما به بررسی آن دسته از ویژگی‌های بد سازمانی می‌پردازیم که می‌توانند مستقیماً روی عملکرد اعضای تیم تأثیر منفی بگذارند.

واقعیت ساده و انکارناپذیر این است که بیشتر شرکت‌ها «مهندسی‌محور» نیستند؛ به این معنا که مهندسی صرفاً ابزار و وسیله‌ای هستند برای احقاق اهداف و مقاصد غیرفنی شرکت. در نتیجه، اداره‌کنندگان امور درک صحیحی از مسائل فنی و مهندسی ندارند و تمرکز آن‌ها روی نیازهای تجاری شرکت است. بنابراین نیازهای غیرمعمول و توقعات غیرواقعی از تیم مهندسی دارند. حتی اگر فردی با دانش فنی مناسب عضو هیئت‌رئیسۀ چنین شرکتی شود و از تیم مهندسی دفاع کند، به مرور با افرادی که سلامتی و رضایت کارمندان را فدای نیازهای تجاری شرکت می‌کنند، جایگزین می‌شود. این طرز فکر رهبران شرکت، خود را به صورت ضرب‌الاجل‌های غیرمنطقی و کمبود نیروهای متخصص و ماهر برای اتمام پروژه‌ها نشان می‌دهد. در چنین شرکت‌هایی شما وقت زیادی را برای درخواست یک قطعه سخت‌افزاری اتلاف می‌کنید و تیم شما مجبور می‌شود هفته‌ها درگیر توسعه پروژه‌ای باشد که می‌توانست با هزینه خیلی کمی خریداری شود. این اتفاق مدام در شرکت‌هایی که با مهندسی خود به چشم یک «واحد کار» یا صرفاً «منبع انسانی» نگاه می‌کنند تکرار می‌شود و آن‌ها را در تصمیم‌گیری‌های مهم دخیل نمی‌کنند.

بدترین نوع سازمان‌ها آن‌هایی هستند که ساختار فرمان و اجرا را مانند حکومت‌های ملوک‌الطوایفی تعریف کرده‌اند. چند سال پیش یکی از دوستان ما به نام ترنس در شرکتی کار می‌کرد که قواعد سفت و سختی برای گزارش ایرادهای نرم‌افزاری بین تیم‌ها داشت. یکی از تیم‌ها با تغییری که در کد ایجاد کرده بود باعث شده بود تا قسمت کد ترنس دچار مشکل شود و هر چند ساعت با مشکل کمبود حافظه مواجه شود. به جای آنکه مستقیماً به تیم مسئول ایمیل بزند، یا به تاریخچه کدهای آن‌ها نگاه کند، ترنس تمام شب را بیدار ماند و تا جایی که می‌توانست درباره این ایراد و مشکل اطلاعات مختلفی جمع کرد تا پرونده محکمی علیه این تیم تهیه کند. نهایتاً تمام اطلاعات جمع‌شده را برای مدیرش ارسال کرد، مدیرش به رئیس بخش ایمیل زد، رئیس بخش به رئیس بخش تیم مسئول ایمیل زد، رئیس بخش تیم مسئول به مدیر تیم مسئول ایمیل زد و مدیر تیم شخصی که درباره آن تکه از نرم‌افزار اطلاعات لازم را داشت شناسایی کرد. بیش از ده روز بعد، ترنس در یک جلسه کنار دو مدیر، دو رئیس بخش و سه مهندس دیگر نشست و درباره این ایراد نرم‌افزاری صحبت کردند و بررسی کردند که آیا می‌توانند این مشکل را قبل از انتشار نسخه بعدی برطرف کنند یا خیر. احمقانه است؟ متأسفانه چنین اتفاقی خیلی معمول است. از سوی دیگر وقتی فیتز به تازگی به گوگل پیوسته بود، همان هفته اول یک اشتباه

تایپی در کد جی. میل پیدا کرد، آن را اصلاح کرد و برای تیم جی. میل فرستاد. آن‌ها نیز از او بابت کمکش تشکر کردند.

بسیاری از شرکت‌ها مملو از آدم‌هایی هستند که روی سلسله‌مراتب اداری وسواس عجیبی دارند.^۱ در نتیجه مبارزه بر سر قدرت بسیار معمول است و مدیران برای حفظ نیروهای ارزشمند خود از انتقال اعضای تیمشان به سایر تیم‌ها جلوگیری می‌کنند، حتی اگر چنین انتقالی به نفع اهداف شرکت و خود عضو تیم باشد.

آیا تاکنون شرکتتان با شما مانند یک کودک شرور رفتار کرده است؟ شرکت با یک فایروال سفت و سخت دسترسی شما به بعضی از وبسایت‌ها را محدود کرده است؟ شما را مجبور می‌کند که تمام ساعات و دقایق کاری روزتان را با دقت ثبت کنید؟ بعضی از سازمان‌ها کارایی شما را با معیارهای بی‌معنی و بسیار نادرست نظیر تعداد خطوط کدهای نوشته‌شده در هفته اندازه‌گیری می‌کنند.^۲



هنوز کارمندانی وجود دارند که معیار موفقیتشان به جای اینکه تعداد محصولاتی که تولید کرده‌اند باشد، تعداد جلساتی است که به آن دعوت می‌شوند. شرکت شما ممکن است در اصول بسیار مهمی نظیر تمرکز، مسیر و هدف (Vision) مشکلات جدی داشته باشد. چنین کمبودهایی معمولاً در نتیجه تعداد بیش از اندازه عناصر

۱. علاوه بر این، در بسیاری از شرکت‌های به‌هم‌ریخته، افراد بیشتر درگیر عناوین شغلی هستند تا کارایی و رضایت خودشان.

۲. اصولاً نباید تعداد خطوط کد برداشته‌شده از کد اضافه‌شده ارزشمندتر باشند؟

تصمیم‌گیرنده است. وقتی کمیته‌ها به جای افراد تصمیم می‌گیرند، دستورات متناقض افزایش یافته و افراد در حلقه‌های کوچک‌تر به دنبال اهداف خودشان حرکت می‌کنند، به جای اینکه همه در جهت مسیر کلی شرکت گام بردارند.

تخلفاتی که مطرح کردیم، شرکت‌های زیادی انجام می‌دهند و تخلفات زیاد دیگری هم وجود دارد که در کنار این‌ها مرتکب می‌شوند. ولی با وجود این، تمام این شرکت‌ها بالاخره مجموعه‌ای از انسان‌ها هستند و برای اینکه بتوانید انسان‌ها را وادار کنید تا به شما کمک کنند نکات و ترفندهایی را می‌توانید به کار بگیرید.

اداره روابط سازمانی

اینجا یک محیط شبیه‌سازی شده برای دعواست! تمام اینجا مشابه محیط ماتریکس برنامه‌نویسی شده. قوانین اصلی مثل جاذبه زمین کاملاً شبیه ماتریکس هستند. چیزی که باید یاد بگیری این است که این قواعد مثل قوانین کدگذاری شده در یک سیستم کامپیوتری هستند. بعضی از آن‌ها را می‌توان خم کرد و بعضی را می‌توان زیر پا گذاشت. فهمیدی؟ حالا اگر می‌توانی من رو بزن!

«مورفیوس در فیلم ماتریکس»

دقیقاً مثل محیط شبیه‌سازی شده مبارزه، شرکت‌ها نیز از دستورات و قواعد مختلفی ساخته شده‌اند. بعضی از این قوانین را می‌توان خم کرد و بعضی را می‌توان زیر پا گذاشت. اگر تمرکز شما روی حالت ایدئال و قواعدی باشد که «باید» در شرکت انجام شوند، معمولاً ناامید و مستأصل خواهید شد. عیب و نقص‌ها را بپذیرید و تلاش کنید تا مابین ساختار و سلسله‌مراتب شرکت راه و روشی را پیدا کنید که به وسیله آن بتوانید کارهای خودتان را پیش ببرید و به اهدافتان برسید. صرف‌نظر از اینکه سازمانی که در آن کار می‌کنید خوب باشد یا بد، معتقدیم که استراتژی‌های زیر به شما برای پیش بردن کارهایتان کمک خواهند کرد.

عذرخواهی از اجازه گرفتن راحت تر است^۱

پیش از هر چیزی به دنبال این باشید که ببینید تا کجا می‌توان از بین مقررات شرکت قسر در رفت. اگرچه اجازه گرفتن از سایرین به شما کمک می‌کند که مسئولیت تصمیم‌گیری را به دوش شخص دیگری بیندازید، ولی در مقابل به آن شخص فرصت «نه» گفتن را نیز می‌دهید. دقت کنید چه کارهایی را می‌توان بدون تأییدیه دیگران در شرکت انجام داد. توصیه ما این است که هر جا ممکن بود، کاری را انجام دهید که فکر می‌کنید به سود اهداف شرکت است.

حتی اگر کاملاً خود را آماده کرده‌اید تا برای بخشش التماس کنید، موضوعاتی که ارزش چنین مبارزه‌ای را دارند با دقت انتخاب کنید. هر بار شما در شرکت ادعایی علیه یا در مقابل فرد دیگری مطرح کنید از ذخیره سرمایه قدرت سیاسی خودتان هزینه می‌کنید. اگر سرمایه سیاسی خود را خرج امور کم‌اهمیت کنید، برای مسائلی که ارزش بیشتری دارند اندوخته‌ای نخواهید داشت. برای اموری مبارزه کنید که یا برایتان بیشترین اهمیت را دارند و یا احتمال می‌دهید که می‌توانید در آن‌ها پیروز شوید. مصرف کردن سرمایه سیاسی‌تان برای مبارزاتی که در آن‌ها شانس برای برنده شدن ندارید، یک کار بیهوده و استرس‌آور است که جلوی پیشرفت کاری شما را می‌گیرد. در بخش «حساب بانکی قدرت سیاسی شما»، در این باره بیشتر صحبت می‌کنیم.

اگر مسیر «معذرت‌خواهی» را انتخاب می‌کنید، داشتن همکارانی که با شما و ایده‌هایتان هم‌نظر هستند بسیار مفید خواهد بود. مخصوصاً اگر ایده‌های شما ریسک و خطر بالایی داشته باشند. بهتر است که این همکاران هم ایده خوبی از کارهایی که در شرکت می‌توان بدون اجازه انجام داد و قسر در رفت نیز داشته باشند.

۱. این جمله را به گریس هاپر، یکی از توسعه‌دهندگان زبان برنامه‌نویسی COBOL نسبت داده‌اند.

اگر راه را پیدا نمی‌کنید، راهی بسازید



یک روش دیگر برای ایجاد تغییر در یک سازمان این است که آدم‌های زیادی را درباره ایده‌هایتان قانع کنید. اگر لازم است که تعدادی از همکارانتان با شما هم‌عقیده شوند، کاغذبازی‌ها و فرایندهای اداری نمی‌توانند جلوی عملی شدن ایده‌های شما را بگیرند و مدیریت به ناچار مجبور می‌شود آن‌ها را به رسمیت بشمارد. ممکن است ایده‌های شما را قبول کنند یا در مقابل آن (با صرف بودجه‌ای از سرمایه سیاسی خود) بایستند. این ترفند طی سال‌های گذشته به مهندسين بسیاری کمک کرده تا نظراتشان را عملی کنند. مثلاً به این روش توانسته‌اند مدیریت را قانع کنند که اجازه دهند از نرم‌افزارهای متن‌باز برای آسان‌تر کردن کارها در پروژه استفاده کنند.

متقاعدسازی به کمک دیگران

یک روش برای افزایش احتمال موفقیت در متقاعد کردن یک شخص این است که آدم‌های دیگری را با خود هم‌نظر کنید و کاری کنید تا یکی از آن‌ها ایده یا پیشنهاد شما را با آن شخص مطرح کند. حتی اگر شخص مورد نظر از این نقشه شما به خوبی مطلع باشد، طبق اصول ابتدایی روان‌شناسی انسان، احتمال تأثیرگذاری پیشنهاد شما افزایش می‌یابد، چرا که این ایده از طرف آدم‌های متعددی مطرح می‌شود.

اصولاً موضوع ایده‌پردازی، عجیب خارق‌العاده است: یک ایده به راحتی می‌تواند به موفقیت برسد، اگر بحث مالکیت ایده و اعتباردهی به یک شخص خاص برای بیان آن مطرح نباشد! گاهی می‌بینید که آدم‌ها فقط در صورتی حاضرند یک ایده را پیگیری کنند که در اعتبار طرح ایده سهمی داشته باشند. بنابراین باید تصمیم بگیرید

که چه چیزی برای شما اهمیت بیشتری دارد: اخذ اعتبار برای طرح یک ایده یا انتشار هرچه بیشتر موضوع ایده. با وجود اینکه احتمالاً بسیار سخت خواهد بود که ببینید شخصی ابتکار و پیشنهاد شما را به نام خودش ثبت کرده و مطرح می‌کند، این روش سریع‌ترین راه برای عملی شدن ایده شماست. به این فکر کنید که در این حالت ایده شما به جای اینکه در نطفه خفه شود، به گوش تعداد بیشتری از آدم‌ها خواهد رسید.

حذف عادات بد در یک سازمان، درست مثل عادات بد انسان‌ها سخت و دشوار است. یکی از معلمان قدیمی بن حرف جالبی می‌زد: «حذف یک عادت بد غیرممکن است، مگر آنکه عادت بد را با یک عادت خوب جایگزین کنید.» هر آدمی که برای ترک سیگار تلاش کرده باشد، به خوبی با این قضیه آشناست. شرکت‌ها و سازمان‌ها هم همین طور هستند. اگر می‌خواهید یک فرایند و عادت بد را از بین ببرید، یک عادت بهتر را جایگزین آن کنید. یک جلسه هفتگی را دوست ندارید؟ آن را با یک جلسه متفاوت یا یک نوع ارتباط جمعی دیگر جایگزین کنید. از مراحل یک گزارش نویسی ناراضی هستید؟ ناله و شکایت نکنید، گزارش بهتری بنویسید که به قدری خوب باشد تا هیچ‌کس نتواند به آن بی‌توجهی کند. وقتی عادت جایگزین را پیدا کردید، زمان این می‌رسد تا با نیروی اینرسی دائمی که همیشه جلوی تغییر را می‌گیرد مبارزه کنید. توصیه ما این است که از تکنیک «پیشنهاد امتحان کردن» استفاده کنید. بدین صورت که به سایرین پیشنهاد کنید تا فرایند جایگزین را چند هفته به صورت آزمایشی امتحان کنند تا ببینند چطور است. این روش باعث می‌شود که تغییرات جدید ابتدا کمتر دائمی به نظر برسند و کمتر ترسناک باشند. وقتی همه از روش جدید راضی باشند، فراموش خواهند کرد که این عادت جدید را به صورت آزمایشی انجام می‌دادند.

مدیریت فرا تیمی

صرف‌نظر از اینکه عضو یا رهبر تیم باشید، بخشی از وقتتان را باید برای مدیریت مسائل خارج از تیمتان اختصاص دهید. این کار به این معنی است که باید تلاش کنید تا هم مدیرتان و هم سایر افراد خارج از تیم شما نه تنها از کارهایی که انجام می‌دهید باخبر باشند، بلکه اطمینان دارند که کارهایتان را به خوبی انجام می‌دهید. لازم است که خودتان را به دیگران اثبات کنید و قابلیت‌هایتان را به خوبی نشان دهید.

در فصل ششم توضیح خواهیم داد که در تعهدات خود تا جایی که امکان دارد قول‌های کم و محدودی بدهید، ولی در پایان نتیجه‌ای بیش از آنچه وعده داده بودید تحویل دهید. منظور ما این نیست که برآورد زمانی کارها را بی‌دلیل زیاد کنید و ضرب‌الأجل کارها را بی‌اساس عقب بیندازید؛ بلکه توصیه ما این است که تا حد امکان در مورد کارهایی که احتمالاً نمی‌توانید تهیه و آماده کنید، تعهد ندهید، حتی اگر این به معنی «نه» گفتن‌های بسیار باشد. اگر دائماً مهلت‌ها را از دست بدهید و در تعهداتتان بدقولی کنید، همکارانتان دلیلی برای اعتماد کردن به شما پیدا نمی‌کنند و برای درخواست‌های بعدی به سراغ افراد دیگری می‌روند.

توصیه می‌کنیم که وقت و انرژی خود را بیشتر به انتشار محصولات جدید اختصاص بدهید تا هر چیز دیگری. تولید محصول بیش از هر چیزی در شرکت به شما اعتبار و شهرت می‌دهد و ذخیره سرمایه سیاسی شما را افزایش می‌دهد. هر چقدر که رفع ایراد برنامه‌ها و تمیزکاری کد و سوسه‌برانگیز باشد، تقریباً هیچ‌وقت برای افراد خارج از تیمتان ارزشی نخواهد داشت. در چنین حالتی در موقعیت خجالت‌آوری قرار می‌گیرید که عملاً چیزی برای نمایش ندارید که از لحاظ سیاسی در شرکت حائز اهمیت باشد.^۱ در این حالت نه تنها اعتبار خود را از دست می‌دهید، بلکه شاید حتی پروژه محصول شما منحل شود.

