

## مخفي کاري در برنامه نويسي، عامل ضرر و زيان

اگر شما تمام کارتان را در خلوت تنهائي خودتان انجام بدھي، خطر شکست و همینطور احتمال عدم پيشرفت خودتون رو به شدت افزايش می دهد.

اولا، اصلا از کجا می دانيد که شما در مسیر درستی قرار داريد؟

تصور کنيد شما يك سازنده و طراح دوچرخه هستيد. يك روز يك اиде خارق العاده و ناب يك طرح تازه برای دنده دوچرخه به ذهنتان ميرسد. قطعاتي که لازم داريد را سفارش داده و هفته ها در کارگاهتان روی ساخت يك نمونه اوليه، سخت تلاش ميکنيد. وقتی همسایه شما، که اتفاقاً او هم در ساخت دوچرخه سررشيته دارد از شما درمورد کاري که ميکنيد سوال می پرسد، شما چيزی توضیح نمیدهيد. شما نمی خواهيد که تا زمانی که کارتan کامل و بی نقص تمام نشده، کسی در مورد پروژه شما چيزی بداند. چند ماه دیگر ميگذرد و شما برای ساخت نمونه کارتan دچار مشکل می شويد. ولی خوب چون تمام را مدت رو مخفیانه کار کردید، کمک گرفتن از دیگران در اين مرحله دیگر غيرممکن است. يك روز ناگهان همسایه رو می بینيد که دوچرخه خودش را از پارکينگ بيرون آورده و روی دوچرخه اش يك دنده خيلي مدرن و خاص سوار کرده است. کاشف به عمل می آيد که اتفاقاً او هم روی يك اиде خيلي مشابه به اиде شما کار می کرده ولی از چند نفر دیگر از دوستان و همکاران خودش هم کمک گرفته است. اينجاست که شما دیگر اعصابتان به هم ميريزد. کار خودتان را به همسایه نشان می دهيد و او خيلي سريع به شما نشان ميدهد که کار شما کجاها چه اشکالاتي دارد. اشکالاتي که احتمالا همان هفته اول حل می شدند اگر همان ابتدا همسایه را در جريان پروژه خود گذاشته بوديد.



چند نکته در اين حکایت نهفته است. اگه شما اиде ناب خودتان را از بقیه مخفی نگه داريد و از نشون دادن کار در حال پيشرفتتan به بقیه خودداری کنيد، در حال قمار بزرگی هستيد. اشتباهات اساسی در طراحی های اولیه خيلي به راحتی پيش می آيند. ممکن است که خيلي راحت دوباره چرخ را اختراع کنيد! همچنین فواید استفاده از همکاري بقیه را نيز از دست می دهيد. دقت کرديد که چطور همسایه شما دنده دوچرخه را با کمک همکارهايis سريع تر از شما پياده سازي کرد؟

مردم به همين دليل است که قبل از ينكه در آب شيرجه بزنند، دمای آب را با نوك پنجه های پايشان چک ميکنند. قبل از شروع هر کاري لازم است که مطمئن شويد که اولاً شما روی موضوع درستی کار ميکنيد، ثانياً نحوه اجرای کارتan

صحیح است، و ثالثاً این کار قبل انجام نشده باشد. احتمال سوتی دادن در ابتدای کار بسیار بالاست. هر چه بیشتر از بقیه نظرشان را بپرسید، احتمال اشتباهات اولیه را کمتر میکنید.<sup>۲</sup>

اینکه کارتان را از ابتدا با بقیه به اشتراک بگذارید و نظرشان را بپرسید، فقط باعث کاهش اشتباه و جلوگیری از کج رفتن نمیشود. بلکه باعث تقویت خصوصیتی در پروژه شما میشود که ما اسم آن را می‌گذاریم: ضریب اتوبوسی<sup>۳</sup>!

ضریب اتوبوسی: برابر است با تعداد افرادی که لازم است تا با یک اتوبوس تصادف کنند تا پروژه شما از بین برود.



دانش و اطلاعات چقدر بین افراد تیمان پخش شده؟ اگر شما تنها کسی هستید که جزئیات یک نمونه کار را بلد هستید، هرچند که ممکن است امنیت شغلی 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 کردید. و اگر توانسته باشید که یک تیم را در طراحی و پیاده سازی 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 نخواهند کرد، ولی اتفاقات پیش بینی نشده دیگری ممکن است پیش بیایند. مثلاً یک نفر ممکن است که ازدواج کند، یا به شهر دیگری منتقل شود، یا از شرکت استعفا دهد، یا مجبور شود از یک عضوی از خانواده که مریض شده مراقبت کند. نکته بحث این است که شما باید در مورد پروژه این آینده نگری را داشته باشید و با بالا بردن ضریب اتوبوسی تیم احتمال موفقیت پروژه را افزایش دهید.

