نرم افزار
سال خط دید
سال دوم • شماره ۶
شهریور ماه سال ۱۳۹۷
قیمت: ۱۵۰۰۰ تومان

- دوبارهم خیلی است!
- مدیر پروژه کیست؟
- مقایسهٔ کانبان با اسکرام کدام بهتر است؟

مدیریت پروژه از تئوری تا عمل

- 10 راهکاربرای مالکان محصول درچارچوب اسکرام
- PMOها چگونه عملکرد اجرایی پروژه را بسنجند؟
- چرامدیرپروژه به انعطاف‌پذیری نیازدارد؟

خط دید

www.Khatedid.com

- تولید محتوای تخصصی
- انتشار محتوا در مجله و سایر رسانه‌های خط دید
- مصاحبه با سازمان‌ها و شرکت‌های موفق در حوزه فناوری اطلاعات
- ارائه گزارش از موفقیت‌های صورت گرفته در حوزه فناوری اطلاعات
- انتشار مجله سراسری
- حضور هدفمند و موثر در شبکه‌های اجتماعی در راستای رسالت مجموعه
- برگزاری دوره‌های آموزشی درون سازمان
- برگزاری کارگاه‌های آموزشی در حوزه مدیریت پروژه
- ارائه مشاوره برنامه‌ریزی و کنترل پروژه
- استقرار تیم مدیریت پروژه در شرکت‌ها

میدان ونک، خیابان برزیل، تقاطع خیابان توانیر، بن بست امید، پلاک ٦، واحد ١

تلفن : ٧٠ - ٨٨٦٥٩٤٦٨ ایمیل : info@khatedid.com

بن ریل

BONRAIL

WWW.BONRAILCO.IR

فراپیام

گروه مهندسی نرم‌افزار

فراپیام پیشرو در صنعت نرم افزار

طراحی و مجری سیستم های جامع، اداری و مدیریت

- سیستم های یکپارچه
- نرم افزار فروشگاه اینترنتی
- سیستم های نظام جامع
- اتوماسیون اداری تحت وب
- نرم افزار خدمات پس از فروش
- نرم افزارصورت های مالی
- داشبورد مدیریتی تحت وب

۸۸۷۶۷۷۱۱

www.farapayam.com

ERP

Enterprise Resource Planning

نوماتک
NOMATEC

آیا زمان آن فرا نرسیده که راهکارهای نرم افزاری خود را برون سپاری کنید؟

با استفاده از نیروی انسانی متعهد، متخصص و ماهر، و با بهره‌جویی از پیشرفته‌ترین فناوری ها، می‌کوشیم در تولید و توسعهٔ نرم‌افزارهای ERP، به بهترین شکل، به نیاز مشتریان داخلی پاسخ گوییم و با پیشتازی در امر صادرات، مرزهای دانش داخلی را گسترش دهیم.

خدمات ما :

مدیریت منابع انسانی

مدیریت زنجیره تامین

مدیریت ارتباط با مشتری

مدیریت منابع انسانی

مدیریت تولید و برنامه‌ریزی

مدیریت مالی و حسابداری

مدیریت حمل و نقل

اتوماسیون اداری

ارائهٔ راهکار برحسب نوع کسب‌وکار:

صنایع تولیدی

شرکت‌های بازرگانی

ارائهٔ راهکار برحسب مقیاس کسب‌وکار:

کسب و کارهای بزرگ

کسب و کارهای متوسط

+۹۸ ۲۱ ۲۵۷۱۶

info@nomatec.net

www.nomatec.net

سازمان مدیریت صنعتی
پیشرو در ارائه خدمات مدیریت
DBA
MBA
Post-DBA
Post-MBA
دوره های تخصصی
مدیریت
آموزش‌های مجازی
آدرس :نشانی: تهران خیابان ولیعصر(عج) نبش جام جم
۰۲۱ - ۲۲۰۴۳۰۰۵
INFO@IMI.IR

SHARIFSSL
SHARIFSSL.COM

شریف SSL ارائه دهنده خدمات صدور گواهی نامه معتبر SSL

تائید در سطح دامنه (فوری)، گواهی نامه نوار سبز (GREEN BAR)، تائید در سطح سازمانی و شرکتی (درج مشخصات سازمان در گواهی نامه)

30% تخفیف با کد
KHATEDID

۰۲۱ - ۸۸۶۵۹۴۶۸
تهران، میدان ونک، خیابان برزیل، تقاطع توانیر، بن بست امید، پلاک ٦، واحد ١