کارهای «تهاجمی» و کارهای «تدافعی»

آن اوایل که بن مدیر شده بود، تیم او زیر بار بدهی‌های فنی زیادی در کد و پروژه قرار گرفته بود. او اولویت اصلی تیم را تا مدت‌ها تمیزکاری کدها و رفع اشکالات برنامه قرار داد. مدیران بالاسری او مخالفتی با برنامه‌های او نکردند و اجازه دادند تا بن و تیمش کارشان را شروع کنند. با وجود این تأییدیه ابتدایی، پس از مدتی مدیران بن از عملکرد تیم ناراضی به نظر می‌آمدند و نگران این بودند که چرا تیم «هیچ کاری انجام نمی‌دهد؟» ولی در واقعیت

۱. منظور ما این نیست که فعالیت‌های پیشگیری‌کننده نظیر رفع ایراد و تمیزکاری کد مهم نیستند، بلکه می‌گوییم چنین کارهایی به سختی می‌توانند افراد خارج از تیم و مدیریت بالاتر را تحت تأثیر قرار دهند.

تیم بن کارهای زیادی را انجام داده بودند و موفق شده بودند وضعیت پروژه و کد را به مراتب بهتر و تمیزتر کنند. ولی به نظر می‌آمد که چنین کارهایی هیچ تأثیری از لحاظ روانی روی افراد خارج از تیم نمی‌گذارد و برای سایرین بیشتر کسل‌کننده است.

بعد از چنین تجربه بدی، بن کارهای تیم را به دو دسته «تهاجمی» و «تدافعی» تقسیم کرد. کارهای تهاجمی آن دسته از فعالیت‌هایی هستند که موجب افزایش یک قابلیت درخشان جدید در محصول می‌شوند که برای کاربران قابل تشخیص و قابل رؤیت هستند. چنین کارهایی به سادگی می‌توانند کاربران و سایر افراد خارج از تیم را هیجان‌زده کنند (مثلاً بهتر کردن رابط گرافیکی و یا بیشتر کردن سرعت برنامه). کارهای تدافعی ولی تلاش‌هایی هستند که کمک می‌کنند تا نرم‌افزار در طولانی‌مدت سلامتی و توانایی بیشتری داشته باشد؛ مثلاً افزایش کارایی کد از طریق تغییر الگوهای پایگاه داده، دوباره‌نویسی یک قابلیت نرم‌افزار، انتقال داده‌ها و یا توسعه مراحل بازبینی کارهای تدافعی، به استحکام و پایداری نرم‌افزار کمک می‌کنند و اگرچه بسیار مهم و اساسی هستند، معمولاً هیچ اعتبار و توانایی سیاسی‌ای بابت آن‌ها کسب نمی‌کنید. اگر تمام وقت خود را صرف کارهای تدافعی کنید، همه فکر می‌کنند که توسعه نرم‌افزار محصول تیم شما متوقف شده است. به هر حال باور ما از وضعیت یک چیز، بیشتر به درک و احساس و نحوه تفکر ما درباره آن بستگی دارد.

شانس و محبت کردن

صرف نظر از نوع شرکتی که در آن کار می‌کنید، گاهی شاید ساخت موقعیت و افزایش شانس و اقبال خیلی دشوار نباشد. ریچارد ویزمن، نویسنده کتاب «عامل شانس»^۱، آزمایشی درباره قدرت افراد برای فرصت‌یابی و شناسایی بخت و اقبال انجام داده است^۲:

به هر دو گروه آدم‌های خوش‌شانس و بدشانس یک روزنامه دادم و از آن‌ها خواستم بگویند تعداد عکس‌های چاپ‌شده در روزنامه چقدر است. آدم‌های بدشانس به طور متوسط دو دقیقه صرف شمارش عکس‌ها کردند، در حالی که آدم‌های خوش‌شانس بعد از چند ثانیه آماده پاسخ‌گویی بودند. چرا؟ چون در صفحه دوم روزنامه یک پیام بزرگ با حروف درشت که نیمی از صفحه را اشغال کرده بود وجود داشت که این‌طور نوشته بود: «نیازی به شمارش نیست. این روزنامه ۴۳ عکس دارد.» جواب سؤال من دقیقاً روبه‌روی چشمان شرکت‌کننده‌ها بود، ولی آدم‌های بدشانس عموماً به آن توجهی نداشتند؛ در حالی که خوش‌شانس‌ها آن را دیدند.

او در ادامه توضیح می‌دهد که آدم‌های خوش‌شانس مهارت پیدا کردن فرصت‌ها و موقعیت‌هایی را دارند که شانس و اقبال آن‌ها را افزایش می‌دهد. ما معتقدیم که این نظریه در مورد کارمندان یک شرکت نیز صدق می‌کند. اگر تمرکز شما تمام و کمال روی شغل و وظایف خودتان و در جهت تمام کردن تکالیف خودتان باشد و نه هیچ چیز دیگر، شانس این را ندارید که موقعیت‌ها و فرصت‌های خوب را پیدا کنید. ولی اگر هر زمانی که امکانش را داشتید به همکارانتان در کارهایشان کمک کنید، معمولاً آن‌ها نیز در فرصت مناسب محبت شما را پاسخ می‌دهند. البته هیچ‌وقت هنگام کمک و محبت کردن نباید فکر کنید که شخص مقابل وظیفه دارد کمک شما را جبران کند و انتظار داشته باشید که روزی محبت شما را جواب دهد.

1. The Luck Factor - Published by Miramax - ISBN: ۹۷۸-۱۴۰۱۳۰۹۴۱۶

2. http://bit.ly/luck_skill

حساب پس انداز سیاسی — اجتماعی شما

در هر شرکتی یک بازار مخفی بر پایهٔ محبت و ادای دین بین افراد وجود دارد که هیچ ارتباطی با ساختار سلسله‌مراتبی شرکت ندارد. لطف و کمک شما به همکاران، اصلی‌ترین منبع برای افزایش حساب پس انداز سیاسی — اجتماعی شماست. معمولاً کار یا مشکلی وجود دارد که اگرچه مسئولیت آن به عهدهٔ شخص دیگری است، ولی انجام آن می‌تواند کمک بزرگی باشد. بهتر است همیشه به دنبال فرصتی باشید تا به دیگران کمک کنید. بیشتر اوقات این خود افراد هستند که پیش شما می‌آیند و از شما بابت کاری درخواست کمک می‌کنند. انجام چنین محبت‌هایی هر بار کمی اعتبار به حساب پس انداز سیاسی — اجتماعی شما اضافه می‌کند. به چنین اعتباراتی به چشم شرط‌بندی نگاه کنید. بعضی افراد به شما احساس دین می‌کنند و محبت شما را پاسخ می‌دهند و گاهی عده‌ای محبت شما را فراموش می‌کنند. حتی گاهی اوقات بعضی از همکارانتان محبت شما را چندین برابر بیشتر و بزرگ‌تر جبران می‌کنند. از شما به عنوان شخصی یاد می‌کنند که در زمان گرفتاری به آن‌ها کمک کردید. با مرور زمان وقتی یک روز شما در کاری نیاز به کمک داشتید، آن‌ها حاضر و آماده‌اند تا به شما کمک کنند. حتی اگر هیچ‌وقت محبت شما را پاسخ ندهند، هنگام کمک به آن‌ها، شما تجربهٔ جدیدی کسب کردید و احتمالاً موضوع جدیدی در مورد کار آن‌ها یاد گرفته‌اید. گذشته از همهٔ این صحبت‌ها، کمک و محبت به دیگران، عامل حس رضایت و شادی در شما خواهد بود. هنگام کمک کردن، به غیر از اندکی زمان و تلاش، چه چیز دیگری را از دست می‌دهید؟

وقتی برای کاری نیاز به کمک دیگران دارید می‌توانید از همان حساب پس انداز سیاسی — اجتماعی خود برداشت کنید. شاید لازم باشد از کسی بخواهید لطفی در حق شما انجام دهد یا ممکن است به اشتباه در کار شخص دیگری دخالت کرده باشید، یا حتی شاید در موقعیتی صرفاً با یکی از همکارانتان اختلاف نظر داشته باشید. سعی کنید مهارت تشخیص اینکه چه زمانی به حساب پس انداز سیاسی — اجتماعی خود اضافه می‌کنید و چه زمانی از آن برداشت می‌کنید را به دست بیاورید. در غیر این صورت، به احتمال زیاد حساب پس انداز شما به زودی خالی خواهد شد و در شرکت هیچ قدرت و نفوذی نخواهید داشت.

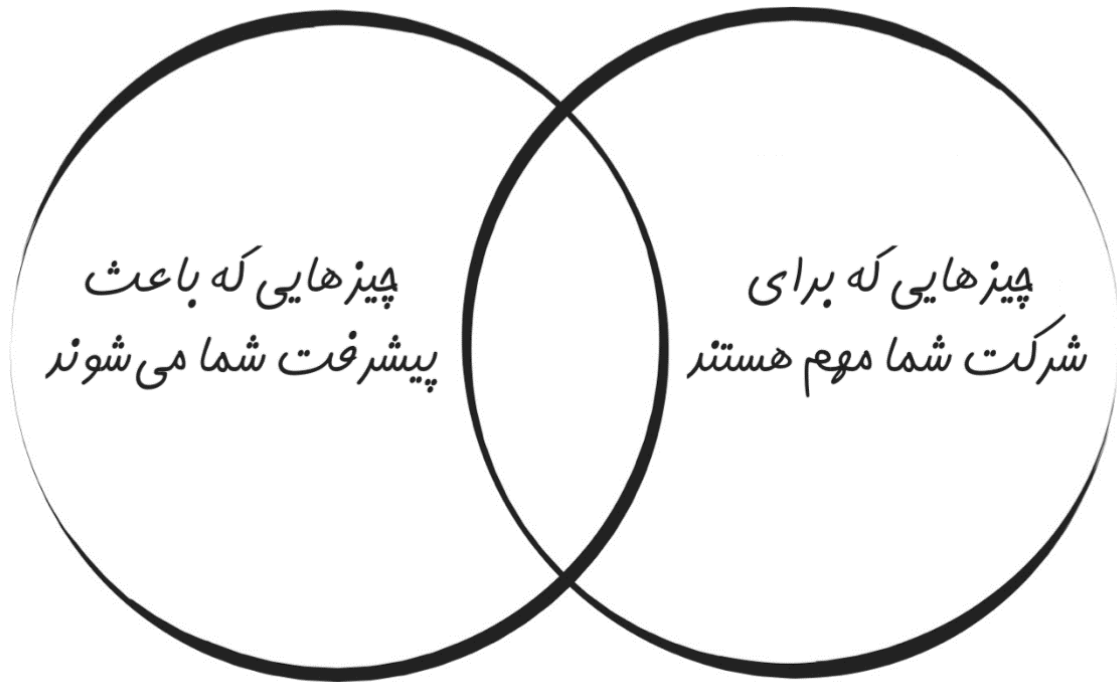


یکی از نکات جالب حساب پس انداز سیاسی - اجتماعی این است که این حساب با خروج شما از شرکت خالی نخواهد شد. هنوز می‌توانید برای درخواست کمک از همکاران قدیمی به سرمایه سیاسی - اجتماعی خود اتکا کنید. دقیقاً به همین دلیل است که هنگام خروج از شرکت نباید تحت هیچ شرایطی پل‌های پشت سر را خراب کرد، اگرچه ممکن است بسیار وسوسه‌کننده به نظر بیایند.^۱

۱. بیشتر صنایع از آنچه که فکر می‌کنید بسیار کوچک‌تر هستند و آدم‌ها بیشتر از چیزی که فکر می‌کنید با همدیگر صحبت می‌کنند. بنابراین فردی که امروز با آن مشکل دارید ممکن است همان فردی باشد که ده سال بعد فرم تقاضای کار شما را بررسی می‌کند. خراب کردن پل‌های پشت سر تقریباً هیچ‌وقت فکر عاقلانه‌ای نیست، مگر آنکه برنامه شما این باشد که به یک جزیره دورافتاده نقل مکان کنید و سبدهای حصیری ببافید. دوست می‌آید و می‌رود، ولی دشمنان روی هم جمع می‌شوند.

به یک جایگاه امن ارتقا پیدا کنید

بیشتر مهندسين فكر مي كنند كه براي كسب ترفيع و ارتقاي كاري، تنها چيزي كه لازم است اين است كه وظايف خود را به نحو احسن انجام دهند. متأسفانه شركت‌هاي كمي اين‌گونه هستند و در بيشتر سازمان‌ها و شركت‌ها، علاوه بر اينكه بايد وظايف خود را به بهترين وجه انجام دهيد، براي گرفتن ترفيع بايد كمي در راستاي «ارتقا و پيشرفت» نقش بازي كنيد و تلاش مضاعف داشته باشيد.



اگر از جايگاه و موقعيت خود در كار راضي باشيد، شايد ترجيح دهيد كه خود را درگير بازي‌هاي ارتقا و ترفيع نكنيد، ولي احتمالاً با اين كار وضعيت شما آسيب‌پذير خواهد شد. مثلاً ممكن است ساختار اداري شركت تغيير كند و شما به تيمي منتقل شويد كه رهبر خوبي نداشته باشد يا درگير شخص سياست‌مدار شركت شويد.

هرچه جايگاه شما در شركت بالاتر برود (چه به عنوان عضو تيم و چه به عنوان رهبر تيم)، روي سرنوشت كاري خود كنترل بيشتري خواهيد داشت. كمي تلاش براي كسب درجه بالاتر و ارتقاي شغلي، وقتي كه حتي از وضعيت اكنون خود راضي هستيد، بهترين سرمايه‌گذاري در شادي و امنيت كاري شماست. دستاوردهاي خود را يادداشت

کنید تا از آن‌ها در ارزیابی‌های شخصی خود استفاده کنید. رزومه خود را به روز نگه داشته^۱ و با مدیر خود به اشتراک بگذارید. فرایند و پیش‌نیازهای ترفیع در شرکت را مطالعه کنید و با مدیر خود درباره کارهایی که لازم است برای به دست آوردن ارتقای کاری انجام دهید، شفاف صحبت کنید. حتی اگر در ترفیع کاری هیچ قطعیتی وجود نداشته باشد، هنوز برای بالا بردن احتمال آن، کارها و تلاش‌های زیادی می‌توان انجام داد.

۱. برخلاف توصیه‌های عمومی ما در این کتاب، متن رزومه شما دقیقاً جایی است که شما باید غرور خود را به کار بگیرید و از فروتنی خودداری کنید. ما طرفدار خیال‌پردازی در رزومه نیستیم، ولی توصیه می‌کنیم که توانایی‌ها و قابلیت‌های خودتان را با صدای بلند فریاد بزنید.

به دنبال دوستان قدرتمند باشید

هر شرکتی یک سلسله‌مراتب نانوشته و در سایه دارد که قدرت و تأثیرگذاری افراد را نشان می‌دهد. آدم‌ها در این سلسله‌مراتب به دسته‌های مشخصی تقسیم می‌شوند.

«متصل‌کنندگان» افرادی هستند که آدم‌های زیادی را در گوشه و کنار سازمان می‌شناسند و می‌توانند شخص مناسب برای هدف شما را معرفی کنند. برای انجام بعضی از کارها فقط کافیست یک فرد خاص را پیدا کرد و متصل‌کنندگان دقیقاً همکارانی هستند که می‌توانند به شما در این راستا کمک کنند.

«کهنه‌کاران» آن دسته از افراد قدیمی در شرکت هستند که شاید مقام بالایی نداشته باشند، ولی به دلیل سابقه کار زیادشان، دانش و تجربه بالایی در مورد نوع کارهای سازمان دارند. در نتیجه، چنین افرادی قابلیت تأثیرگذاری بالایی در امور شرکت خواهند داشت. وقتی می‌خواهید از پیچ‌وخم فرایندهای شرکت سر درآورید یا برای ایده خود نیازمند یک حامی هستید که مورد احترام همه باشد، بهترین کار این است که سراغ کهنه‌کاران شرکت بروید.

بیشتر افراد معمولاً این موضوع را به شوخی بیان می‌کنند، ولی واقعیت این است که منشی‌ها و دستیاران اداری قدرت زیاد و توانایی تأثیرگذاری بالایی در سازمان‌ها دارند. آن‌ها معمولاً نماینده مدیران و هیئت‌رئیس هستند و نقشی کلیدی در راستای هماهنگی امور و اداره کارها دارند. بنابراین با مسئولیت خود (و با به خطر انداختن زندگی کاری خود) با آن‌ها بد رفتار نکنید. هیچ فرصتی را برای خوشحال کردنشان از دست ندهید. زیربنای اقتصاد مبتنی بر لطف و کمک، آن‌ها هستند.

چطور از یک شخص بلندپایه درخواستی داشته باشیم... در ایمیل

وقتی مدتی در یک شرکت کار کرده باشید، بالاخره زمانی خواهد رسید که برای درخواستی باید به یک مدیر پرمشغله یا هر شخص دیگری ایمیل بنزید. شاید برای تیمتان ابزاری نیاز داشته باشید یا بخواهید مسئله‌ای را رفع و رجوع کنید. هرچه باشد، احتمالاً اولین بار است که می‌خواهید با این فرد ارتباط برقرار کنید. در چنین حالتی تقریباً همه اشتباه تازه‌کاران را مرتکب می‌شوند. جمله‌های بیهوده و بی‌معنی و طولانی می‌بافند.

فیتز، چهارده سال پیش زمانی که در اپل کار می‌کرد، یک کامپیوتر «مک» برای مادرش خریداری کرده بود که چند ایراد کارخانه‌ای داشت. طبق توصیه همکاران، تصمیم گرفت یک ایمیل «کوتاه» برای «استیو جابز» بفرستد. از این ایمیل می‌توان به عنوان یک الگو برای نحوه مناسب ارتباط با مدیران بلندپایه استفاده کرد^۱:

تاریخ: پنجمین، اول فوریه ۲۰۰۱

به sjobs@apple.com:

موضوع: تبریۀ بسیار بد کاربر با سفت‌افزار ما - چه کار می‌توانم انجام دهم؟

بسیار سپاسگزار فواهم شد اگر توصیه کنید بطور می‌توانم این مشکل را حل کنم. این قضیه هم برای من و هم برای شرکت اپل قبالت آور است.