صرف نظر از بحث ضریب اتوبوسی، مسئله سرعت کلی کار هم حائز اهمیت است. به راحتی می‌شود فراموش کرد که چقدر سرعت کار کردن انفرادی پایین است. خیلی پایین تر از میزانی که آدم ها دوست دارد به آن اعتراف کنند. وقتی تنها کار میکنید، چقدر چیز جدید یاد می‌گیرید؟ سرعت کارتان چقدر است؟ اینترنت دنیای اطلاعات است، ولی به هیچ وجه جای تجربه انسان ها رو نمیگیرد. کار کردن در تیم باعث می‌شود که خرد جمعی پشت یک پروژه افزایش پیدا کند. وقتی روی یک مساله ای گیر میکنید، چقدر از وقتتان را هدر می‌دهید تا خودتان را از چاله ای که در آن افتادید 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 که کجا اشتباه کرده اید و چطور می‌توانید مسئله را حل کنید. دقیقاً به همین دلیل است که اعضای تیم کنار هم می‌نشینند (یا بصورت دو نفره برنامه نویسی میکنند).

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

اجازه دهید یک مثال دیگر بزنم. فرض کنید شما در حال کد نویسی با یک زبان برنامه نویسی که compiler دارد هستید. آیا روزها می نشینید و ده هزار خط کد می نویسید و وقتی مطمئن شدید که همه چیز را کامل و بی نقص تمام کردید، دکمه کامپایل رو برای اولین بار فشار می دهید؟! البته که نه! چه فاجعه ای به بار می آید. ما برنامه نویس ها با تغییرات کوچک که بتوانیم سریع نتیجه آنها را ببینیم، خیلی بهتر کار میکنیم. یکتابع می نویسیم، کامپایل می کنیم، یک تست به آن اضافه می کنیم، دوباره کامپایل می کنیم، کمی کد را تغییر می دهیم، مجدداً کامپایل می کنیم. اشتباهات تایپی و اشکالات کد را بلافضلله حل میکنیم. دائمآ نیاز داریم که به کامپایلر تکیه کنیم تا هوای ما را داشته باشد. این طور می توانیم کیفیت کدمان را بالا ببریم و نرم افزار را کم کم بهتر و قوی تر کنیم.

این نیاز به پیش بردن کار در تیکه های کوچک، نه فقط در سطح کدنویسی بلکه در سطح کل پروژه نیز وجود دارد. تغییرات اجتناب ناپذیرند و پروژه ها باید دائمآ خود را با این تغییرات تطبیق دهند. پروژه ها معمولاً با موانع از قبل پیش بینی نشده ای برخورد می کنند. گاهی اوقات کلاً متوجه می شویم که هیچ چیز طبق برنامه ای که در نظر داشتیم پیش نمی رود. نیازهای پروژه ناگهان تغییر می کنند. چطور ما می توانیم به سرعت برنامه ها و طرح هایمان را با تغییرات جدید تطبیق دهیم؟ جواب: با کار کردن بصورت تیمی.

اریک ریموند یک جمله معروف دارد:

وقتی تعداد چشم ها زیاد شوند، باگ ها و ایرادهای نرم افزار نمایان می شوند.

شاید این جمله را بتوان کاملتر و بهتر این طور بیان کرد: وقتی تعداد چشم ها زیاد شوند، پروژه شما از مسیر درست خارج نمی شود.

آدم هایی که توی غارتنهایی خود کار میکنند، روزی بیرون می آیند و می بینند که با وجود این که کارشان را به اتمام رسانندن، ولی دنیای بیرون غار تغییر کرده و ایده اولیه پروژه آنها دیگر در این دنیای جدید معنی نمی دهد.

### مهندسين و دفترکار شخصي

بیست سال پیش، ذهنیت کلی آدم ها 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 اون زمان این بود که این دفتر شخصی، باعث می شود که یک برنامه نویس مدت زمان بیشتری را بتواند برای خودش داشته باشد و فقط روی کد زدن تمکز کند. ولی ما فکر می کنیم نه تنها اتاق و دفتر شخصی برای بیشتر مهندسین نرم افزار اصلا ضروری نیست، بلکه حتی خطرناک هم هست. امروزه، نرم افزارها توسط تیم ها نوشته می شوند، نه اشخاص. داشتن یک کانال ارتباطی قابل اتکا با سایر اعضای تیم، حتی از داشتن اتصال پرسرعت به اینترنت هم مهم تر و ضروری تر است. تمام وقت دنیا را هم اکر بدون هیچ مزاحمی داشته باشی، اگر روی کار اشتباهی تمکز کرده باشی، وقتی رو تلف کردي.