تسکولو

نرم افزار مدیریت پروژه آنلاین برای شرکت‌ها

تسکولو سیستمی امن و آسان برای مدیریت کار و پروژه هاست که بهره‌وری و بازدهی تیم شما رو افزایش میده

- تیمی روی پروژه‌ها کار کنید
- به سادگی و زیبایی گزارش بگیرید
- امنیت رو فدای سادگی نکنید

WWW.TASKULU.COM

همین امروز با مشاوران ما صحبت کنید

TaskuluHQ Taskulu +TaskuluHQ Taskulu_ir

مدرس آنلاین

هر زمان و هر مکان، پای درس بهتـــــرین استاد

WWW.MODARES.ONLINE

چرا مدرس آنلاین؟

با مدرس آنلاین ارزش‌های زیادی را یک‌جا دریافت کنید.
آموزش‌های دلخواه شما در هر زمان و مکانی در دسترس شماست.
با مدرس آنلاین، برای بهترین آموزش‌ها کمترین هزینه را بکنید!

«به‌جای کلاس»، چرا استفاده از «فیلم‌های آموزشی»؟

- هزینه بسیار اندک
- مدیریت مؤثر زمان
- آسایش آموختن در زمان و مکان دلخواه
- عدم نیاز به جابه‌جایی برای حضور فیزیکی در کلاس
- مقرون‌به‌صرفه بودن
- امکان بیشتر برای مرور آموخته‌ها
- تنوع مدرسان و دوره‌ها
- بهره‌وری بالا

ویـــدئوبوک راهی نـــوین

برای افزودن مالتی مدیا به **کتـــب** می باشد.

کاربران با استفاده از اپلیکیشن ویدئو کتاب می توانند نسبت به اسکن کردن بارکد تصویری درج شده در کتاب ها اقدام کنند و فایل ویدئو یا صوتی مرتبط با آن را به سادگی دریافت و یا خریداری کنند.

هر کتاب می توانند چندین بارکد تصویری در بخش های مختلف داشته باشد و خواننده در هر بخش از کتاب، به مالتی مدیا مربوطه دسترسی پیدا می کند.

روزانه

۵.۰۰۰.۰۰۰.۰۰۰.۰۰۰

جستجو انجام می شود دیده شوید !

دیجیتال مارکتینگ و بهینه سازی تخصصی سایت

ارائـــه مشـــاوره و آمـوزش هـــای سازمـــانی

ارائـــه سرویس های سئـو کـــلاه خاکـــستری

آدرس: تهران، میدان ونک، خیابان برزیل، تقاطع توانیر، بن بست امید، پلاک ۶، واحد ۱ ○ شماره تماس: ۸۸۶۵۹۴۶۸-۰۷ ۰۲۱ ○ وب سایت: www.dadehpardaz.com/seo

نرم افزار خط دید

فهرست

مدیریت پروژه از تئوری تا عمل
شماره ۶ • شهریور ماه ۱۳۹۷

اولین قدم در دفتر مدیریت پروژه ۱۱
مقایسه‌ی چابک با آبشاری ۱۲
آیا تیم شما وقت همه را تلف می‌کند؟ ۱۴
آیا «مستندسازی Agile» یک ترکیب ضد و نقیض است؟ ۱۶
چهار کلید برای درک چرخه حیاتی مدیریت پروژه ۹۸
فرهنگ‌سازی، عامل ماندگاری ۲۰
پرینس ۲ یا چابک؟ ۲۱
۹ ابزار ضروری برای تیم‌های چابک (Agile) توسعهٔ محصول ۲۲
۱۰ راهکار برای مالکان محصول در چارچوب اسکرام ۲۴
PMOها چگونه عملکرد اجرایی پروژه را بسنجند؟ ۲۷
۵ راهنمایی برای برنامه‌ریزی پروژه ۲۸
مدیر پروژه کیست؟ ۳۰
مدیریت پروژه ۳۲
چه کنیم که بودجهٔ یک پروژهٔ آی‌تی را به باد ندهیم ۳۵
مقایسهٔ کانبان با اسکرام کدام بهتر است؟ ۳۸
چگونه با MS Project یک محیط چندپروژه‌ای را نجات بدهیم؟ ۴۰
برگزاری دوره‌های کارآموزی مدیریت پروژه در بریتانیا ۴۲
هوش تجاری و کاربردهای آن ۴۴
نگاهی به موقعیت‌های شغلی فناوری اطلاعات در سال ۲۰۱۷ ۴۶
چرا مدیر پروژه به انعطاف‌پذیری نیاز دارد؟ ۴۸
دوبار هم خیلی است! ۵۱
چگونه می‌توان یک برنامه‌نویس فوق‌العاده شد؟ ۵۲
مدیران پروژه دنیا را تغییر می‌دهند ۵۵
کتاب جدیدی در زمینهٔ برنامه‌ریزی از APMG ۵۶
نرم‌افزار یادگیری گوگل می‌آموزد که نرم‌افزار یادگیری بنویسد ۵۸
چگونه اعتبار رهبری بیابیم؟ ۶۰
محمد بهنیا کارشناس سئو و دیجیتال مارکتینگ ۶۲
میلاد نکوفر توسعه‌دهنده نرم‌افزار وب و موبایل ۶۴
مصاحبه با ویکتور تامپسون برنامه‌نویس ۶۶
مت مارکوئیز طراح وب و توسعه‌دهنده ۶۸
جِف دِین توسعه‌دهنده، محقق (گوگل) ۶۹
شِلی اُلْسن توسعه‌دهندهٔ بازی، سازندهٔ دوبازی ۷۰
آیا می‌توانید درست تصمیم بگیرید؟ ۷۱