روز مادر، برای مادرم یک کامپیوتر iMac فریدم. او معاون یک مدرسه در نیواورلئان است و در مدرسه با یک مکینتاش قدیمی کار می‌کند. مادرم از داشتن این iMac بسیار هیجان‌زده بود و حتی تصمیم گرفته بود برای آزمایشگاه مدرسه نیز چند iMac خریداری کند و بودجه لازم را هم تهیه کرده بود.

در ماه ژوئیه، بعد از قرار گرفتن روی حالت خاموشی موقت دیگر روشن نشد. یکی از تعمیرگاه‌های مورد تأیید اپل تشفیص داد که یکی از بردهای الکترونیک آن فراب شده و آن را تعویض کرد. مادرم iMac را به خانه آورد، آن را روشن کرد و با صداهای عجیب و غریب و تصویر مک غمگین مواجه شد. به تعمیرگاه بازگشت و آن‌ها این بار یکی از بردهای آنالوک آن را تعویض کردند. در ماه سپتامبر مادرم را قانع کردم که دوباره از قابلیت خاموشی موقت استفاده کند، اما iMac دوباره روشن نشد و این بار بعد از تلاش بسیار و قطع و وصل کامل

۱. فیتز ابتدا قصد داشت که یک ایمیل طولانی و مفصل برای استیو بفرستد، ولی همکارانش توصیه کردند که متنش را خلاصه کند و نکاتی مختصر و مفید به همراه یک درخواست مشخص در آن قرار دهد.

برق آن موفق شدیم که آن را روشن کنیم. پس از آن کلاً قابلیت خاموشی موقت را غیرفعال کردیم. در ماه (سامبر، صغفه مانیاتور دوباره شروع به سوسو زدن کرد و لرزیدن رنگ‌های زرد و سبز و آبی بیشتر شد. مادرم می‌درد آن را نزد تعمیرگاه برد و اکنون آنجاست.

این وضعیت الان من است. مادرم فکر می‌کند که من سر به سرش گذاشته‌ام و به همه می‌گویم که iMac او یک کامپیوتر به‌درز نفور است. تا آنجا که من می‌شناسم، هیچ‌کس در شرکت نمی‌تواند به من کمک کند. آیا می‌توانم کاری انجام دهم تا او یک iMac سالم بگیرد؟

با احترام؛ فیتز

کمتر از ۲۰ ساعت بعد، شخصی از طرف استیو با فیتز تماس گرفت و دو هفته بعد مادر فیتز یک iMac جدید در اختیار داشت.

یک راز بزرگ: وقتی فرصت حل و فصل یک مسئله وجود داشته باشد، معمولاً مسئولان و مدیران صرف‌نظر از میزان مشغله کاری‌شان خوشحال خواهند شد تا به شما کمک کنند. خیلی از آن‌ها از حل کردن مشکلات لذت می‌برند و تمام آن‌ها اهمیت به دست آوردن اندکی سرمایه سیاسی را درک می‌کنند. متأسفانه صندوق ایمیل این افراد مملو از ایمیل‌های بی‌شمار است. و اگر با یک ایمیل از طرف کسی که هیچ‌وقت ندیده‌اند مواجه شوند که از ۳۰۰۰ کلمه و یک متن طولانی، بدون هیچ فاصله‌ای بین پاراگراف‌ها تشکیل شده باشد، به احتمال زیاد ۱۵ کلمه اول را می‌خوانند، دکمه حذف را فشار می‌دهند و سراغ ایمیل بعدی می‌روند.

ولی اگر بتوانند با ۱۰ ثانیه خواندن یک ایمیل و تکان دادن عصای سحرآمیزشان (ایمیل زدن به دستیارشان جهت اقدام) مشکلی را رفع و رجوع کنند، به احتمال خیلی زیاد این کار را انجام می‌دهند. چند ثانیه برای محول کردن کار وقت صرف خواهند کرد و مقدار زیادی امتیاز سیاسی از شما کسب می‌کنند.

پس از سال‌ها سعی و خطا، به این نتیجه رسیده‌ایم که احتمال دریافت پاسخ از ایمیل‌های کوتاه بیشتر است.

ما این روش را «تکنیک سه نکته و یک درخواست» می‌نامیم. این روش، احتمال به نتیجه رسیدن درخواست و یا حداقل دریافت پاسخ از هر شخصی (نه لزوماً یک مدیر مسئول) را به شدت افزایش می‌دهد.^۱

یک ایمیل خوب و مطابق با اصول سه نکته و یک درخواست، (حداکثر) سه نکته اصلی در مورد موضوع مورد بحث و یک و فقط یک درخواست دقیق و مشخص دارد. همین! نه هیچ چیز بیشتر و اضافه‌ای. باید ایمیلی بنویسید که بتوان آن را به راحتی برای شخص دیگری ارسال کرد. اگر در ایمیل خود از این در و آن در صحبت کنید، مطمئن باشید که مخاطب فقط به یکی از موضوعات (دقیقاً همان موضوعی که کمترین اهمیت را برای شما دارد)

۱. اخطار: این توصیه به شما کمک نمی‌کند تا با رئیس جمهور یا مسئول فروش یک سوپرمارکت زنجیره‌ای بزرگ زمان مصاحبه تنظیم کنید. این تکنیک فقط به درد درخواست‌های واقع‌بینانه می‌خورد.

پاسخ خواهد داد. یا حتی شاید بدتر، نیاز بار ذهنی ایمیل شما به قدری زیاد می‌شود که شخص مورد نظرتان احتمالاً از پاسخ‌گویی صرف‌نظر می‌کند.

نکات مطرح‌شده باید جملات کوتاهی باشند (هر کدام در یک خط جای بگیرند). درخواست شما نیز باید تا آنجا که ممکن است کوتاه باشد. اگر به دنبال گرفتن جواب هستید، پاسخ‌گویی را برای مخاطب آسان کنید تا بتواند با یک یا دو کلمه کار شما را راه بیندازد. نیم دوجین سؤال نپرسید. سعی کنید که در هر پاراگراف و ترجیحاً در هر ایمیل فقط یک سؤال پرسیده شود.

نهایتاً، ایمیل شما باید بر پایه و اصول HRT نوشته شود: مؤدبانه، با احترام و عاری از اشکالات املائی و دستوری. اگر واقعاً نمی‌توانید ایمیلتان را کوتاه کنید و نیاز دارید که حتماً اطلاعات تکمیلی بیشتری ارائه دهید، این اطلاعات را در انتهای ایمیل و بعد از امضای خودتان تحت عنوان «اطلاعات بیشتر» و یا «موضوعات پیش‌زمینه» قرار دهید.



اکنون که به گذشته نگاه می‌کنیم، به نظرمان ایمیل فیتز به استیو خیلی هم خوب نبود و شاید طولانی‌تر از حالت ایدئال بود. اگر می‌خواستیم ایمیل را امروز بنویسیم، این‌طور می‌نوشتیم:

تاریخ: پنجشنبه، اول فوریه ۲۰۰۱

به sjobs@apple.com :

موضوع: تجربه بر یک مشتری - می‌توانید کمک کنید؟

برای مادرم، که معاون یک مدرسه است، یک کامپیوتر iMac خریدم. مادرم از داشتن این iMac بسیار هیجان‌زده بود و حتی می‌خواست برای آزمایشگاه مدرسه نیز چند iMac بفرود و بودجه لازم را هم تهیه کرده بود.

ماه ژوئیه، شرکت اپل یکی از بردهای دیجیتال و یک ماه بعد یکی از بردهای آنالوگ آن را عوض کرد. در ماه سپتامبر قابلیت شاموشی موقت آن به کلی از کار افتاد و در ماه دسامبر صفحه نمایشگر دچار مشکلاتی شد. در حال حاضر این کامپیوتر در تعمیرگاه است. مادرم به همه می‌گوید که iMac او یک کامپیوتر به‌درزفور است و تا آنجا که من می‌شناسم هیچ‌کس در شرکت نمی‌تواند به من کمک کند. آیا می‌توانم کاری انجام دهم تا او یک iMac سالم به دست بیاورد؟

با احترام، فیتز

این ایمیل بازنویسی شده، بسیاری از حواشی نوشتاری را حذف کرده و یک مدیر پرمشغله، به راحتی و ظرف ۱۰ ثانیه می‌تواند آن را بخواند.

در طول زندگی کاری‌مان بارها و بارها به وسیله این تکنیک‌ها کارهایمان را راه انداخته‌ایم. ولی گاهی اوقات تمام فوت‌وفن و حقه‌های حرفه‌ای دنیا نیز برای نتیجه گرفتن یک کار کافی نخواهند بود.

برنامه جایگزین: خارج شوید!

در تمام این سال‌هایی که درباره نحوه مواجهه با سازمان‌های بد و یا آدم‌های سمی صحبت کرده‌ایم و سمینار ارائه داده‌ایم، همیشه آدم‌هایی را دیده‌ایم که پس از پایان سخنرانی‌هایمان نزد ما آمده و ناامید می‌گفتند که همه این راه و روش‌ها را امتحان کرده‌اند، ولی هیچ بهبودی در روابط و پیشرفتی در کارهایشان ندیده‌اند. چه کار می‌توانند انجام دهند؟ متأسفانه پاسخ ما به آن‌ها این است: «احتمالاً هیچ کار دیگری نمی‌توانید انجام دهید. قربانی نباشید. وسایلتان را جمع کنید و از آنجا خارج شوید.»

اگر نمی‌توانید سیستم را تغییر دهید، ائتلاف وقت و انرژی هیچ فایده‌ای ندارد. به جای تلاش برای تغییر یک نظام غیرقابل تغییر، نیرو و زمان خود را در راستای خروج از نظام سرمایه‌گذاری کنید: «رزومه خود را به‌روز کنید و از دوستان خود در مورد موقعیت‌های شغلی موجود سؤال کنید. مهارت‌های جدید را یاد بگیرید؛ چرا که یکی از فواید مهارت‌های دانش‌محور این است که این روزها همیشه مورد نیاز هستند و به شما کمک می‌کنند تا کنترل آینده خود را به دست بگیرید.»

توانایی کنترل زندگی، به شدت آزادی‌بخش است. وقتی با کمی جست‌وجو متوجه شوید که فرصت‌های شغلی دیگری نیز برای شما موجود است، کارایی شما در کارتان افزایش و تنش و آشفتگی شما کاهش می‌یابد. زیرا متوجه خواهید شد که اگر کارفرمای شما تصمیم بگیرد شما را اخراج کند، دنیا برای شما به اتمام نخواهد رسید.

یکی از مهندسين گوگل به نام چید منگ‌تن، در وبلاگ خود متنی نوشت که الهام‌بخش نحوه تصمیم‌گیری ما در کارهایمان شد:

کار درست را انجام دهید، سپس صبر کنید تا اخراج شوید.

کارمندان جدید گوگل (به آن‌ها نوگلر می‌گوییم) گاهی از من می‌پرسند که راز موفقیت من چیست. جواب توأم با شوخی من به آن‌ها این است که من همیشه کاری را انجام می‌دهم که برای گوگل و دنیا خوب و درست باشد. سپس منتظر می‌نشینم تا مرا اخراج کنند. اگر اخراج نشدم، کار درست را برای همه انجام داده‌ام. ولی اگر اخراج شدم، پس برای کارفرمای اشتباهی کار می‌کردم. در هر دو صورت من برنده نهایی خواهم بود. این استراتژی کاری من است.

اگر همیشه آماده باشید و حدود اختیار خود را بشناسید، آزادترین انسان روی زمین خواهید بود. از تغییر کارتان نترسید. بیش از پنج سال است که دائماً همین نصیحت را به مهندسين می‌کنیم و خوشحالییم که بارها افراد زیادی به ما ایمیل زده‌اند تا اطلاع دهند با عمل به این توصیه ما، اکنون جایی کار می‌کنند که احساس رضایت و شادی بیشتری دارند. یکی از بهترین ایمیل‌هایی که دریافت کردیم و شدیداً ما را خوشحال کرده این است:

تاریخ: پنجمین، اول دسامبر ۲۰۱۱

موضوع: تشکر

از: الکس مراولژویچ

به: برایان فیتز

سلام برایان،

امتلاً مرا به یاد نمی‌آورید، ولی شما دو تا از بهترین توصیه‌هایی که زندگی کاری‌ام را از این‌رو به آن‌رو کرده، به من داده‌اید. بعد از سفرانی شما در Google IO سال ۲۰۱۰ نزد شما آمدم تا دربارهٔ وضعیت کاری‌ام در آن زمان مشورت کنم. به سادگی به من گفتید که هرچه سریع‌تر از محل کارم بیرون بروم. رزومهٔ خودم را برای شما و به عنوان یک مدیر پروژه فرستادم. پاسخ شما این بود که کوکل فیلی به ندرت مدیر پروژه‌های غیر فنی را استفدام می‌کند.

در نتیجه تصمیم گرفتم که در مورد زندگی کاری‌ام به طور اساسی تجدیدنظر کنم. خط مشی شرکت کوکل هرچه باشد، بیشتر شرکت‌ها نیز از آن پیروی خواهند کرد. تنها انتقاب من این بود که مسیر شغلی خودم را تغییر دهم. راحت‌ترین راه برای من «مدیریت محصول» بود. سفت تلاش و مطالعه کردم و نه تنها از شرکتی که در آن بودم، بلکه از کشورم هم خارج شدم، چرا که موقعیت‌های شغلی زیادی برای مدیریت محصول در ونزوئلا وجود نداشت. این بهترین تصمیم زندگی من شد!

اکنون و به دلیل همین تصمیم، اخیراً یک پیشنهاد کاری فوق‌العاده به عنوان «رهبر محصول» در ژاپن به دست آوردم و ماه فوریه به شهر «کوبه» نقل مکان می‌کنم. به خودم قول داده بودم که اگر همه‌چیز به خاطر توصیهٔ شما خوب پیش رفت، برایتان یک نامهٔ تشکر بفرستم.

این خدمت شما؛ سپاسگزارم!

الکس

همه چیز از دست نرفته است

تمام این صحبت‌ها درباره استعفا از کار یا صبر کنید تا اخراج شوید، به این معنی نیست که اگر از کارت‌تان یا موقعیت خود ناراضی هستید، باید رزومه خود را به‌روز کنید و راهی خیابان شوید. کاملاً برعکس، اولین هدف شما باید ایجاد تغییرات لازم برای بهبود وضعیت کنونی و تلاش برای رسیدن به اهدافتان در کارت‌تان باشد. این فصل به شما ابزار و تکنیک‌های زیادی در این راستا معرفی کرده است. اگر برای اداره روابط و مسائل سازمانی تلاش نکنید، بخشی زیادی از سرنوشت خودتان را به دست شانس و اقبال سپرده‌اید.

فصل ششم

کاربران هم آدم‌اند

تاکنون فهرستی بلندبالا از موارد اساسی برای توسعه موفقیت‌آمیز یک نرم‌افزار را بررسی کردیم.

با یک گروه از آدم‌های باهوش و خلاق شروع کنید. فرهنگی قوی مبتنی بر تواضع، اعتماد و احترام را به تیم تزریق کنید. رهبر خدمتگزار آن‌ها باشید و به آن‌ها اختیار تصمیم‌گیری و قدرت همکاری اهدا کنید. آب، نور کافی، خط سیر و انگیزه‌های درونی مناسب را برای آن‌ها فراهم کنید. از آن‌ها در برابر نفوذ رفتارها و محیط‌های مخرب که فرهنگ و قدرت پیشرفت تیم را تهدید می‌کنند، محافظت کنید. آن‌ها را در دمای ۲۲ درجه سانتی‌گراد به مدت شش ماه خوب بپزید تا یک نرم‌افزار فوق‌العاده تولید کنید. به همین سادگی، نه؟

بسیاری از برنامه‌نویسان کارشان را در همین نقطه به اتمام می‌رسانند. یک نرم‌افزار برای خودشان توسعه داده‌اند و با رضایت از نتیجه کارشان اعلام پیروزی می‌کنند.

متأسفانه، دنیای واقعی این‌طور کار نمی‌کند. یک «نرم‌افزار خوب» بخش کوچکی از تعریف موفقیت است. اگر به دنبال کسب درآمد (و یا حتی بهتر کردن رزومه کاری خود) هستید، به افرادی نیاز دارید که با رضایت از نرم‌افزار شما استفاده کنند. فرایند توسعه نرم‌افزار با پرتاب محصول نهایی به آن سوی دیوار به اتمام نمی‌رسد. در حقیقت، هیچ‌وقت به اتمام نمی‌رسد. مردم از نرم‌افزار شما استفاده می‌کنند و شما باید نسبت به خواسته‌های آن‌ها عکس‌العمل نشان دهید و به مرور زمان نرم‌افزار را بهتر و مفیدتر کنید. اگر مهارت لازم برای مدیریت حلقه بازخورد از کاربران را به دست نیاورید، محصول شما از بین خواهد رفت.