متاسفانه به نظر می آید که شرکت های مدرن دنیای تکنولوژی، این روزها بعضاً از آن سمت بوم افتاده اند. وقتی وارد شرکت آن ها می شویم، ۵۰ یا حتی تا ۱۰۰ نفر را می بینید که در سالن های بسیار بزرگ در کنار هم کار می کنند. این طرح پلان آزاد و باز در دفاتر مهندسی این روزها موافق و مخالف های خودش را دارد. همه از کوچک ترین مکالمه ها خبردار می شوند و آدم ها برای صحبت کردن احساس راحتی نمی کنند چون نگران هستند که باعث اذیت و آزار بقیه شوند. این هم به اندازه دفترهای شخصی مضر است.

ما معتقدیم که حد وسط این دو حالت احتمالاً بهترین راه حل باشد. تیم ها را به گروه های ۶ تا ۱۲ نفره تقسیم کنید و هر تیم را در اتاق مخصوص خودشان قرار دهید. بدین صورت صحبت های خود را در کنار هم راحت تر شکل می گیرند و کسی از حرف زدن خجالت نمی کشد و باعث آزار دیگران هم نمی شوند. قطعاً در این حالت هم مثل هر حالت دیگری، تک تک افراد همچنان نیاز دارند که جلوی سرو صدای مزاحم را بگیرند. به همین دلیل است که خیلی از آدم ها راه حل های خلاقانه ای پیدا کردنند تا بتوانند به بقیه نشان دهند که در کاری عمیق شدند و نمی خواهند که کسی مزاحم آن ها شود. ما در تیمی کار میکردیم که بین خودمان یک قانون گذاشته بودیم که هر موقع هر کسی با فرد دیگری کار داشت اسمش رو صدا میکرد و میگفت Breakpoint. اگر فرد مورد نظر فرصت صحبت کردن داشت با صندلی خود من چرخید و با هم صحبت میکردند و اگر فرصت نداشت، میگفت Ack به این معنی که در اولین فرصتی که پیدا

میکرد می آمد و با همکارش صحبت میکرد. تیم های دیگری را هم دیدیم که از هدفون های مخصوص Noise Cancellation استفاده می کردند تا سرو صدای محیط پیرامون رو حداقل کنند. به طور کلی خود عمل گذاشتند هدفون روی سر می تواند یک نشانه تلقی شود ازین که "الآن مزاح من نشوید مگر اینکه کارتان خیلی واجب باشد". حتی تیم هایی را هم دیدیم که از عروسک یا وسایل دیگری استفاده میکردند تا نشان دهنده در حال حاضر سرشان شلوغ است و یا در کاری عمیق شده اند.

منظور ما را اشتباه برداشت نکنید. بله ما معتقدیم که برنامه نویس ها نیاز به بازه های زمانی بدون وقفه و مزاحم برای عمیق شدن در برنامه نویسی دارند. ولی در عین حال عمیقاً انتظار داریم که آن ها باید یک کانال ارتباطی قوی و همیشه در دسترس با سایر هم تیمی های خودشان داشته باشند. هنر واقعی، پیدا کردن حد وسط بین این دو نیاز است.

خوب نتیجه این شد که: ریسک تنها یک کار کردن ذاتاً از کار کردن با بقیه بیشتر است. به جای اینکه از این نگران باشید که یک نفر ایده کار شما را بذدد یا نحوه کار شما را قضاوت کند، خیلی بیشتر باید ازین بترسید که وقتتان را روی یک کار اشتباه تلف کنید. متاسفانه، مسئله مخفی کردن ایده ها فقط محدود به رشته مهندسی نرم افزار نیست. این یک مشکل فراگیر در همه رشته هاست. به عنوان نمونه، 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 دست یابی به بودجه های تحقیقاتی باعث نتیجه عکس شده است. متفکران بزرگ ایده های خود را با بقیه تقسیم نمی کنند. به سختی در تلاش برای پنهان کردن کارهایشان هستند و در غار تنها یک خود به تحقیق می پردازند و از ایرادهای کارشان بی خبر می مانند. نهایتاً مقاله خودشان را چاپ میکنند و طوری وانمود میکنند که انگار پشت کارشان تلاش و کوشش بی وقفه ای نبوده. نتیجه ها عموماً فاجعه بار هستند: یا به طور اتفاقی کار یک نفر دیگر را تکرار کرده اند، یا کارشان ایرادهایی دارد که متوجهش نبودند، و یا نهایتاً کاری را عرضه کردنده که هر چند در زمان شروع موضوعی ناب و جالب بوده، ولی الان دیگر بی معنی و بی فایده هست. میزان زمانی که توی این کار تلف میشود، بسیار غم انگیز اس. شما یکی مثل بقیه نباشید!

---

<sup>۱</sup> مخصوصاً اگه واقعاً سازنده دوچرخه باشید :)

<sup>۲</sup> البته نظرسنجی بیش از اندازه در ابتدای کار پروژه می تونه خطرناک باشه. در مورد این موضوع در فصل های آینده بیشتر صحبت خواهیم کرد.