صاحب امتیاز: داده‌پرداز پویای شریف
مدیرعامل: مهدی استوی
مدیر مسئول و سردبیر: محمدرضا اسماعیل طهرانی
مدیر اجرایی: امیر تیمور نژاد
گرافیک و صفحه‌آرایی: امیر شریف
واحد جذب آگهی: بازرگانی نشریه خط دید
اعضای تحریریه: محمدرضا اسماعیل طهرانی، محمد بهنیا، علیرضا فیروزی

آدرس: تهران، میدان ونک، خیابان برزیل، تقاطع توانیر، بن‌بست امید، پلاک ۶، واحد ۱
تلفن دفتر نشریه: ۰۲۱-۸۸۶۵۹۴۶۸-۷
نمابر: ۰۲۱-۸۸۶۵۹۴۶۸

www.khatedid.com
info@khatedid.com
telegram.me/khatedid
www.instagram.com/khatedid

لیتوگرافی، چاپ و صحافی: چاپ ایران کهن

» تحریریه در رد، تلخیص و ویرایش مطالب ارسالی مجاز است.
» مسوولیت مطالب و آگهی‌ها به عهده نویسنده و سفارش‌دهنده است.
» چاپ مطالب و تصاویر مجله بدون اجازه کتبی مجاز نمی‌باشد.
» مطالب دریافت شده بازپس داده نمی‌شوند.

نویسنده: محمدرضا اسماعیل طهرانی

اولین قدم در دفتر مدیریت پروژه

چالش همیشه در همه پروژه‌ها وجود دارد، مسئله اصلی روش برخورد با آن است.
اما چالش چیست؟ چالش واقعی چیست؟ بهترین راه‌حل چیست؟ معتقدم مسئله اصلی یافتن راه‌حل نیست. زیرا همه چالش‌ها پروژه را دچار اختلال یا ناهماهنگی نمی‌کنند که نیاز به راه‌حل داشته باشد، گاهی برخی از چالش‌های ایجادشده، منجر به ایجاد انگیزه، ایجاد بهره‌وری بالاتر، ایجاد رقابت و یا حتی جمع‌آوری اطلاعات بهتر می‌کند.
لذا معتقدم نحوه نگاه ما به مسائلِ رخ‌داده، جبهه گرفتن افراد دخیل در مسئلهٔ پیش‌آمده و حتی نحوه مذاکره برای تصمیم‌گیری مناسب در مثبت بودن یا نبودن یک چالش مؤثر است.
دقت کردید؟ در هیچ کجای این متن از کلمه مشکل استفاده نکردم، زیرا مشکل کلمه‌ای است که بار منفی دارد، به‌جای آن از مسئله استفاده کردم که به‌ناچار این کلمه ما را یاد کلمه مناسبی، نظیر راه‌حل می‌اندازد. دید ما به چالش‌ها با رعایت همین مسائل ساده می‌تواند متفاوت باشد.
همیشه به یاد شعر زیبای سهراب سپهری باشیم که می‌گفت: چشم‌ها را باید شست، جور دیگر باید دید...

مقایسه‌ی چابک با آبشاری

چرا دوندگان دوی امدادی باید راگبی بیاموزند؟

۱

بگذارید تیم‌های آبشاری (Waterfall)، آبشاری بمانند و تیم‌های چابک (Agile)، چابک بمانند.