در این فصل سه مرحله کلی تعامل کاربر با نرم‌افزار را بررسی می‌کنیم. ابتدا باید کاربران را متوجه کار خود کنید. آیا آن‌ها از وجود نرم‌افزار شما مطلع هستند؟ قبل از شروع به کار چه درک و فهمی نسبت به نرم‌افزار شما دارند؟ سپس باید به تجربه کاربری آن‌ها با نرم‌افزار توجه کنید. آیا انتظارات آن‌ها برآورده می‌شود؟ آیا نرم‌افزار قابل استفاده است؟ آیا به کاربران قدرت به دست آوردن اهدافشان را می‌دهد؟ و نهایتاً بررسی می‌کنیم که چطور می‌توانیم به طور مؤثر با کاربران فعال نرم‌افزار ارتباط برقرار کنیم. تمام موارد مطرح شده بخشی از مراحل گردشی و طبیعی توسعه نرم‌افزار است.

حرف آخر ما این است: مشارکت برای توسعه یک محصول خوب فقط محدود به کار کردن با اعضای تیمتان نیست. لازم است که دائماً با کاربران نرم‌افزار نیز همکاری و ارتباط داشته باشید.

اگر به این توصیه‌ها عمل نکنید، تمام چیزی که به دست می‌آورید یک نرم‌افزار درخشان، ولی بدون هیچ کاربر و استفاده‌کننده است. در این شرایط شاید بهتر باشد که در انتخاب شغل خود تجدیدنظر کنید!

مدیریت تصورات عمومی

وقتی کلمه «بازاریابی» را می‌شنوید، نخستین چیزی که به ذهنتان می‌رسد چیست؟

احتمالاً مثل بیشتر آدم‌ها به یاد فروشندگان فریبکار می‌افتید که با لبخندهای تصنعی و موهای ژل‌زده سعی دارند تصویری دل‌خواه از یک محصول به شما ارائه دهند. اگر محصول شما یک تکه گوشت خام باشد، کار افراد بخش بازاریابی و تبلیغات این است که با اضافه کردن صدای جلیز و یلیز آن را تبدیل به یک استیک جذاب برای مشتریان بکنند.

چرا این قضیه این قدر اذیت‌کننده است؟ چرا فکر تبلیغات و بازاریابی لرزه به تنمان می‌اندازد؟

علت این است که برای ما برنامه‌نویسان، بازاریابی معرف فرهنگی متضاد با هنجارهای مهندسی است. تمام فکر و ذهن ما درگیر حقیقت است. کد ما یا کامپایل می‌شود یا خیر. نرم‌افزار ما یا یک قابلیت را دارد یا نه. یا مسئله‌ای را حل می‌کند یا خیر. ما توصیفمان از دنیا را نمی‌پیچانیم. حقایق را آن‌طور که هستند درک می‌کنیم و سعی می‌کنیم تا آن‌ها را تغییر دهیم. از دید ما بازاریابی سرشار از دروغ و فریب است و ما دوست نداریم که دروغ بشنویم. هنگام تصمیم‌گیری ما به دنبال نظم، وقایع قابل پیش‌بینی و توضیحات دقیق هستیم.

اصول بازاریابی و تبلیغات در تناقض با میل درونی یک سازنده برای شایسته‌سالاری است. ما معتقدیم که همیشه بهترین محصول شایسته پیروزی است و منظور ما از «بهترین» محصول آن نرم‌افزاری است که واقعاً و حقیقتاً بالاترین کیفیت را داشته باشد، نه لزوماً آنکه بهترین تبلیغ را کرده است. به تناوب، از اینکه می‌بینیم فناوری‌ها و نوآوری‌های بهتر جای خود را به سایرین می‌دهند ناراحت شده‌ایم. بسیاری از افراد معتقدند که بیتاماکس از VHS بهتر بود، یا لیزردسک از دی وی دی بهتر بود و یا لیسپ^۱ بهترین زبان برنامه‌نویسی است. حتی در دنیای سیستم‌های کنترل نسخه، ساب‌ورژن بین شرکت‌ها و سازمان‌های بزرگ جایگاه خود را پیدا کرد، در حالی که سیستم‌های جدیدتر مثل گیت^۲ بسیار بهتر بودند.

۱. LISP؛ یک زبان برنامه‌نویسی رایانه است که در سال ۱۹۵۸ به وسیله جان مک‌کارتی ابداع شده است. (و.)

۲. Git؛ گیت یک نوع سیستم کنترل ورژن VCS است که می‌توانید تغییرات اعمال‌شده در فایل‌ها را ساده‌تر پیگیری کنید. (و.)



از همه این‌ها بدتر اینکه، به عقیده ما اعضای بخش تبلیغات معمولاً بیش از حد توانایی به مشتریان قول می‌دهند. در نتیجه همیشه این‌طور به نظر می‌رسد که مهندسين انتظارات را برآورده نمی‌کنند. این موضوع به قدری ما مهندسين را عصبانی می‌کند که گله‌مان سوت می‌کشد!

ما اینجا برای شما یک خبر بد و یک خبر خوب داریم.

خبر بد اینکه خیر، نمی‌توان از تبلیغات صرف‌نظر کرد. بازاریابی مهم است و باید آن را انجام داد. اما خبر خوب این است که می‌توان با بخش تبلیغات و بازاریابی همکاری کرد. لزومی ندارد روابط شما با آن‌ها شکرآب باشد. در حقیقت می‌توان مشارکت با تبلیغات را به یک ابزار قدرتمند برای موفقیت تبدیل کرد.

برنامه‌نویسان معمولاً توانایی منطق و استدلال قوی‌تری نسبت به قوه‌احساسات دارند، در حالی که بیشتر آدم‌ها به یک اندازه تحت‌تأثیر منطق و احساس هستند. بازاریابان در استفاده از احساسات انسان‌ها تخصص دارند. به همین دلیل است که در کارشان بسیار کارآمد هستند. حقایق و احساسات را با هم تلفیق می‌کنند تا توجه مخاطب را به دست بیاورند. اگر می‌خواهید برای نرم‌افزارتان کاربران جدیدی پیدا کنید، مجبور هستید که به درک و برداشت احساسی آن‌ها از نرم‌افزارتان اهمیت دهید. هیچ‌گاه نمی‌توانید نحوه‌تصمیم‌گیری افراد را تغییر دهید.

شرکت اپل، استاد تمام‌عیار در زمینه جذب احساسات مشتری‌های غیر فنی است. چند سال به عقب بازمی‌گردیم و می‌پرسیم: «آیا گوشی آیفون از اندروید بهتر است؟ از لحاظ قابلیت‌های فنی، هر دو تقریباً مشابه یکدیگر هستند، ولی اگر یک کاربر غیر فنی عمیقاً اعتقاد داشته باشد که آیفون یک گوشی سحرآمیز است، در نتیجه آیفون حداقل برای آن تک‌کاربر، یک گوشی سحرآمیز می‌شود. ادراک احساسی و تصورات ما تشکیل‌دهنده واقعیت هستند. در حقیقت «تصور، نود درصد از واقعیت است.»

وسوسه‌برانگیز است که به اشتباه این‌طور فکر کنیم که بهترین راه برای پیروزی در این بازی، شرکت نکردن در بازی است. ولی باید توجه داشت که بازاریابی بازی‌ای نیست که بتوان از آن صرف‌نظر کرد. حتی برای اینکه بتوانید نرم‌افزار خود را صرفاً عرضه کنید و در معرض دید کاربران قرار دهید، نیاز به یک تدبیر حداقلی در زمینه تبلیغات و بازاریابی دارید. اگر درباره‌آن هوشمندانه عمل کنید، متوجه می‌شوید که بازاریابی می‌تواند نیرویی باشد که تلاش‌های شما در مهندسی نرم‌افزار را چند برابر می‌کند. در این بخش، توصیه‌هایی بنیادی را برای در دست گرفتن کنترل امور بازاریابی بر پایه اصول HRT بیان می‌کنیم.

به تأثیرات اولیه توجه کنید

اگر گرسنه باشید و به دنبال یک رستوران بگردید، شکل ظاهری رستوران در خیابان برای شما اهمیت خواهد داشت. اگر ظاهر زشت و چندش‌آوری داشته باشد، وارد آنجا نمی‌شوید. ولی اگر گرم و دوستانه باشد و میزبانی مهربان داشته باشد، مشتاق می‌شوید تا آنجا را امتحان کنید. تأثیر احساسی ابتدایی کاربران را وقتی که اولین بار نرم‌افزار شما را می‌بینند دست‌کم نگیرید. اگر تاکنون جعبه‌آی‌پد یا ترموستات‌های نست^۱ را باز کرده باشید، می‌دانید که دقیقاً درباره‌ی چه حرف می‌زنیم.

محصول شما در مقابل دید یک تازه‌کار چگونه است؟ آیا کاربر از آن استقبال می‌کند و گشت‌وگذار در نرم‌افزار را تشویق می‌کند؟ از طرف دیگر برای یک کاربر حرفه‌ای نیز آشنا به نظر می‌آید؟ در نگاه اول، آیا نرم‌افزار شما کارایی را فریاد می‌زند یا نشان می‌دهد که نیازمند آموزش‌های طولانی و طاقت‌فرساست؟ به طور خاص، کاربر نرم‌افزار در سی ثانیه‌ی نخست کار، با محصول شما چه تجربه‌ای کسب می‌کند؟ به یک جواب منطقی اکتفا نکنید که او یک منو شامل چند انتخاب می‌بیند و یک محل مشخص برای ورود به نرم‌افزار. بلکه از بعد عاطفی و احساسی نیز به قضیه توجه کنید. نرم‌افزار بعد از یک دقیقه‌ی اول چه احساسی به کاربر شما القا می‌کند؟ توانمندی یا سردرگمی؟ برای بهتر شدن این حس، چه کارهایی می‌توانید انجام دهید؟ یک قدم به عقب بردارید و به وب‌سایت محصول خودتان نگاه کنید. حرفه‌ای و جذاب مثل ویتترین خوب یک مغازه به نظر می‌آید؟ به تمام این موارد باید با دقت توجه کنید تا دیگران نرم‌افزار شما را جدی بگیرند.

۱. Nest؛ ترموستات‌های هوشمند نیست، یک ترموستات متصل به برنامه‌های اندرویدی است که به شما این اجازه را می‌دهد تا از راه دور دمای خانه را کنترل کنید. (و.)

کم وعده دهید و فراتر از انتظار تحویل دهید

هنگام قول و قرار با مشتریان، اجازه پیش‌دستی به تیم تبلیغات و بازاریابی را ندهید. وقتی کاربران درباره یکی از قابلیت‌های آینده نرم‌افزار یا برنامه زمان‌بندی انتشار نرم‌افزار سؤال می‌کنند، در برآورد و تخمین بسیار محافظه‌کارانه عمل کنید. اگر اختیار عمل را دست تیم تبلیغات بدهید، به سرنوشت بازی Duke Nukem Forever دچار می‌شوید که پانزده سال پس از موعد مقرر و وعده داده‌شده منتشر شد. ولی اگر پیام شما (که دقیق‌تر است) زودتر به گوش کاربران برسد، همیشه از نتایج کارتان خوشحال و هیجان‌زده می‌شوند. گوگل در این زمینه بسیار خوب عمل می‌کند. آن‌ها هیچ‌گاه قابلیت‌ها و نوآوری‌های خودشان را از قبل اعلام نمی‌کنند. معمولاً قابلیت‌های جدیدشان پس از انتشار، کاربران را شگفت‌زده می‌کنند. این کار همچنین از نگرانی‌های بی‌مورد برای رسیدن به ضرب‌الأجل‌های خودساخته و اعلام‌شده جلوگیری می‌کند. نرم‌افزار وقتی منتشر می‌شود که واقعاً آماده باشد.

با تحلیل گران صنعت با احترام رفتار کنید

بسیاری از برنامه‌نویسان از دنیای رسانه بیزار هستند. از نظر آن‌ها رسانه‌ها همان تبلیغات و بازاریابی هستند در یک لباس دیگر. وقتی یک مجله یا ماهنامه تجاری به سراغ شرکت می‌آید، بسیاری از شرکت‌ها هرچه را در دست دارند زمین می‌گذارند و در برابر تمام درخواست‌های آن‌ها سجده می‌کنند. شرکت‌ها متوجه هستند که یک مقاله مثبت (یا منفی) درباره آن‌ها چقدر روی تصور مشتریان از محصولاتشان تأثیر می‌گذارد. مهندسين ولی عموماً از قدرت بالای چنین مقالاتی خشمگین هستند.

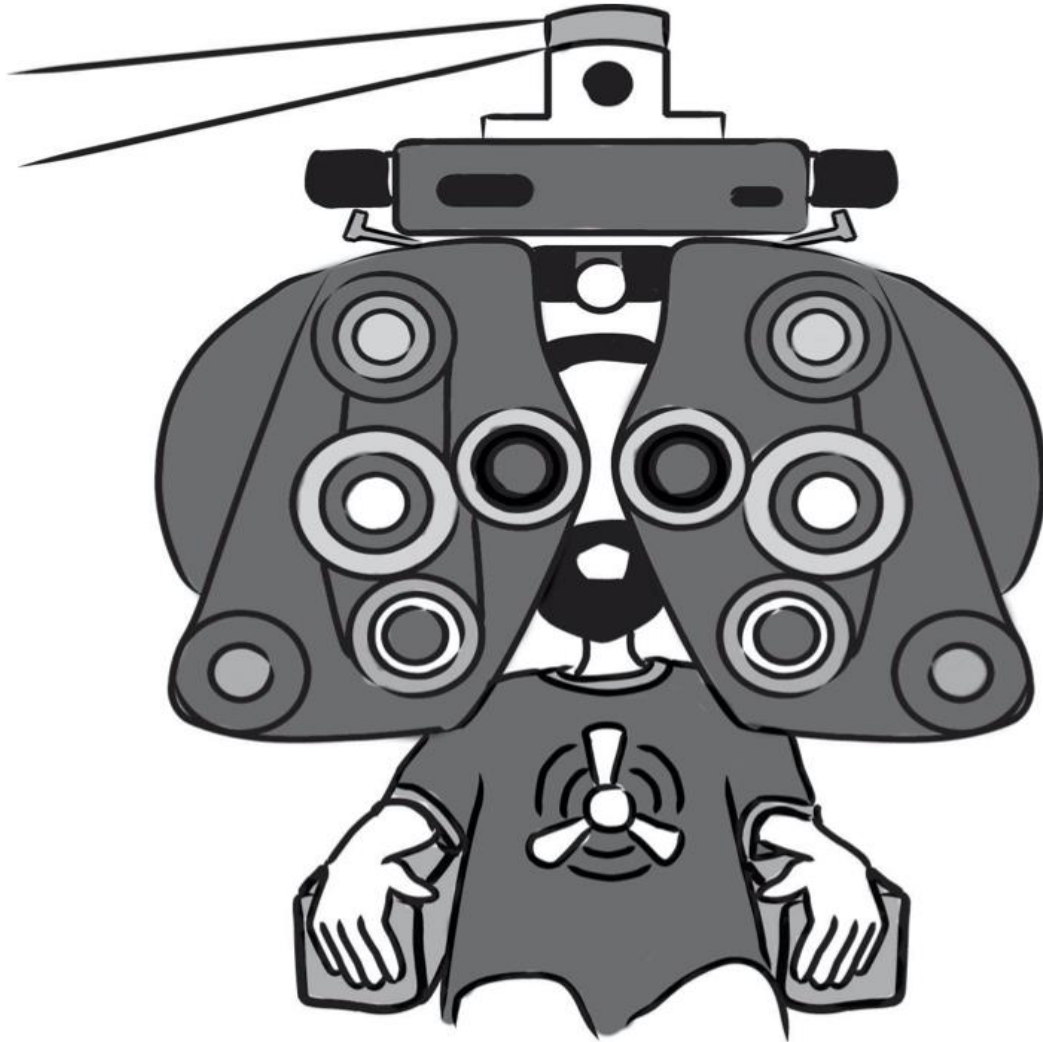
به عنوان مثال، زمانی بود که اعضای بنیاد نرم‌افزار آزاد آپاچی در مواجهه با اعضای رسانه و تحلیل گران مشکلاتی داشتند. یکی از تحلیل گران رسانه که به دنبال مقاله استاندارد و صنعتی برای توضیح مشخصات سرور HTTPD بود، با این پاسخ مواجه می‌شد: «برو مثل بقیه، مستندات نوشتاری پروژه رو بخون!» در حالی که این جواب ممکن بود حس درونی برنامه‌نویسان در راستای عدالت و شایسته‌سالاری را ارضا کند، ولی کمکی به دیدگاه و تصور عمومی از پروژه نمی‌کرد، مخصوصاً تصور کلی کاربران سازمانی و شرکتی. نهایتاً مدیر روابط عمومی تلاش کرد تا اعضای بنیاد را در راستای نحوه برخورد با رسانه و تحلیل گران تعلیم دهد و از مشکلات مشابه جلوگیری کند. خشونت پنهان و پرخاشگری منفعلانه هیچ کمکی نمی‌کند. مثل این می‌ماند که به یک منتقد رستوران بگوییم در انتهای صف بایستد. آیا با منتقدان باید رفتار خاص و ویژه داشت؟ احتمالاً نه؛ ولی آیا باید اصول و قوانین را مدام جلوی چشمشان آورد؟ قطعاً خیر. موضوعاتی را که ارزش نبرد و مبارزه دارند با دقت و وسواس انتخاب کنید.

نرم افزار شما چقدر قابل استفاده است؟

یک حقیقت سخت: کاربران نرم افزار شما برنامه نویس نیستند، مگر آنکه محصول شما ابزارهای مهندسی باشد. نتیجه این واقعیت این است که شما به عنوان یک مهندس، بدترین شخص برای ارزیابی قابلیت استفاده از نرم افزارتان هستید. یک واسط کاربری که شاید از دید شما بسیار منطقی و خوب به نظر بیاید، ممکن است باعث شود همسایه غیر فنی شما موهایش را از شدت عصبانیت از جا بکند.

اگر فرض کنیم که «نرم افزار موفق» یعنی نرم افزاری که «آدم‌های زیادی از آن استفاده کرده و راضی باشند»، باید به کاربران خود توجه ویژه داشته باشید. گوگل در این راستا شعار معروفی دارد:

«روی کاربران تمرکز کنید، سایر قضایا به دنبال شما می‌آیند.»



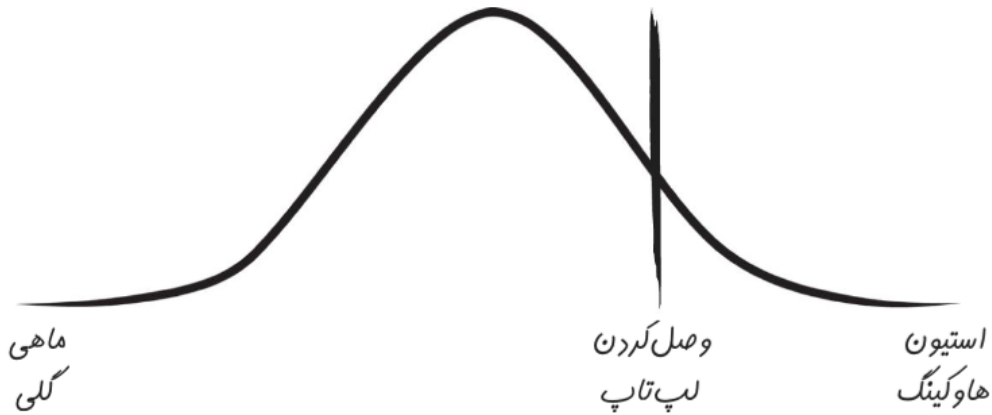
این توصیه ما شاید خیلی پیش پا افتاده به نظر برسد، ولی در طول عمر کاری مان بارها و بارها نرم افزارهایی را دیده ایم که فقط به خاطر قابلیت استفاده شان موفق شده یا شکست خورده اند.

یکی از نقاط عطف در پیشرفت گوگل وقتی بود که میزان تأثیر تبلیغاتشان را سنجیدند. اگر کاربری روی یک لینک تبلیغاتی کلیک کند، پس آن لینک حتماً برای او مفید بوده است. اگر هیچ وقت کسی روی آن کلیک نکند، پس احتمالاً آن تبلیغ بی فایده و مزاحم است. تبلیغات بد از سیستم خارج می شوند و به صاحب آن ها این بازخورد داده می شود تا نحوه تبلیغاتش را بهتر کند. این استراتژی ابتدا ممکن بود زیان بخش به نظر برسد، چرا که گوگل عملاً از پذیرش درآمد امتناع می کرد؛ ولی با مرکز توجه قرار دادن «جست و جوگر» به جای «تبلیغ کننده» قابلیت استفاده و کارایی نرم افزار را شدیداً بهبود بخشیدند و تعداد کاربران را در طولانی مدت افزایش دادند.

در این بخش درباره تعدادی از روش های مهم تمرکز روی کاربران صحبت می کنیم.

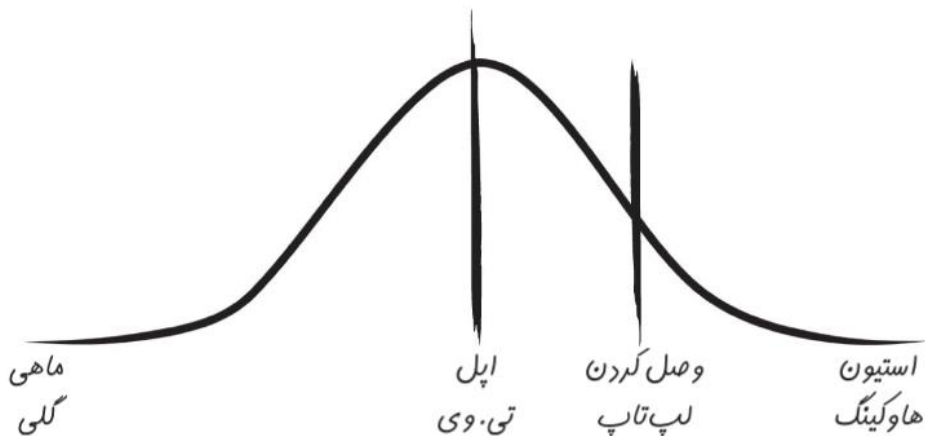
مخاطبان اصلی را پیدا کنید

پیش از هر چیز، تصور کنید که کاربران شما از نظر قابلیت‌های فنی روی محور زیر قرار می‌گیرند:



کاربران ایدئال نرم‌افزار شما کجای این محور قرار می‌گیرند؟ اگر یک خط عمودی، بیانگر نقطه ایدئال روی محور در مرکز این طیف قرار دهیم، به این معنی است که نیمی از کاربران (آن دسته که سمت راست خط عمودی روی طیف کاربران قرار دارند) از کار کردن با نرم‌افزار شما راضی خواهند بود.

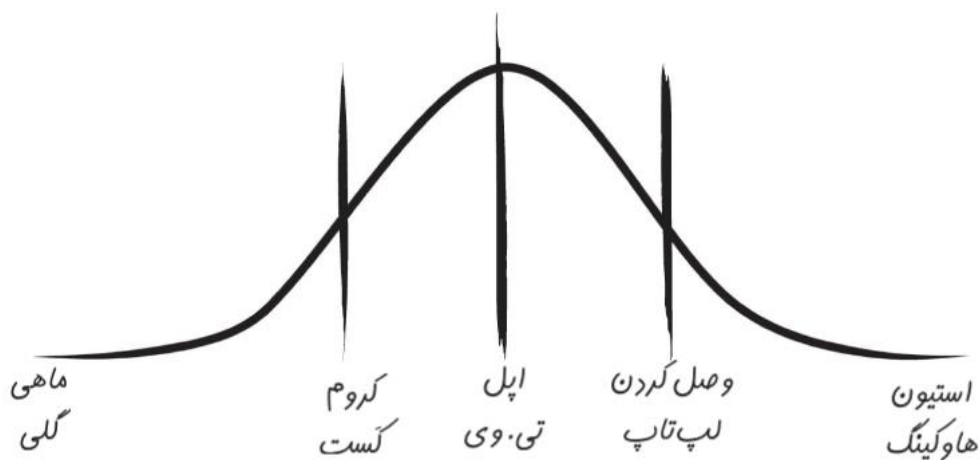
به عنوان مثال، به مسئله نمایش محتوای اینترنت روی تلویزیون‌های بزرگ توجه کنید. راه‌های مختلفی با قابلیت‌های کاری متفاوت برای این کار وجود دارند و برای به دست آوردن مخاطب‌های بیشتر با یکدیگر رقابت می‌کنند. در ابتدا کاربران مجبور بودند که لپ‌تاپ‌های خودشان را به کمک یک کابل مخصوص به تلویزیون متصل کنند. برای این کار لازم بود که درک خوبی از ورودی‌های تلویزیون و خروجی‌های لپ‌تاپ و همچنین کابل‌های ویدئویی و صوتی مختلف داشته باشند.



برای راحت کردن این کار، شرکت اپل محصولی را معرفی کرد تحت عنوان اپل تی. وی^۱ که یک دستگاه کوچک مشابه کامپیوتر یا تلفن هوشمند بود و به کاربران قابلیت پخش تصاویر خصوصی یا اینترنتی خودشان را روی تلویزیون می‌داد. عدهٔ بیشتری از کاربران بدون نیاز به دانش خاص فنی موفق شدند از این دستگاه استفاده کنند. همراه این دستگاه کابل‌های مخصوص و مناسب نیز وجود داشت و کار را برای کاربران راحت‌تر می‌کرد.

مدتی بعد، گوگل با معرفی دستگاه کروم‌کست^۲ رقابت را در این راستا شدیدتر کرد. یک دستگاه کوچک که مستقیماً به ورودی HDMI تلویزیون‌ها متصل می‌شود و تنظیمات بسیار ساده‌تری دارد. علاوه بر سادگی، این قابلیت را نیز به کاربران می‌دهد تا از دستگاه‌های اپل یا غیر اپل خود تصاویر را روی تلویزیون خود پخش^۳ کنند. اکنون که این کتاب را می‌نویسیم، تلویزیون‌های جدید به طور پیش‌فرض قابلیت اتصال به وای‌فای و پخش اینترنتی را دارند. فرزندان ما احتمالاً هیچ‌وقت به یاد نمی‌آورند که روزی تلویزیون‌ها به طور پیش‌فرض نت‌فلکس را نداشته‌اند!

نکتهٔ کلیدی بحث ما این است که هنگام توسعهٔ نرم‌افزار، باید تا آنجا که ممکن است این خط عمودی را به سمت چپ محور هدایت کنیم تا کاربران بیشتری بتوانند به راحتی از نرم‌افزار ما استفاده کنند. به طور کلی هرچه تعداد کاربران ما بیشتر باشد، نرم‌افزار ما موفق‌تر است (و شرکت ما درآمد بیشتری کسب می‌کند!) نتیجهٔ اخلاقی این است که وقتی کاربران نرم‌افزار را در نظر می‌گیرید، با دقت فکر کنید که مخاطب اصلی شما کدام دسته از کاربران هستند. آیا نرم‌افزار شما برای بزرگ‌ترین گروه کاربران قابل استفاده است؟ به همین دلیل است که سادگی در رابط‌های کاربری و نیز راهنماهای آموزشی دقیق بسیار اهمیت دارند.



1. Apple TV
2. Chromecast
3. cast

موانع استفاده از نرم‌افزار را بشناسید

به کاربرانی فکر کنید که نخستین بار از نرم‌افزار شما استفاده می‌کنند. ورود به نرم‌افزار شما تا چه حد می‌تواند برای آن‌ها دشوار باشد؟ اگر کاربران به راحتی نتوانند محصول شما را امتحان کنند، نرم‌افزار شما را روی سیستم خود نگه نخواهند داشت. در نخستین مواجهه با نرم‌افزار، کاربر به دنبال مقایسه قابلیت‌های محصول شما با رقبا نیست. او فقط می‌خواهد کاری را سریع و بی‌دردسر به نتیجه برساند.

برای درک بهتر موضوع به زبان‌های برنامه‌نویسی‌ای که برای نوشتن اسکریپت^۱ استفاده می‌شوند توجه کنید. اکثر برنامه‌نویسان معتقدند که پرل یا پایتون^۲ نسبت به پی.اچ.پی^۳، زبان‌های «بهتری» هستند. آن‌ها باور دارند برنامه‌هایی که به وسیله پرل و پایتون و یا روبی^۴، نوشته شده‌اند در طولانی‌مدت قابلیت خوانایی و نگهداری راحت‌تری نسبت به پی.اچ.پی دارند؛ از مجموعه کتابخانه‌های بالغ‌تری برخوردارند و هنگام استفاده از اینترنت به طور ذاتی امنیت بیشتری دارند. با وجود این، می‌بینیم که زبان پی.اچ.پی حداقل برای برنامه‌نویسی‌های تحت وب محبوب‌تر است. چرا؟ چون هر دانش‌آموز دبیرستانی‌ای می‌تواند با کپی کردن کدهای دوستش وبسایت خودش را با پی.اچ.پی درست کند. برای یادگیری اولیه، نیازی به مطالعه هیچ کتاب یا حضور در هیچ کلاس درس طولانی‌ای و یا یادگیری الگوهای پیچیده برنامه‌نویسی نیست. فقط کافیست تا وبسایت خودتان را سر هم کنید و فوت و فن‌های پی.اچ.پی را از دوستانتان یاد بگیرید.

یک مثال دیگر می‌تواند نرم‌افزارهای ویرایش متن باشد. آیا برنامه‌نویسان بهتر است از «ای‌مکس استفاده کنند یا وی‌آی»^۵؟ اصلاً اهمیتی دارد؟ نه در واقع، ولی اصولاً چرا باید یک نفر یکی از این دو را به دیگری ترجیح دهد؟ یک حکایت واقعی: وقتی بن در دوره یک کارآموزی در سال ۱۹۹۰ آموزش لینوکس را یاد گرفت، اولین بار نرم‌افزار ویرایش متن «وی‌آی» را باز کرد و در کمتر از بیست ثانیه کلافه شد. با اینکه می‌توانست بین مکان‌های مختلف فایل حرکت کند، ولی نمی‌توانست هیچ‌چیزی بنویسد! البته که علاقه‌مندان «وی‌آی» می‌دانند برای نوشتن ابتدا باید وارد ادیت مود^۶ یا حالت ویرایش شد. ولی این محدودیت می‌تواند یک تجربه نامطلوب برای تازه‌کاران ایجاد کند. وقتی بن سراغ ای‌مکس رفت، به راحتی توانست همان ابتدا فایل مورد نظرش را ویرایش کند؛ چرا که تجربه کاربری ای‌مکس بسیار شبیه نرم‌افزارهای ویرایش فایل خانگی نظیر ورد^۷ بود. بن بعد از آنکه چند دقیقه با ای‌مکس

1. script

۲. Perl & Python؛ دو تا از زبان‌های برنامه‌نویسی (و.)

3. PHP

4. Ruby

۵. Emacs & Vi؛ توسعه‌دهندگان و ادمین‌های سیستم معمولاً برای کارهای روزمره خود از یکی از ویرایشگرهای متن وی‌آی یا ای‌مکس استفاده

می‌کنند. هر دو ویرایشگر در اکثر توزیع‌های مطرح لینوکسی در دسترس هستند. (و.)

6. Edit mode

7. Word Processor

کار کرد، خودش را طرفدار آن اعلام کرد. اگرچه این دلیل منطقی‌ای برای انتخاب یک نرم‌افزار نیست، ولی اتفاقی است که همیشه می‌افتد. اولین دقیقه کار با یک محصول بسیار حیاتی است.^۱

البته که راه‌های دیگری برای خراب کردن برداشت‌های اولیه کاربران از نرم‌افزار نیز وجود دارند. وقتی اولین بار کاربر وارد نرم‌افزار می‌شود، یک فرم بزرگ و اجباری برای پر کردن و تنظیم مواردی که ترجیح می‌دهند، جلوی آن‌ها قرار ندهید. تحمیل ثبت‌نام و ایجاد حساب کاربری برای استفاده از نرم‌افزار هم خیلی جالب نیست. به کاربران القا می‌کند که باید تعهدی طولانی‌مدت بدهند، بدون آنکه حتی از سرویس استفاده کرده باشند. یک خط قرمز شخصی دیگر برای ما وقتی است که وبسایت‌ها در همان دو ثانیه اول درخواست می‌کنند تا برای خبرنگارهایشان نام‌نویسی کنیم. تمام این موارد باعث می‌شوند تا کاربران فریادزنان به سمت دیگری فرار کنند.

یک مثال خوب از یک سرویس بدون محدودیت ورود شاید وبسایت TripIt باشد که برای مدیریت برنامه‌های سفری استفاده می‌شود. برای استفاده از این سرویس فقط کافی‌ست ایمیل‌های تأییدیه بلیت‌های هواپیما، رزرو هتل و یا کرایه ماشین را به ایمیل plans@tripit.com بفرستید. همین! در همان لحظه شما استفاده از خدمات وبسایت TripIt را آغاز کرده‌اید. سرویس برای شما یک حساب کاربری موقت درست می‌کند، ایمیل‌های شما را تجزیه و تحلیل می‌کند و یک صفحه منظم و زیبای برنامه سفر برای شما درست می‌کند. سپس یک ایمیل برای شما می‌فرستد و می‌گوید که کار شما آماده است. دقیقاً مشابه یک منشی شخصی کارهای شما را منظم می‌کند، در حالی که کل کاری که شما باید انجام می‌دادید این بود که چند تا ایمیل را برای آن‌ها بفرستید. بدون هیچ زحمتی شما وارد سیستم شده‌اید و از سرویس استفاده می‌کنید. اینجاست که دیگر شاید دوست داشته باشید حساب کاربری دائمی نیز بسازید.

اگر از موانع ورودی نرم‌افزار خود مطمئن نیستید، سعی کنید چند آزمایش ساده انجام دهید. نرم‌افزار خود را به تعدادی انسان عادی (هم فنی و هم غیر فنی) بدهید و نحوه استفاده آن‌ها از کارتان را در حد یکی دو دقیقه مشاهده کنید. احتمالاً از چیزی که می‌بینید شگفت‌زده خواهید شد.

۱. البته که به طور کلی احتمالاً یادگیری کامل‌ای مکس، به اندازه یادگیری «وی» پیچیده است؛ ولی ما اینجا درباره برداشت‌های اولیه صحبت می‌کنیم تا مقایسه منطقی.

میزان استفاده را اندازه‌گیری کنید، نه استفاده‌کنندگان را.

هنگام بررسی کاربران و قابلیت استفاده از نرم‌افزارتان، به معیار سنجش و اندازه‌گیری میزان استفاده از نرم‌افزار توجه کنید. دقت کنید که مقصود اندازه‌گیری استفاده است و نه استفاده‌کنندگان و تعداد دفعات نصب نرم‌افزار. شما به دنبال کاربرانی هستید که از نرم‌افزار شما فعالانه استفاده کنند، نه اینکه صرفاً تعداد دفعات دانلود و نصب نرم‌افزار زیاد باشد. شاید گاهی اوقات می‌شنوید که کسی می‌گوید: «محصول من سه میلیون بار دانلود شده! یعنی سه میلیون کاربر از نرم‌افزار من راضی هستند!» صبر کن، تند نرو! چه تعداد از این سه میلیون نفر در حقیقت از نرم‌افزار استفاده کرده‌اند؟ منظور ما از «استفاده» همین است.