به نظر می‌رسد که چابک تمام دنیا را اشغال می‌کند. هر حرکتی که با یک مانیفست آغاز می‌شود احتمال جذب طرفداران مختص به خود را دارد؛ طرفداران چابک معتقدند که چابک بی‌همتا و تنها راه برای جلو بردن توپ در میدان است. اما در حقیقت هر دو روش هنوز جایگاه خود را دارند. اهمیت و تأثیرگذاری رویکرد چابک بر کسی پوشیده نیست؛ با این حال شرکت‌ها، همچنان متوجه منافع فوق‌العاده‌ی ساختارها و پیش‌بینی‌پذیری اکثر فرایندهای سنتی آبشاری هستند. مطابق گزارش سال ۲۰۱۴ از CRASH «در میان تمام روش‌ها، روش‌های ترکیبیِ چابک و آبشاری رتبه‌های بالاتری را از خودِ چابک یا آبشاری دارند.» روش‌های ترکیبی در صورتی امکان‌پذیرند که تمام تیم‌ها مادام خدمت به خود، نیازهای یکدیگر را درک کنند. وقتی پروژه‌هایی که با آبشاری شروع شدند به تیم چابک داده می‌شوند، این پروژه‌ها اکنون پروژه‌های چابک هستند. وظیفه‌ای که ماهیت آن تکرارشونده است، وقتی که در چابک است، نباید با وابستگی‌های دیگر تداخل بیابد؛ در غیر این صورت چابک نیست فقط یک وظیفه آبشاری دیگر است. به خاطر داشته باشید که چابک و آبشاری گزینه‌های مناسبی برای کار متناسب با خود هستند. در تلفیق دو روش باید به چابک و آبشاری وزن یکسانی را داد. کلمه‌ی اسکرام در لغت‌شناسی چابک، مخفف کلمه‌ی "scrummage" است و از بازی راگبی گرفته شده؛ این کلمه اولین‌بار در مقاله‌ی تأثیرگذاری در سال ۱۹۸۶ با عنوان «بازیِ بسیاربسیار جدید توسعه‌ی محصول» وارد جهان مدیریت پروژه شد؛ در این مقاله گفته شد: «رویکرد کل‌گرا یا «راگبی» بیشتر، پاسخگوی نیازهای رقابتی جهان امروز است؛ در راگبی تیم سعی می‌کند با دست‌به‌دست کردن توپ به جلو و عقب، فاصله را به‌صورت یک واحد بپیماید. درحالی که شاید رویکرد متوالی سنتی یا «مسابقه امدادی» برای توسعه‌ی محصول... با اهدافی مثل سرعت و انعطاف‌پذیری بهینه تعارض داشته باشد.»

بعد از گذشت دو دهه این بحث همچنان ادامه دارد و البته هم‌زمان با رشد نیاز سازمان‌ها این دو رویکرد به رویکردهای ترکیبی تکامل یافته‌اند. سؤال این است: آیا بازی در یک زمین‌بازی یکسان، می‌تواند یک مسابقه راگبی را با یک مسابقه‌ی دوی امدادی انطباق دهد؟ نه تنها ممکن است، بلکه در بازار فوق‌رقابتی فناوری امروز که سرعت و انعطاف‌پذیری بالاترین جایگاه را دارند، یک ضرورت است. در این مقاله، سه راه برای مدیران پروژه‌ی آبشاری (دوندگان دوی امدادی) ارائه می‌شود تا بتوانند با تیم‌های چابکِ همکارِ خود بهتر کار کنند. همانطور که در بخش اول ذکر شد، استفاده از هر دو روش، در صورت به‌کارگیری درست هر کدام، شفافیت و بهره‌وری را در سازمان افزایش می‌دهد. ترفند این کار، دانستن روش مناسب برای نوع کار مناسب، و پذیرا بودن نسبت به رویکردهای ترکیبی است؛ تحمل کردن کافی نیست، باید پذیرا باشیم. برخی از سازمان‌ها چابک را در سازمان فناوری اطلاعات خود تقویت می‌کنند اما هنوز نیاز به راهی دارند که کار را مثل یک وسیله برقی به جعبه تقسیم سازمانی وسیع‌تری وصل کنند. دیگر سازمان‌ها از روش آبشاری برای پروژه‌های زیرساختی یا کاری که با اقلام تحویلی فیزیکی همراه است استفاده می‌کنند؛ اما فرایندهای چابک را برای نرم‌افزار، توسعه‌ی وب یا برنامه‌های کاربردی تلفن همراه که به تکرارهای مکرر و ابتکار نیاز دارند به کار می‌برند. و اما راه دوم.

۲

جایگاه همکاری خود را با تیم چابک تثبیت کنید.