یک مثال شاید کمی افراطی دیگر: چه تعداد کامپیوتر در دنیا نرم‌افزار بایگانی‌سازی ar را نصب دارند؟ پاسخ این است که تقریباً تمام سیستم‌عامل‌های مبتنی بر یونیکس^۱، بعضی از نسخه‌های لینوکس، مک. او. اس^۲، بی. اس. دی^۳ و ... این نرم‌افزار را به صورت پیش‌فرض نصب دارند. ولی چه تعداد از کاربران از این ابزار استفاده می‌کنند؟ چند نفر از آن‌ها حتی می‌دانند که چنین ابزاری وجود دارد؟ این نرم‌افزار با وجود اینکه میلیون‌ها بار روی سیستم‌های مختلف نصب شده، تقریباً هیچ استفاده‌ای ندارد.

میزان استفاده، معیاری است که خیلی از شرکت‌ها (از جمله گوگل) به طور مدام آن را ارزیابی می‌کنند. یکی از واحدهای اندازه‌گیری متداول، تعداد کاربران فعال در بازه‌های زمانی هفت روزه و سی روزه است. یعنی چند نفر از کاربران از نرم‌افزار در طول هفته یا ماه گذشته استفاده کرده‌اند. این داده به شما کمک می‌کند که ایده خوبی نسبت به میزان موفقیت نرم‌افزارتان داشته باشید. به تعداد دانلودها توجه نکنید. راهی پیدا کنید تا میزان استفاده فعال از نرم‌افزار را بسنجید. به طور مثال اگر محصول شما یک وب‌سایت است، از ابزاری نظیر ابزار آنالیز گوگل^۴ استفاده کنید. این ابزار به شما کمک می‌کند تا نه تنها تعداد کاربران فعال را بدانید، بلکه اطلاعاتی نظیر اینکه کاربران چگونه به وب‌سایت شما می‌آیند و چه مدت زمانی را در سرویس شما سپری می‌کنند هم به دست بیاورید. این داده‌ها شاخص‌های فوق‌العاده‌ای برای درک عملکرد نرم‌افزار شما هستند.

1. Unix
2. MacOS
3. BSD
4. Google Analytics

الگوهای طراحی

قبل از ظهور اینترنت، بزرگ‌ترین چالش فروش هر نوع محصولی، پخش و توزیع آن بود. شرکت‌های انگشت‌شماری استطاعت توزیع بین‌المللی محصولاتشان در هزاران فروشگاه سراسر دنیا را داشتند. در نتیجه وقتی شرکتی محصولی یا نرم‌افزاری را تولید می‌کرد مجبور بود که تلاش بسیار زیادی در راستای تبلیغات و بازاریابی انجام دهد. طبیعتاً در هر دسته و موضوع نرم‌افزاری فقط یکی دو «برنده» می‌توانستند در سطح بین‌المللی ظاهر شوند: مثلاً ماکروسافت وُرد در برابر وُرد پرفکت^۱ یا اکسل^۲ در مقابل لوتوس^۳ و ... برای کاربران مهم‌ترین معیارهای تصمیم‌گیری هنگام انتخاب بین گزینه‌های موجود، قابلیت‌های نرم‌افزار و هزینه آن بود. کسی به زیبایی یا زشتی نرم‌افزار توجهی نمی‌کرد.

ولی این قضیه تغییر کرده است.

اینترنت یک شبکه بزرگ توزیع بین‌المللی است که باعث شده کاربران تقریباً بدون هیچ هزینه‌ای به هر نرم‌افزاری که بخواهند دسترسی داشته باشند و شبکه‌های اجتماعی نیز این امکان را فراهم کرده‌اند تا کاربران به راحتی نظرات و احساساتشان درباره محصولات مختلف در دنیا را با یکدیگر به اشتراک بگذارند. این دو تغییر بزرگ و اساسی (در کنار تغییرات کوچک و بی‌شمار دیگر) به مصرف‌کنندگان دامنه انتخاب وسیع‌تری داده‌اند. در این محیط رقابتی بزرگ، صرفاً تولید یک نرم‌افزار با قابلیت‌های مورد نیاز کافی نیست. محصول شما باید زیبا و راحت باشد. امروزه هیچ قدرت بازاریابی و تبلیغاتی نمی‌تواند یک محصول ضعیف را نجات دهد. ولی نرم‌افزاری که به خوبی طراحی شده باشد کمک می‌کند تا کاربرانی که از آن لذت می‌برند، خود تبدیل به مبلغانی برای توزیع هرچه بیشتر آن شوند.

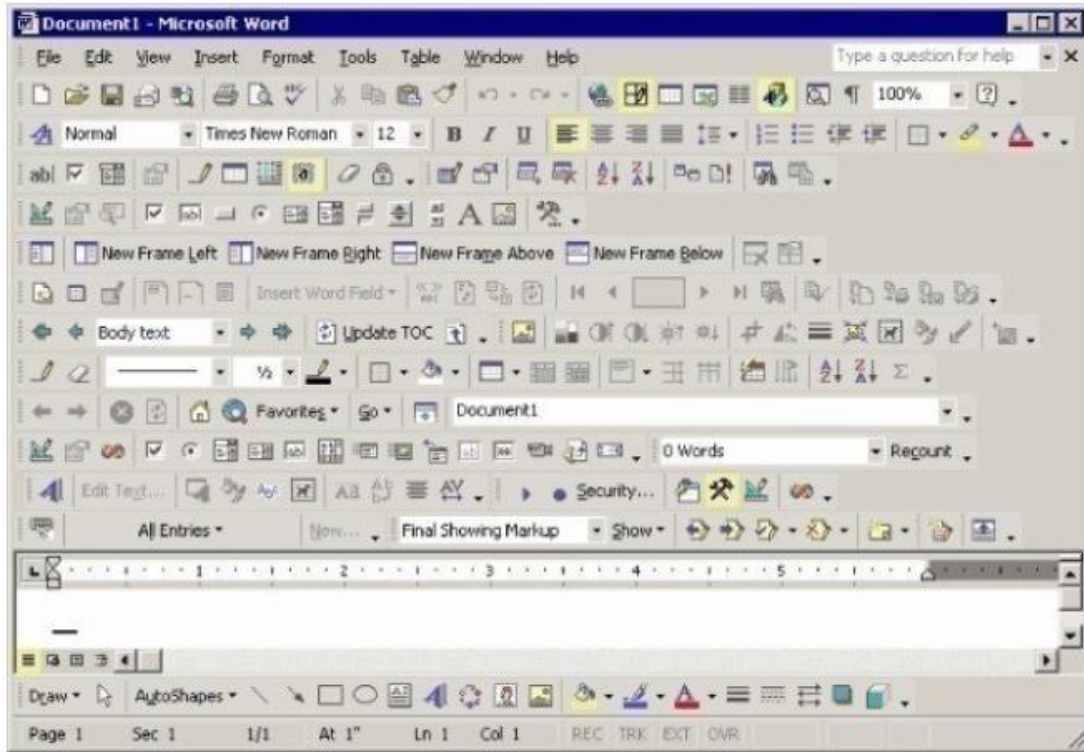
1. WordPerfect
2. Excel
3. Lotus

اولویت اول را به کاربران بدهید

وقتی می‌گوییم اولویت اول با کاربران است، منظورمان این است که شما و تیمتان باید تمام تلاشتان را انجام دهید تا نحوه کار با نرم‌افزار جدید را راحت‌تر کنید. گاهی این کار به معنی حل یک سری مسائل سخت مهندسی است، ولی معمولاً به این معنی است که باید به جای کاربران در مورد طراحی ظاهری نرم‌افزار تصمیمات سختی بگیرید؛ به نحوی که قدرت تصمیم‌گیری بیش از اندازه به کاربران داده شود. به این موضوع «تنبلی محصول» می‌گوییم. تنبلی از نظر بعضی از افراد ویژگی مثبتی برای برنامه‌نویسان تلقی می‌شود؛ چرا که معمولاً برنامه‌نویسان تنبل کارهای بیشتری را به صورت خودکار طراحی می‌کنند. از طرف دیگر، تنبلی در محصول باعث رنج و دردسر زیادی برای کاربران می‌شود. طراحی واسط کاربری‌ای که استفاده از آن برای همه راحت باشد، یکی از بزرگ‌ترین چالش‌های توسعه نرم‌افزار است.

یکی از مثال‌های معمول در تنبلی محصول وقتی است که کاربران بیش از اندازه حق انتخاب داشته باشند. امروزه مردم محصولات ماکروسافت آفیس^۱ در اواخر دهه ۹۰ را که پر از نوارهای دکمه‌های متفاوت بودند مسخره می‌کنند. آن نرم‌افزارها برای راحتی بیشتر! همه امکانات موجود را در قالب دکمه‌های مختلف در اختیار کاربران قرار می‌دادند. یکی از لطیفه‌های طراحان واسط کاربری، برگرفته از همین ایده است، وقتی که بیش از اندازه شدید می‌شود:

1. Microsoft Office



حق انتخاب بیش از اندازه، سرسام‌آور، ترسناک و گیج‌کننده است؛ حتی دربارهٔ اینکه چطور داشتن انتخاب‌های زیاد باعث ایجاد اضطراب و افسردگی می‌شود کتاب‌های مختلفی نوشته شده‌اند.^۱ حتی وقتی می‌خواهید به کاربران امکان تنظیم سلیقه‌ها و علاقه‌شان را نشان بدهید نیز باید وسواس به خرج دهید. آیا می‌دانید ادورا^۲ که یک نرم‌افزار محبوب برای ارسال و دریافت ایمیل بود، سی صفحهٔ مختلف برای تنظیم سلیقه‌ها داشت؟! وقتی که کاربران باید فرمی را با اطلاعاتشان پر کنند، در مورد نوع اطلاعات وارد شده زیاد سخت‌گیرید. خودتان مشکل فاصله‌های اضافه، نقطه‌گذاری و ... را حل کنید. کاربر را مجبور به تجزیه و تحلیل اطلاعات ورودی‌اش نکنید. توهین‌آمیز است وقتی که ببینیم برنامه‌نویس می‌توانسته واسط کاربری را ساده‌تر کند، ولی اهمیتی به این کار نداده است.

۱. به عنوان مثال به کتاب بَری شوارتز با عنوان «تناقض در انتخاب» مراجعه کنید. (Ecco) The Paradox of Choice: Why More is Less

2. Eudora

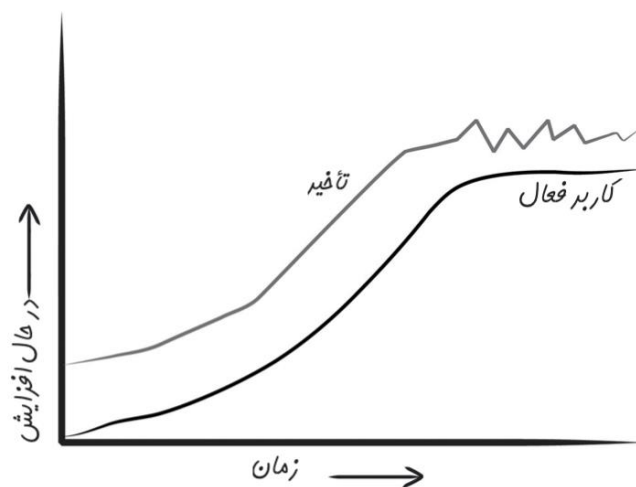
سرعت اهمیت دارد

بسیاری از برنامه‌نویسان اهمیت سرعت نرم‌افزار (یا با بیان علمی‌تر: تأخیر - latency) را دست‌کم می‌گیرند. سرعت نرم‌افزار تأثیراتی اساسی و عمیق دارد.

اولاً، کندی نرم‌افزار خود یک مانع ورود برای کاربران است. سرعت وب‌سایت‌ها ما را بدعادت کرده‌اند. هنگام بررسی یک وب‌سایت جدید، اگر ظرف مدت حداکثر ۳ تا ۴ ثانیه آماده نشود، بیشتر آدم‌ها کلافه شده و مرورگر را می‌بندند. هیچ بهانه‌ای قابل قبول نیست. مرورگر اینترنت کار را برای ما راحت کرده تا بتوانیم وب‌سایتی را ببندیم و ۱۲ جای دیگر برای توجه داشته باشیم. ما کارهای بهتری برای انجام دادن داریم تا اینکه بخواهیم بنشینیم و منتظر بارگذاری وب‌سایتی شویم.

ثانیاً، وقتی نرم‌افزاری با سرعت عمل کند، از لحاظ روانی تأثیر مثبتی روی کاربران خواهد داشت. آن‌ها علاقه‌مند خواهند شد که بیشتر و بیشتر از آن استفاده کنند، چرا که همه‌چیز سلیس و روان به نظر می‌آید. کاربران در ناخودآگاهشان قدرت و توانایی بالاتری حس خواهند کرد. در مقابل وقتی نرم‌افزاری کند باشد، کاربران را کلافه می‌کند و باعث می‌شود که به مرور کمتر و کمتر از آن استفاده کنند، حتی بدون آنکه خودشان متوجه شوند.

مشاهده رشد تعداد کاربران و استفاده آن‌ها از نرم‌افزار در روزهای اول انتشار نرم‌افزار بسیار هیجان‌انگیز است. اما پس از مدتی میزان استفاده محدودتر شده و نمودار رشد متوقف شده و خطی می‌شود. این همان زمانی است که تیم تبلیغات و بازاریابی فریادزنان وارد می‌شوند و درخواست اضافه برای قابلیت‌های جدیدتر، رنگ‌های زیباتر و فونت‌های جذاب‌تر را می‌کنند. گاهی اوقات ولی دلیل اصلی و واقعی توقف در رشد این است که نرم‌افزار کند و سرسام‌آور شده است. همان طور که در نمودار زیر مشاهده می‌کنید، تعامل کاربر با محصول همراه با افزایش تأخیرهای نرم‌افزار، کاهش می‌یابد.



یک حکایت واقعی از گوگل: یکی از تیم‌های مهندسی یک روز بدون اعلام و تبلیغ قبلی، سرعت سرویس نقشه گوگل^۱ را به شدت بهبود بخشید. هیچ اطلاعیه یا حتی پستی در وبلاگ برای این تغییر وجود نداشت و انتشار این تغییر کاملاً پنهانی رخ داده بود. با وجود این، یک افزایش چشمگیر و دائمی در تعداد کاربران سرویس بعد از این تغییر مشاهده شده. یک موضوع روان‌شناسی عمیق و قدرتمند در اینجا نهفته است.

حتی بهبودهای کوچک هم در سرعت وبسایت‌ها مهم هستند. فرض کنید مثلاً صفحه اصلی وبسایت شما ۷۵۰ میلی‌ثانیه طول می‌کشد تا بارگذاری شود. زمان زیادی به نظر نمی‌رسد، درست است؟ برای یک کاربر خاص زمانی کلافه‌کننده نیست؛ ولی اگر بتوانید این زمان را به ۲۵۰ میلی‌ثانیه کاهش دهید، این نیم‌ثانیه افزایش سرعت در مجموع خیلی به چشم خواهد آمد. اگر به فرض یک میلیون کاربر داشته باشید که هر کدام بیست درخواست در روز می‌دهند، با این افزایش سرعت در مدت زمان یک سال، در مجموع شما حدود صد و شانزده سال زمان برای کاربران صرفه‌جویی می‌کنید. وقت کاربران را تلف نکنید! افزایش سرعت، یکی از بهترین راهکارها برای افزایش استفاده کاربران از نرم‌افزار و راضی نگه داشتن آن‌هاست. به قول مؤسسان شرکت گوگل: «سرعت، یکی از قابلیت‌های نرم‌افزار است.»

1. Google Maps

همه کاره نباشید

آیا نرم‌افزار شما تلاش می‌کند که همه کاری انجام دهد؟ ممکن است این سؤال احمقانه به نظر بیاید، ولی بعضی از بدترین نرم‌افزارهای موجود به این دلیل بد هستند که بیش از اندازه جاه‌طلب هستند. تلاش می‌کنند که همه کار را برای همه کس انجام دهند. در سوی دیگر، بعضی از بهترین نرم‌افزارها به این دلیل خوب هستند که صورت مسئله را به خوبی برای خود محدود کرده‌اند و یک راه‌حل کامل ارائه می‌دهند. به جای اینکه تمام مسائل موجود را ناقص برطرف کنند، یک مسئله معمول را برای بیشتر کاربران به خوبی حل می‌کنند.

گاهی تبلیغات بعضی از ابزارآلات خنده‌دار را مسخره می‌کنیم. اینجا را ببینید، یک چراغ‌قوه برای اردو که درون خود یک رادیو هم دارد... و خوب یک تلویزیون هم می‌شود... و ساعت زنگ‌دار و ... تمام این‌ها در کنار هم به شدت گیج‌کننده است. در عوض فرض کنید که نرم‌افزار شما یک اجاق کوچک است. آیا می‌تواند همه چیز بپزد؟ قطعاً خیر؛ ولی بسیاری از غذاهای ساده و محبوب را می‌تواند بپزد و تقریباً برای همه مفید است. مثل یک اجاق کوچک باشید. کمتر بهتر است.



پیچیدگی‌ها را مخفی کنید

ممکن است بگویید: «نرم‌افزار من پیچیده است. مسائل سخت و دشواری را حل می‌کند. چرا باید پیچیدگی‌هایش را مخفی کنیم؟» نگرانی شما منطقی است. ولی در عین حال یکی از اصلی‌ترین چالش‌های طراحی یک محصول خوب دقیقاً همین موضوع است. یک طرح ظریف، موضوعات ساده را ساده و مسائل دشوار را قابل حل جلوه می‌دهد. نرم‌افزار شما حتی اگر روی مسئله پیچیده‌ای کار می‌کند، باید ظاهری مطمئن و یکپارچه داشته باشد. باز هم باید روی احساسات کاربران تمرکز کرد.

دشواری و پیچیدگی نرم‌افزار را پشت صحنه پنهان کنید تا محصول به ظاهر ساده و روان به نظر بیاید.

یک بار دیگر به شرکت اپل دقت کنید. طراحی محصولات در اپل، خارق‌العاده است. یکی از ذکاوت‌مندانه‌ترین کارهایی که اپل در طراحی انجام داد وقتی بود که مسئله مدیریت فایل‌های موسیقی MP3 را با خلاقیت حل کرد. قبل از به وجود آمدن آیفون^۱، بازیچه‌های زیادی سعی کردند که برای مدیریت فایل‌های MP3 راه‌حلی ارائه دهند؛ ولی هوشمندی اپل در این بود که متوجه شد دسته‌بندی این فایل‌ها پیچیده‌تر از آن است که بتوان به راحتی روی صفحه‌های کوچک دستگاه‌های قابل حمل انجام داد. بنابراین با معرفی آی‌تونز^۲ راه‌حل را به صفحه نمایش‌های بزرگ‌تر منتقل کردند. به کمک کامپیوتر خود که صفحه نمایش بزرگ، ماوس و کیبورد راحت‌تری دارد، فایل‌های MP3 خود را مرتب و منظم می‌کنید و از آیفون فقط برای پخش استفاده می‌کنید. در نتیجه آیفون یک دستگاه ساده، ظریف و زیبا خواهد بود و شما هم مشکل مرتب کردن فایل‌های موسیقی را نخواهید داشت.

سرویس جست‌وجوی گوگل یک مثال دیگر از مخفی کردن پیچیدگی‌هاست. از دید کاربران، گوگل تقریباً هیچ مانعی برای ورود و استفاده ندارد. فقط و فقط یک جعبه جادویی ساده برای تایپ کردن وجود دارد. پشت این جعبه جادویی ولی هزاران سرور در اقصی نقاط دنیا به ازای هر یک دکمه‌ای که فشار می‌دهید در حال پردازش و جست‌وجو هستند. وقتی که دکمه جست‌وجو را فشار می‌دهید، جست‌وجوی شما از قبل انجام شده و آماده است. پیچیدگی تکنولوژی و پردازشی که پشت صحنه انجام می‌شود، حیرت‌آور است، ولی تمام آن از دید کاربر پنهان است. مثل یک شعبده عمل می‌کند.^۳ هدف طراحان نرم‌افزار باید پنهان کردن پیچیدگی‌های نرم‌افزاری مثل گوگل باشد که مظهری بدون حریف و کاملاً ساده است.

1. Ipod

2. iTunes

۳. قانون سوم کلارک را مطالعه کنید. https://en.wikipedia.org/wiki/Clarke's_three_laws#Third_Law_and_variants

از همه این حرف‌ها گذشته، باید توجه داشت که در بحث مخفی کردن پیچیدگی باید متوجه یک استثنا هم بود. با وجود اینکه پنهان کردن پیچیدگی همیشه قابل ستایش است، ولی نباید تبدیل به ابزاری برای بستن دستان همه کاربران شود. پوشاندن پیچیدگی‌ها تقریباً همیشه باعث ایجاد لایه‌های انتزاعی خلاقانه می‌شود که مانند پرده روی هسته نرم‌افزار شما قرار می‌گیرند. به عنوان یک مهندس نرم‌افزار باید توجه داشته باشید که این لایه‌ها خواه‌ناخواه روزی بالاخره کنار خواهند رفت. ممکن است این نفوذ چیزی ساده مثل یک پیام ارور ۴۰۴ از سوی سرویس وب باشد. وحشت نکنید! برعکس از چنین نفوذهایی با ارائه راه‌حل‌های مناسب استقبال کنید. یک راه مناسب برای این کار مثلاً این است که برای سایر برنامه‌نویسان API‌های مناسب ارائه دهید تا بتوانند به کمک نرم‌افزارهای خودشان با سیستم شما ارتباط برقرار کنند. یا برای آن دسته از کاربران که دانش فنی بیشتری دارند، یک نسخه یا واسط کاربری مخصوص متخصصان ارائه دهید تا بتوانند روی محصول شما کنترل بیشتری داشته باشند و پرده‌هایی که پیچیدگی‌ها را پوشانده‌اند کنار بزنند.

علاوه بر اینکه واسط کاربری باید قابل انعطاف باشد و قابلیت تغییر شکل داشته باشد، لازم است که داده‌های کاربران نیز همیشه به سادگی در دسترس باشند. فیتز همیشه وسواس زیادی روی «آزادی داده» در محصولات گوگل داشت، به این معنی که کاربران باید بتوانند به راحتی داده‌های خودشان را از نرم‌افزار خارج کرده و هر کاری که دوست داشته باشند با آن انجام دهند. یک نرم‌افزار صرف‌نظر از سادگی و یکپارچگی، هیچ‌وقت نباید کاربران را درون خود زندانی کند. با اجازه دسترسی آزادانه کاربران به داده‌هایشان، منصفانه رقابت کنید. کاربران از نرم‌افزار شما به این دلیل استفاده می‌کنند که دلشان می‌خواهد، نه اینکه مجبورند. سعی کنید تا کاربران به شما اعتماد کنند. اعتماد، باارزش‌ترین دارایی شماست که جلوتر درباره‌اش بیشتر صحبت خواهیم کرد.

مدیریت روابط با کاربران

خوب پس نرم افزار شما توجه کاربران را جلب کرده است. مانع خاصی برای ورود ندارد و کاربران پس از استفاده از آن راضی به نظر می آیند. چند ماه بعد چه می شود؟ با کاربرانی که سال ها روزانه از نرم افزار شما استفاده می کنند چطور ارتباط برقرار می کنید؟

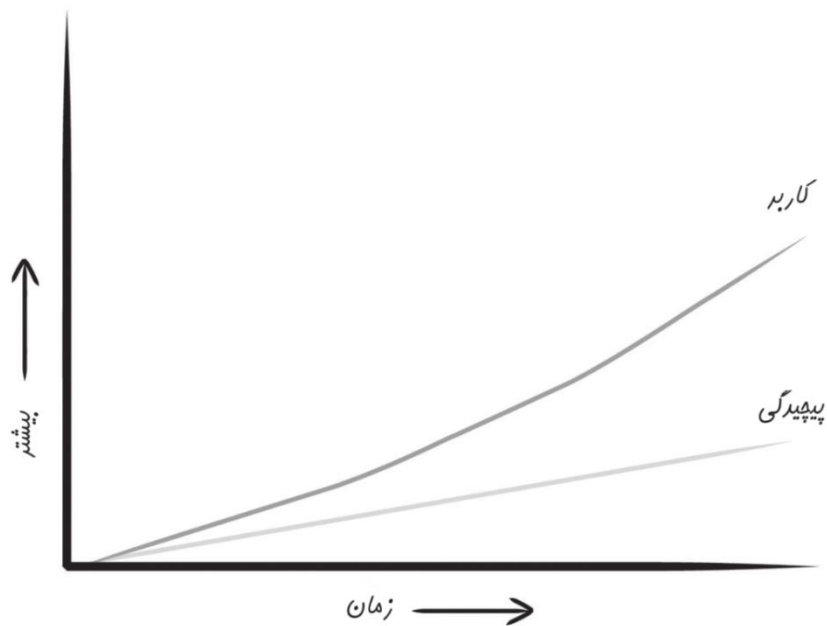
شاید باورش دشوار باشد، ولی خیلی از کاربران دوست دارند که با شرکت و تیم شما ارتباط داشته باشند. آن دسته از کاربران که از محصول راضی هستند می خواهند که در مورد برنامه های آینده شما بدانند و آن گروه که به هر دلیلی از نرم افزار عصبانی هستند به دنبال راهی برای گلایه و شکایت هستند. یکی از بزرگ ترین اشتباهاتی که برنامه نویسان مرتکب می شوند این است که نرم افزار خودشان را پشت دیوار و برای کاربران می اندازند و راهی برای دریافت نظرات و پیشنهادات آن ها ایجاد نمی کنند.

درست مثل تبلیغات و بازاریابی، واژه خدمات مشتری نیز بار منفی مخصوص به خودش را به همراه دارد. بخش خدمات پس از فروش به اشتباه اتاقی پر از افرادِ تلفن به دست را به ذهن انسان می آورد که همه در حال مکالمه با مشتریان هستند. ولی در واقعیت، خدمات مشتری یک شغل جان فرسای پشت میز نشین برای افراد غیر فنی نیست، بلکه یک فلسفه و طرز فکر در نحوه نگرش به نوع رابطه با مشتریان است. این توجه به مشتریان باید در خون شرکت شما، به عنوان یک تیم خلاق باشد و نه صرفاً روشی برای پاسخ به سؤالات و درخواست های کاربران.

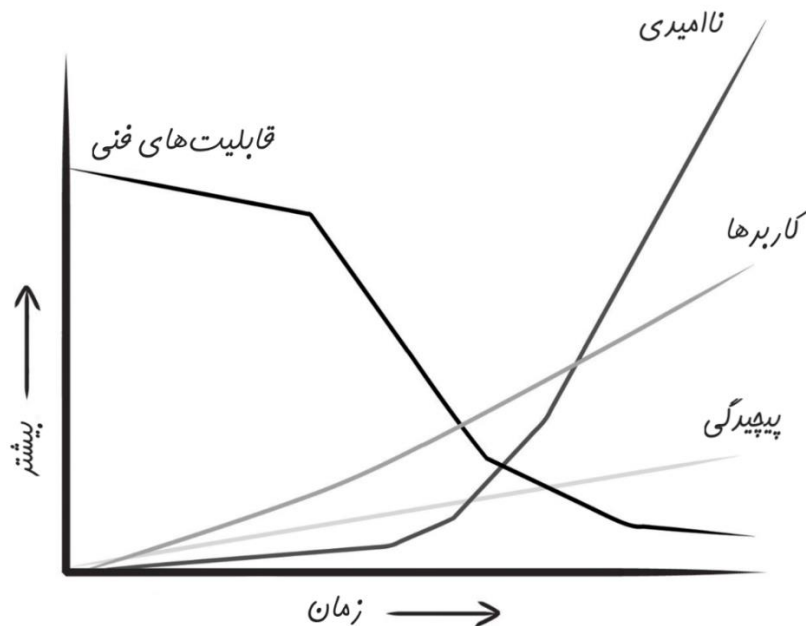
مهندسين معمولاً از تعامل مستقیم با کاربران فراری هستند. معتقدند که «کاربران هیچ نمی فهمند.» فکر می کنند که «همیشه روی مخ هستند و اصلاً نمی شود با هیچ کاربری صحبت کرد.» هیچ کس از شما انتظار ندارد که تک تک کاربران را از عشق و محبت سرشار کنید، ولی حقیقت ساده این است که بدانید کاربران دوست دارند شنیده شوند، حتی اگر پیشنهادات غیرممکن یا انتقادات غیرمنطقی داشته باشند، باید به حرف دل آن ها گوش کرد و به صحبت هایشان توجه داشت. هرچه بیشتر به حرف های آن ها اعتنا کنید، وفادارتر و راضی تر خواهند بود. مجبور نیستید که با آن ها موافقت کنید، ولی لازم است که نظراتشان را گوش کنید. این کار دقیقاً در راستای اصل «احترام» از اصول HRT است. دنیای شبکه های اجتماعی درس های زیادی را به شرکت ها داده است. فقط کافی است تا مثل یک انسان عادی نزد کاربران بروند و نه مثل یک سازمان بزرگ و بی روح. مردم از دیدن شرکت ها وقتی بر مبنای اصول HRT رفتار می کنند، لذت می برند.



اهمیت مدیریت روابط عمومی با کاربران را به کمک یک نمودار (کمی غیرعلمی) این طور نمایش می‌دهیم. با گذر زمان، تعداد کاربران نرم‌افزار شما به مرور افزایش می‌یابد. همچنین هرچه قابلیت‌های نرم‌افزار را «بهبود» می‌بخشید، نرم‌افزار شما پیچیده‌تر می‌شود:



مسئله اینجاست که با افزایش تعداد کاربران، میانگین قابلیت‌های فنی کاربران کاهش می‌یابد، چرا که شما جمعیت بیشتری از آدم‌ها را حمایت می‌کنید. این کاهش قابلیت‌های فنی هم‌زمان با افزایش پیچیدگی نرم‌افزار باعث ایجاد یأس و ناامیدی کاربران می‌شود:



ناامیدی کاربران یعنی افزایش انتقادات، عصبانیت و نیاز تمام‌نشده‌ی به ارتباط مستقیم بین کاربران و برنامه‌نویسان.

چطور می‌توان از این روند جلوگیری کرد؟

اولین قدم این است که وجود این مسئله را باور داشته باشید. بسیاری از شرکت‌ها دیوارهای محکمی بین کاربران و برنامه‌نویسان قرار می‌دهند. تلفن‌های پیغام‌گیر چندمرحله‌ای جلوی کاربران قرار می‌دهند تا نهایتاً یک «درخواست کمک» به نام آن‌ها از طرف اشخاصی که هیچ‌وقت کد را ندیده‌اند ثبت شود. پیام کاربران بین لایه‌های مختلف ردوبدل می‌شود تا مبادا مزاحم وقت برنامه‌نویسان شود و در نتیجه کاربران احساس عدم توجه و ناتوانی می‌کنند و به مرور از نرم‌افزار جدا و دور می‌شوند.

راه بسیار بهتر این است که نیاز به ارتباط با کاربران را قبول کنید و راهی عام، باز و مستقیم مثل گزارش ایراد نرم‌افزار برای آن‌ها فراهم کنید. یک آدرس مناسب ایمیل برای آن‌ها اختصاص دهید یا در صفحات فضای مجازی با آن‌ها مستقیماً ارتباط داشته باشید. اگر کد نرم‌افزار شما می‌تواند به صورت متن‌باز در اختیار همه باشد که چه

بهتر. هرچه از دید کاربران بیشتر شبیه یک آدمیزاد در دسترس و قابل ارتباط باشید، آن‌ها بیشتر به کار شما اعتماد می‌کنند. حواستان را جمع کنید که گاهی کاربران از محصولات شما طوری استفاده می‌کنند که اصلاً انتظارش را ندارید. این کار آن‌ها می‌تواند گاهی حتی زیرکانه و هیجان‌انگیز باشد. فقط و فقط از طریق برقراری یک رابطه مستقیم و دوطرفه است که می‌توانید بفهمید کاربران چطور از نرم‌افزار شما استفاده می‌کنند.

به هوش کاربران احترام بگذارید

به طور پیش فرض همیشه احترام کاربران را حفظ کنید. یک باور غلط که عمدتاً باعث ترس از برقراری ارتباط با کاربران می‌شود این است که آن‌ها آدم‌های باهوشی نیستند. کاربران نرم‌افزار را درست نکرده‌اند، پس حتماً باید از همه چیز بی‌خبر باشند، مگر نه؟ وقتی که فرصت صحبت با کاربران را به دست می‌آورید، اولین و مهم‌ترین اصلی که باید به یاد داشته باشید این است که از غرور و خودپسندی دور باشید و با فروتنی و خضوع رفتار کنید. جُربزه و زبردستی در کار با کامپیوتر، معیار باهوشی و خردمندی نیست. بسیاری از افراد نابغه و با استعداد در دنیا از کامپیوتر فقط به عنوان یک ابزار کار استفاده می‌کنند، نه هیچ چیز بیشتر و دیگری. آن‌ها علاقه‌ای برای به‌کارگیری روش‌های اصولی و علمی حل مسئله‌های الگوریتمی و کامپیوتری ندارند. خیلی از ما هیچ سررشته‌ای در تعمیر قطعات ماشین‌هایمان نداریم. اگر فرض کنید کاربران نرم‌افزار شما گیج و کم‌هوش هستند، درست مثل این است که مکانیک ماشین شما تصور کند شما خنگ و نادان هستید؛ چرا که بلد نیستید جعبه دندهٔ اتومبیل خود را عوض کنید. ماشین شما برای شما مثل یک جعبهٔ سیاه در بسته می‌ماند که شما از آن فقط برای نقل مکان استفاده می‌کنید. برای بیشتر آدم‌ها نیز، کامپیوتر و نرم‌افزار شما مثل یک جعبه سیاه سربسته است. کاربران معمولاً علاقه‌ای به تحلیل و تجزیهٔ آن ندارند؛ صرفاً دوست دارند تا کارشان را انجام دهند و این قضیه هیچ ارتباطی با سطح درک و هوش آن‌ها ندارد!