به دلیل اینکه این دو روش اساساً متضاد یکدیگرند، ممکن است تمایل به برقراری سیلوها و ماندن در آن‌ها باشد. یکی از بهترین روش‌های جلوگیری از حالت تدافعی، اتخاذ رویکردهای پیشگیرانه در ارتباطات با تیم چابک است. باید خود را در دنیای آن‌ها دخیل کنید، آن‌ها هم باید همین‌طور در دنیای شما مشارکت کنند.

کل فلسفه‌ی چابک بر مبنای اعتماد است و به همین دلیل خطوط ارتباطی، بسیار باز و غالباً رودررو هستند. فقط از روی دیوار یک ایمیل به آن‌طرف پرت نکنید، به امید اینکه به دستشان برسد! شاید ایمیل شما آن‌قدر که برای شما مهم است برای آن‌ها مهم نباشد، مگر اینکه آن‌ها خود پیگیر باشند یا در اسپرینت فعلی‌شان تعریف شده باشد.

از تیم چابک اجازه بگیرید در یکی از جلسات برنامه‌ریزیِ اسپرینت و یا مرور اسپرینت حضور داشته باشید؛ از نمودار کار باقیمانده چشم برندارید. به این ترتیب نه‌تنها روند کار و زبان آن‌ها را یاد می‌گیرید و زبان آن‌ها را می‌آموزید، بلکه که در جریان موضوعاتی که برای شما مهم است قرار می‌گیرید. اگر بدانید که پروژه‌های شما بخشی از اسپرینت جاری تیم چابک است، می‌توانید بیشتر پاسخگو و یا در لحظه دردسترس باشید. اگر فهمیدید وظیفه شما بخشی از اسپرینت هفته آینده است، می‌توانید سؤال‌ها و پیشنهادهای خود را تا جلسه‌ی بعد نگه دارید تا یک مخاطب سروپاگوش داشته باشید.

هرچند که اصطلاحات و فرایندهای چابک بیگانه به نظر می‌رسند، تیم‌های چابک مسئول تحویل تعهدات خود هستند، درست مثل شما. به آن‌ها فرصت جلب اعتماد بدهید و شما هم اعتماد آن‌ها را جلب خواهید کرد.

۳

معیارهای سنجش چابک را به زبان آبشاری ترجمه کنید.

تیم چابک یک اسپرینت دوهفته‌ای در پایان این جمعه با ۱۱۰ نقطه‌ی داستان (Story Point) کامل دارد. شما مسئولید به رئیس بگویید با چه ویژگی‌هایی باید شروع کرد و چه‌موقع؟ مثل این می‌ماند که باید ترجمه کنید.

هرچند که تیم‌های چابک باید بتوانند مانند تیم‌های چابک کار کنند، باید به بقیه شرکت با اصطلاحاتی گزارش بدهند که رهبران سنتی پروژه مایل به شنیدن آن هستند. متریک‌ها باید برای تیم آبشاری تشریح شوند تا کارکرد تیم چابک را بتوان به‌موازات تیم‌های دیگر مقایسه کرد. به‌طور مثال تخمین‌های نقاط داستان را می‌توان معادل ساعت‌های برنامه‌ریزی‌شده در گزارش مدیریت سنتی پروژه تعبیر کرد. تکرارها یا تاریخ‌های اسپرینت را می‌توان محدودیت‌های وظیفه در نظر گرفت. وقتی مجموعه‌ای از معیارهای سنجش، تحت گزارش واحدی در جای خود قرار دارند، ارتباطات مناسب و تلفیق این دو فلسفه‌ی رقیب آسان‌تر می‌شود. برای اجتناب از ساعت‌ها کارِ دستی، ابزار مدیریتی خودکاری را بیابید که داده‌های تیم چابک را به گزارش‌های آبشاری تبدیل می‌کنند. با فناوری مناسب، می‌توان داده‌های چابک تیم را به داشبوردهای آبشاری تبدیل کرد و پیشروی تمام تیم‌ها، کل کار و روش‌ها را گزارش داد.

هم‌سطح کردن زمین بازی

شاید چابک و آبشاری تقریباً از همه‌ی جهات، متضاد یکدیگر باشند (یکی از بالا به پایین و دیگری از پایین به بالا عمل می‌کند). اما حرفی نیست که به هر دو نیاز است؛ خصوصاً برای تیم‌های توسعه یا مدیران پروژه‌ای که هنوز مسئولیت گزارش زنجیره را به مدیریت دارند. توسعه چابک همچنان روش منتخب بسیاری از تیم‌های IT خواهد بود و گزارش‌های آبشاری نیز جایگاه خود را از دست نمی‌دهند. هم‌زمان که در دور مسابقه امدادی خود می‌دوید، نگاهی هم بر اسکرام در مرکز زمین داشته باشید. اگر پذیرای قابلیت‌های چابک هستید، اگر در مورد کج‌فهمی در ارتباطات، پیشگیرانه عمل می‌کنید و معیارهای مناسبی را برای سازمانتان ترجمه می‌کنید، همه می‌توانند از میدان، برنده بیرون بیایند.

آیا تیم شما وقت همه را تلف می‌کند؟

وقتی که مدیریت عدم اطمینان و ریسک در یک پروژه ضعیف باشد، چه اتفاقی می‌افتد؟ معمولاً نتیجه آن، نرسیدن به مهلت‌های تحویل، اتلاف زمان و ناخوشنودی ذی‌نفعان است.

فرض کنید که مسئول پروژه‌ای هستید که اسناد نیازمندی‌های پروژه نداشته باشد؛ توسعه‌دهندگان نیز بدون تصویر مشخصی از آنچه باید انجام شود، کدها را بنویسند؛ وقتی نتایج ارائه شوند ذی‌نفعان شکایت می‌کنند و توسعه‌دهندگان را به دوباره‌نویسی کد وامی‌دارند و وقت همه تلف می‌شود. چه بسا بعداً معلوم شود که سخت‌افزار کار نمی‌کند و لازم شود با یک معماری جدید کار را از اول شروع کرد. نتیجه تلف شدن زمان بیشتری است؛ تکمیل پروژه دو برابر بیشتر زمان می‌گیرد.

مدیریت و کنترل ضعیفِ عدم اطمینان و ریسک در پروژه، به چنین تجربیاتی می‌انجامد.

در پروژه‌هایی مانند توسعه‌ی محصول سخت‌افزاری که عدم اطمینان و پیچیدگی بسیاری وجود دارد، اینکه کدام وظیفه از بیشترین اهمیت برخوردار است بسیار ضرورت دارد. در غیر این صورت بیشتر تلاش تیم هدر می‌رود. موفقیت در مدیریت این پروژه‌ها مستلزم ایجاد توازن ظریفی مابین حذف عدم اطمینان، کم کردن ریسک و پیش‌روی در مسیر بحرانی است. می‌توان این رویکرد را «مدیریت تطبیقی پروژه» نامید که هم از مدیریت پروژه استاندارد آبشاری (Waterfall) مؤلفه‌هایی دارد (که در آن کنترل پیچیدگی مناسب و کنترل عدم اطمینان ضعیف است) و هم مدیریت پروژه چابک (Agile) (که کنترل عدم اطمینان در آن مناسب و کنترل پیچیدگی در آن ضعیف است).

۱

حذف عدم اطمینان

در ابتـدای یـک پـروژه‌ی تطبیقی، باید تمرکـز بر حذف عدم اطمینان باشـد. چطور می‌تـوان بدون فهمیدن نیازهای کاربر انتهایـی، کـدی نوشـت یا بـدون اطلاع از بهتریـن فناوری موردنیاز پروژه، نمونه‌های اولیه ساخت؟ در این مرحله، باید بر مشارکت با ذی‌نفعان یا کاربران انتهایی، بررسی ظرفیت‌های فناوری‌هـای موردنظـر، و درک منابع و محدودیت‌های بودجه متمرکز بود. این تلاش در کل پروژه ادامه می‌یابد اما وقتی که پروژه آن‌قدر فهمیده شـد که بتوان یک سـاختار شکسـت کار (WBS) ایجاد کرد، می‌توان از این تلاش کم کرد. در WBS بـرای حـذف عدم اطمینان باقیمانـده، وظایفی در نظر گرفته می‌شوند. طبیعت پروژه‌های تطبیقی به‌گونه‌ای است که مانند پروژه‌های آبشاری، به‌جای به صفر رساندن کل عدم اطمینان از ابتـدای کار بـا چنـد مورد از عدم اطمینان، پیشـروی را ادامه می‌دهیم؛ همین که عدم اطمینان کم می‌شـود، شباهت برنامه به آبشاری بیشتر می‌شود.

۲

کاهش ریسک

ریسـک‌ها در آینده معلوم می‌شـوند، مشـکلات همین الآن اتفاق می‌افتند؛ مشـکل این اسـت که شاید ریسک‌ها رشـد کنند و به مشـکلات بدل شـوند. حل مشـکلات زمان می‌برد که نتیجه آن تأخیرها و افزایش هزینه‌ها است. هرچه زودتر بتوان ریسک‌ها را به مشکلات بدل کرد یا از آن‌ها خلاص شد، بهتر است.