صبور باشید

نتیجه تمام بحث ما در این بخش این است: عمیقاً صبور باشید. بیشتر کاربران صرفاً لغات مناسب برای ابراز مشکلات نرم‌افزاری خودشان را بلد نیستند. سال‌ها تمرین و تجربه لازم است تا یاد بگیریم آن‌ها دقیقاً چه می‌گویند. کافیست از افرادی که تلاش کرده‌اند مشکلات کامپیوتری پدر و مادرشان را پشت تلفن حل کنند (احتمالاً بیشتر خوانندگان این کتاب!) بپرسید. نیمی از صحبت بر سر این است که روی لغات و کلمات مناسب با آن‌ها هم‌نظر شوید. خیلی از آدم‌ها نمی‌دانند که مثلاً «مرورگر وب» چیست و فرض می‌کنند که بخشی از کامپیوتر آن‌هاست. نرم‌افزارها را به عنوان «یک اتفاق» توصیف می‌کنند یا درباره‌ی آیکون‌های روی صفحه نمایش به صورت یک فرایند اسرارآمیز صحبت می‌کنند. واقعیت این است که باهوش‌ترین افراد نیز راه و روش خودشان را برای درک و توصیف رفتارهای کامپیوتری ابداع می‌کنند. در نتیجه مشکلات کامپیوتری خود را با اصول و قواعدی که فقط در دنیای خودشان معنی می‌دهد توضیح می‌دهند.

پدر: کامپیوتر من خیلی کند شده؛ فکر می‌کنم به خاطر اینه که حافظه‌اش پر شده.

شما: چرا فکر می‌کنی حافظه‌اش پر شده؟ فضای خالی حافظه‌اش را چک کردی؟

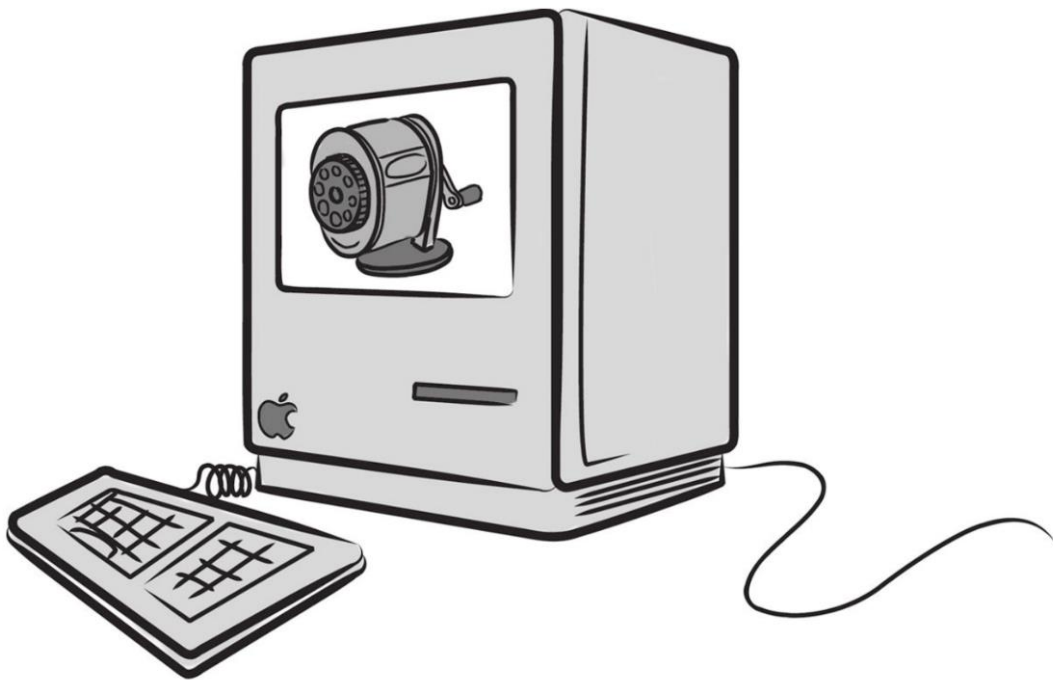
پدر: آره... خوب... بین تمام صفحه نمایش من پر شده از آیکون‌های مختلف. احتمالاً هیچ جایی برای ایمیل‌های من باقی نمونده. شاید لازم باشه که تاریخچه‌ی مرورگر رو پاک کنم؛ نه؟ دفعه‌ی پیش همین طوری درست شد.

شما:؟!

مهارت حیاتی در اینجا این است که یاد بگیرید منظور دقیق پشت کلمات کاربر چیست، نه اینکه تک‌تک واژه‌ها را موبه‌مو دنبال کنید. برای این کار علاوه بر مهارت ترجمه‌ی مفاهیم، نیاز به کمی هوش اجتماعی و ذهن خوانی نیز دارید.

در این راستا فیتز خاطره‌ای شنیدنی دارد. مادر بزرگ فیتز یک روز با او تماس گرفت و پرسید: «برایان، این کامپیوتر قدیمی من هیچ ارزشی دارد؟» فیتز پاسخ داد: «نه، یک مک کلاسیک خیلی قدیمی است و بهتر است که آن را دور بیندازیم.» مادر بزرگ پاسخ داد: «بسیار خوب، الان هم فقط وقتی روشنش می‌کنم که می‌خواهم مدادم را تیز کنم.»

فیتز که کاملاً گیج شده بود، بعد از یک مکث طولانی تصمیم گرفت سؤالات بیشتری بپرسد تا منظور مادر بزرگ را متوجه شود. نهایتاً کاشف به عمل آمد که مداد تیزکن و کامپیوتر قدیمی مادر بزرگ هر دو به یک رابط برق متصل هستند. وقتی مادر بزرگ وارد اتاق می‌شود تا مدادش را تیز کند، رابط برق را وصل می‌کند و باعث می‌شود کامپیوتر نیز روشن شود. بعد از تیز کردن مداد و هنگام خروج از اتاق رابط برق را مجدداً قطع می‌کند و کامپیوتر بیچاره بدون آنکه فرصت کامل روشن شدن را پیدا کند، مجدداً خاموش می‌شود.^۱ این داستان یک مثال خیلی خوب از وقتی است که یک کاربر غیرفنی با دایره لغات محدود خود سعی می‌کند مشکلات کامپیوتری‌اش را توضیح دهد.



گاهی آدم‌ها حتی در مورد موتور جست‌وجوی گوگل باورهای عجیبی دارند. خیلی‌ها فکر می‌کنند که این جست‌وجوگر صرفاً بخشی از کامپیوتر آن‌هاست. سال ۲۰۰۵ وقتی به بقیه می‌گفتیم که در شرکت گوگل کار می‌کنیم با تعجب به ما نگاه می‌کردند و می‌گفتند: «واقعاً؟ فکر نمی‌کردیم کسی در آنجا کار کند!» از سمت دیگر یک بار مادر بزرگ فیتز از اینکه همه کارمندان گوگل به یک سفر دسته‌جمعی می‌رفتند تا اسکی بازی کنند، دل‌خور شده بود (وقتی که گوگل هنوز بسیار کوچک بود) و می‌گفت: «خیلی بده! چطور همه با هم سفر می‌کنند؟»

۱. اگر نگران آن کامپیوتر بیچاره هستید، از آن وضعیت وخیم خارج شده است.

پس کی جواب جست‌وجوهای اینترنتی من رو می‌ده؟! «چقدر رفتار گوگل غیرمسئولانه بود که کارمند کافی برای فشردن دکمه‌های مناسب و پاسخ‌گویی به جست‌وجوهای مادر بزرگ باقی نمی‌گذاشت!

اعتمادسازی و دل‌شاد کردن

دو شعار هنگام برقراری ارتباط با کاربران را هیچ‌گاه فراموش نکنید: اعتمادسازی کنید و کاربران را دل‌شاد کنید.

اعتماد، لغت ظریفی است. هنگام مرور اصول HRT دربارهٔ اعتماد بحث کردیم و روی اهمیت اعتماد به همکاران و هم‌تیمی‌ها تأکید کردیم. اکنون دربارهٔ کسب اعتماد از سوی کاربران محصول صحبت می‌کنیم. اعتماد یک کاربر به شما یا شرکت شما معمولاً در احساسات او ریشه دارد. به ندرت می‌بینید کسی بگوید: «من به فلان شرکت اعتماد دارم، چرا که این لیست طولانی از اصول و حقایق موجود نشانگر حداقل میزان خطر من در مورد ارتباط با آن‌هاست!» کاربران به شما اعتماد می‌کنند، زیرا براینده تمام ارتباطاتی که آن‌ها با شما داشته‌اند برای آن‌ها از نظر عاطفی و احساسی مثبت بوده است.

به دوستان و خانوادهٔ خود توجه کنید، چه تعداد از آن‌ها یک مکانیک اتومبیل می‌شناسند که عمیقاً به او اعتماد می‌کنند؟ این روزها تقریباً کسی به هیچ مکانیکی اعتماد ندارد، چرا که سال‌هاست وقتی برای یک سرویس معمولی ساده مثل تعویض روغن نزد آن‌ها می‌روید، شما را با یک مشت سرویس غیرمنتظرهٔ دیگر روبه‌رو می‌کنند. هیچ‌کس نظر آن‌ها را کامل باور نمی‌کند، زیرا آن‌ها می‌خواهند از هر فرصتی برای حداکثر کردن سود خود استفاده کنند. چیزی به نام «عدم صداقت موقتی» وجود ندارد.

روابط خوب طولانی‌مدت می‌توانند به راحتی قربانی سودجویی‌های کوتاه‌مدت شوند. کمی اینجا و اندکی آنجا از مشتریان خود سوءاستفاده کنید تا نهایتاً شانس اعتماد آن‌ها را در طولانی‌مدت از دست بدهید. از سوی دیگر، هرگاه فقط کمی برای آن‌ها مفید باشید یا درخواست آن‌ها را با جدیت دنبال کنید، اندکی اعتماد درون صندوق بانک فرضی ذهن آن‌ها ذخیره کرده‌اید. در طول سال‌ها رابطه با مشتریان، این صندوق‌های اعتماد به مرور پربارتر و مفیدتر می‌شوند تا وقتی اسم محصول شما می‌آید، کاربران با رضایت قلبی به شما اعتماد می‌کنند.

ولی اعتماد می‌تواند بسیار خطرناک نیز باشد؛ چرا که این قابلیت را دارد که یک‌جا و به کل از بین برود. دقیقاً مثل یک حساب بانکی که به راحتی و با یک خرید احمقانه می‌تواند کاملاً خالی شود. اگر شرکت شما اشتباهی مرتکب شود (حتی ناخواسته) که موجب بی‌احترامی به کاربران شود، صندوق بانکی اعتماد یک‌شبه خالی می‌شود. یک مثال خوب در این راستا مربوط به سال ۲۰۱۱ است که نت‌فلیکس رابطه با کاربران را موقتاً خراب کرد. نت‌فلیکس دو سرویس برای ارائهٔ فیلم دارد: یکی از طریق مشاهدهٔ اینترنتی و دیگری از طریق کرایهٔ دی‌وی دی فیلم‌ها و دریافت آن‌ها به صورت پستی. در طول یک دهه فعالیت، نت‌فلیکس به مرور محبوبیتی بین کاربران کسب کرده بود. ساده، مدرن، راحت و ارزان بود و اوایل سال ۲۰۱۱ نزدیک به ۲۳ میلیون مشتری داشت.

ولی نت‌فلیکس متوجه شد که سرویس پخش اینترنتی و سرویس کرایه دی وی آن‌ها در واقع دو شرکت کاملاً مجزا و با مدل سوددهی و نیاز مدیریتی متفاوت بودند. بنابراین تصمیم گرفتند که این دو را از هم جدا کنند و از کاربران به صورت مجزا هزینه را دریافت کنند که در عمل باعث می‌شد هزینه بعضی از کاربران تا ۶۰ درصد افزایش یابد. مشتری‌ها به شدت خشمگین شدند. در نتیجه نت‌فلیکس اعلام کرد که برای شفافیت و راحتی بیشتر به دو شرکت متفاوت تقسیم می‌شود. این خبر ولی برای کاربران به این معنی بود که باید به جای یکی، دو صورت حساب مجزا پرداخت کنند. وقتی نت‌فلیکس متوجه فاجعه شد، تلاش کرد تا تمام اخبار و اعلامیه‌های خود را پس بگیرد و به وضعیت سابق برگردد، ولی دیگر دیر شده بود. با وجود یک دهه رشد متوالی حدود هشتصد هزار مشتری را طی مدت کوتاه سه ماه از دست دادند. سال‌ها اعتمادی را که به زحمت جمع کرده بودند، با چند تصمیم کوچک و ساده به باد دادند. تصمیماتی که به ظاهر در راستای منافع شرکت بودند، ولی در عمل توجهی به مشتریان نداشتند. خوشبختانه طی سال‌های بعد موفق شدند دوباره اعتماد خود را نزد مشتری‌هایشان به دست آورند و با توجه به نیاز دقیق مشتریان خود، قوی‌تر از قبل بازگردند.

اعتماد، مقدس‌ترین ذخیره شماست. با دقت از آن مراقبت کنید. قبل از هر تصمیمی به صندوق بانکی اعتماد خود نزد مشتریان توجه کنید و ببینید که تصمیم شما چه تأثیری روی آن خواهد گذاشت. روی جلوه طولانی مدت خود تمرکز کنید، نه راحتی‌های کوتاه مدت.

درست شبیه اعتمادسازی، دل‌شاد کردن مشتریان، یکی دیگر از ابزارهای شما برای بهبود روابط با کاربران است. خوشحال کردن آن‌ها وقتی که انتظارش را ندارند، باعث ایجاد همان دلگرمی ناشی از رضایت در مشتریان می‌شود و کمک می‌کند که در دید آن‌ها بیشتر شبیه یک آدمیزاد به نظر بیایید.



برای شروع، به یاد داشته باشید که لازم نیست خودتان را بیش از حد جدی بگیرید. گوگل رسمی داشت که همیشه «روز دروغ آوریل» یک ادعای عجیب و غریب از یک محصول می‌داد. مثلاً یک سال هر ویدئویی که در یوتیوب پخش می‌شد، به یک تکه رقص از یک ویدئوی معروف تبدیل کرده بود. یا به عنوان یک مثال دیگر به www.woot.com توجه کنید. یک وبسایت ساده برای فروش محصولات مختلف با تخفیف‌های روزانه، ولی با این تفاوت که محصولی را که هر روز به عنوان تبلیغ معرفی می‌کند، در کنار شوخی و لطیفه‌های مسخره قرار می‌دهد.

سعی کنید تا کاربران را با تکه‌های کوچک شادی و موضوعات خارق‌العاده، شگفت‌زده کنید. با وجود اینکه گوگل یک شرکت بزرگ بر پایه علوم مطلق کامپیوتر است، هیچ‌چیز به اندازه لوگوهای کارتونی آن که به مناسبت‌های مختلف عوض می‌شوند کاربران را هیجان‌زده نمی‌کند. یک طرح گرافیکی ساده که وارد زندگی روزمره مردم شده، باعث نامه‌های بی‌شمار کاربران و صحبت‌های تمام‌نشده بین آدم‌ها شده است.

القای اندکی وحشت اگر همراه با شوخی باشد نیز گاهی می‌تواند مفید باشد! یک شرکت فضای مجازی برای اینکه کاربران را تشویق کند تا یک عکس از خودشان آپلود کنند، به طور پیش‌فرض عکس دیک چنی^۱ را روی پروفایل آن‌ها قرار می‌داد. مردم برای آنکه عکس دیک چنی از روی پروفایل خود بردارند، خیلی سریع عکس‌های خود را به جای آن آپلود کردند.

کمی شادی و شوخ‌طبعی اگر همراه با استراتژی مناسب باشد، به مشتریان شما نشان می‌دهد که شما به آن‌ها توجه می‌کنید. در نتیجه رابطه شما با کاربران بهتر و صمیمی‌تر می‌شود.

۱. Richard Bruce Cheney؛ متولد ۱۹۴۱ در لینکلن، نبراسکا؛ سیاست‌مدار آمریکایی و چهل‌وششمین معاون رئیس‌جمهور ایالات متحده آمریکا از سال ۲۰۰۱ تا ژانویه ۲۰۰۹ و در دوران جرج دبلیو بوش پسر، در این کشور بود. (و.)

کاربران را فراموش نکنید

نکات متنوعی را در این فصل مطرح کردیم؛ ولی نهایتاً سه اصل ساده را در مورد کاربران به یاد داشته باشید:

- تبلیغات و بازاریابی: به دیدگاه مردم از نرم‌افزار خودتان توجه کنید؛ چرا که عامل اصلی در تصمیم‌گیری آن‌ها برای امتحان کردن یا نکردن محصول است.
- طراحی محصول: اگر نرم‌افزار شما ساده، سریع، قابل استفاده و در دسترس نباشد، کاربران دیگر از نرم‌افزار شما استفاده نخواهند کرد و از آن خارج می‌شوند.
- خدمات مشتری: نحوه ارتباط شما با کاربران در طولانی‌مدت، معیار موفقیت محصول شما خواهد بود.

کار روزانه ما برنامه‌نویسان پر است از عوامل حواس‌پرتی؛ نظیر مرور کدها، جلسات طراحی، مبارزه‌های تمام‌نشدنی با ابزارهای برنامه‌نویسی، خاموش کردن آتش سرورها، دسته‌بندی کارها. به راحتی می‌توان فراموش کرد که چرا ما به دنبال توسعه نرم‌افزار هستیم. هدف نه شما هستید، نه تیمتان و نه شرکتی که برای آن کار می‌کنید. هدف این است که زندگی را برای کاربران نرم‌افزارتان ساده‌تر کنید. بنابراین لازم است که به صحبت‌های آن‌ها توجه کنید، نحوه فکر کردنشان را درک کنید و تجربه آن‌ها با نرم‌افزار را مشاهده کنید. کاربران، خون جاری در رگ‌های نرم‌افزار شما هستند. هر آنچه را بکارید درو خواهید کرد.