وقتی که پروژه به‌خوبی فهمیده شـد، لازم اسـت یک ثبت‌نامه‌ی خطر یا «ریسـک رجیسـتر» [۱] ایجاد شـود. به‌این ترتیـب فهرسـتی از آنچـه امـکان دارد خـوب پیش نرود، میزان امکان وقوع آن، میزان شـدت آن در صورت وقـوع و چگونگـی کاهـش آن در صـورت وقـوع ایجاد می‌شـود. می‌توان از ۱ تا ۵ رتبه‌بنـدی کرد. برای «احتمال وقـوع»، مثـلاً می‌تـوان ۱ را معـادل بـا کمتر از ۲۰ درصد و ۵ را معادل بیش از ۸۰ درصد قرار داد. برای «شـدت»، ۱ معادل تأثیری بر پروژه اسـت که کمتر، تاریخ‌های مایلسـتون (نقاط رویدادها) و محدودیت‌های بودجه را در خطر قرار می‌دهد. ۵ یعنی اینکه شـاید نیاز به لغو کل پروژه باشـد، احتمالاً به این دلیل که ویژگی بخصوصی به‌دسـت نخواهد آمد یا بهای کالای فروخته‌شـده بسـیار بیش از انتظار اسـت. به‌علاوه می‌توان یک سـتون برای «اهمیت» اضافه کرد که حاصلِ ضرب احتمال وقوع و شـدت اسـت و ریسـک‌ها را به ترتیب اهمیت رتبه‌بندی کرد (جدول ۱ مشـاهده شـود). بخشـی از تلاش‌ها باید همیشـه به کاهش ریسـک‌های مهم اختصاص بیابد، چه با آزمون وقوع مشـکلات و چه با اعمال روش‌هایی برای کاهش. بخشـی از سـاختار شکسـت کار همین است. همچنین برای کم کردن اتفاقات غیرمنتظره در مسـیر، می‌توان برای ریسـک‌های احتمال‌بالا در برنامه روش‌های کاهشـی در نظر گرفت.

ریسک	احتمال وقوع	شدت	اهمیت	کاهش
دستگاه در تست انطباق‌پذیری الکترونیکی خطا داشته باشد	۴	۳	۱۲	ایجاد قفس فارادی در اطراف تجهیزات الکترونیکی
خنک‌سازی پسیو کافی نباشد	۳	۲	۶	اضافه کردن یک فَن

جدول ۱ این یک ثبت‌نامه‌ی خطر (ریسـک رجیسـتر) سـاده است. نسخه‌های پیچیده‌تر دارای مواردی مثل «قابلیت کشف‌شـدن» (چقدر احتمال دارد که ریسـک اتفاق بیفتد اما امکان آشکارسـازی خطا نباشـد) و «امکان وقوع غیرمنتظره» (اگر با کاهش مشکل حل نشد چه باید انجام شود) است. تجربه نشان می‌دهد که یک ریسک رجیستر ساده، کافی است و به‌روز نگه داشتن آن که موضوع مهمی است، آسان‌تر است.

۳

پیش‌روی در مسیر بحرانی

بـرای اطمینـان از تحویـل به‌موقـع اقلام تحویلی و برآوردن نیازهـای ذی‌نفعان می‌توان ابزاری را به‌کار برد. در غیر این صورت لازم است اهمیت این موارد تحویلی با ذی‌نفعان بررسی شود؛ بررسی می‌شـود کـه آیـا امـکان کم کردن اقلام تحویلی وجود دارد؟ راه دیگـر انتقال منابع مؤثر بر کاهش ریسک به وظایف مسیر بحرانی است. در صورت انتخاب این مسیر، لازم است به ذی‌نفعان توضیح دهید که مسـئله، احتمال وقوع مشـکلات بعدی را در پروژه بالا می‌برد و آن‌موقع، برطرف کردن مشکل دشوارتر است.

یک مدیر پروژه خوب، بندباز است

شـغل مدیریت پروژه ایجاد توازن مابین سـه بخش مذکور اسـت. اضافه‌کاری بر مسـیر بحرانی، اهمیت بیشـتر و کاهش ریسـک و عدم اطمینان، اهمیت کمتری دارد؛ اما هیچ فرمولی نیسـت که نشان دهد چه باید کرد.

در آخر موفقیت به مدیر پروژه و تیم او بستگی دارد که باید بفهمند:

- آیا عدم اطمینان آنقدر کاهش یافته که مدیر بتواند تمرکزش را بر خطرات بگذارد؟
- آیا ریسک‌ها آنقدر کاهش یافته‌اند تا بتوان بر مسیر بحرانی تمرکز کرد؟
- و آیا در این لحظه هر کسی بر مهم‌ترین وظیفه‌اش کار می‌کند؟

اگر این سـؤال‌ها اشـتباه برآورد شـوند (مثلاً ریسـکی که احتمال وقوعش کم بود به یک اشتباه‌زننده در پروژه تبدیل شـود)، فقط کافی اسـت لبخند بزنیم و تیم را بر مهم‌ترین وظایف فعلی‌اش متمرکز کنیم.

آیا «مستندسازی Agile» یک ترکیب ضد و نقیض است؟
شناخت نقش مستندسازی در محیط‌های توسعه‌ی اَجایل

برای خیلی از تیم‌های توسعه‌ی چابک (Agile)، کلمه‌ی «مستندسازی» کلمه‌ی زننده‌ای محسوب می‌شود. به‌علاوه تیم‌های چابک بر مبنای این ایده کار می‌کنند که معمولاً عدم اطمینان وجود دارد و این عدم اطمینان آن‌قدر تغییر سریعی دارد که صرف زمان زیادی بر طراحی یا معماری در ابتدا، بسیار نادرست است یا جلوتر در آینده، نتایج نامرتبطی خواهد داشت. به‌جای مستندسازی، توسعه‌دهندگان چابک تکرارهای کوتاه و مکرر را انتخاب می‌کنند و برای طراحی و معماری، رویکرد فاکتورگیری‌های مجدد را دنبال می‌کنند.

در یک کلام می‌توان گفت که تیم‌های توسعه چابک با یک ایده شروع می‌کنند، آن را به ویژگی‌ها و کارکردها تقسیم می‌کنند، نیازهای سطح‌بالا را می‌نویسند و سپس شروع به کار می‌کنند. اهداف فقط تا این حد مستند می‌شوند که برای پیشروی در ساخت، تکمیل مجموعه‌ی نیازهای اولویت‌دار در هر تکرار، جمع‌آوری نیازها در یک مورد کاری و تولید مستندی که هر کسی در حین کار با نرم‌افزار بتواند از آن استفاده کند، کافی باشد. داشتن اسنادی که هر کسی بتواند از آن استفاده کند هدفی تحسین‌برانگیز است؛ اما گاهی یک سند که قرار است برای هر کسی مناسب باشد برای هیچ کسی مناسب نخواهد بود. اگر تیم‌های چابک بخواهند با محیط‌های سازمانی بزرگ‌تری منطبق و متناسب شوند، باید آنقدر چابک باشند که برای این سازمان‌ها، راهبردهای مستندسازی کامل‌تر و درعین‌حال منعطف‌تری را طراحی کنند.

خلأ مستندات نگهداری

در حال حاضر، در بسیاری از پروژه‌هایی که ازلحاظ تجاری حساس هستند، مانند پرتال‌ها یا برنامه‌های کاربردی تجاری، روش‌های چابک اتخاذ می‌شوند. در این روش‌ها معمولاً تیم‌های بیرونیِ توسعه‌ی چابک وجود دارند که برنامه‌های کاربردی را می‌سازند و این برنامه‌ها باید در ۱۲–۱۸ ماه چه به خود شرکت به کارگیرنده و چه به فروشنده و شخص ثالث دیگری برای نگهداری سپرده شود. هرچند که برای فرایند ساخت مستندات نیازها و داستان‌های ساده‌تر کارساز است، ممکن است برای نگهداری کافی نباشند. به چند دلیل این داستان‌ها و مستندات، از اطلاعات موردنیاز برای نگهداری برنامه‌های کاربردی بی‌بهره‌اند. معمولاً وقتی که یک تیم در میانه یک پروژه‌ی بزرگ وارد می‌شود، امور تغییر می‌کنند و این تغییرات ممکن است که در اسناد اعمال نشوند. معمولاً افراد نسبت به این تغییرات طوری برخورد می‌کنند که انگار نرم‌افزار تحویل داده شده و حال زمان عبور به تکرار بعدی رسیده است. بنابراین روند نگهداری در مستندهای قدیمی متوقف می‌شود. به‌علاوه اسناد چابک بدون بازبینی کلی که معمولاً برای تیم اهمیت دارد، ارزش چندانی ندارد.

خلأ مستندات بازرسی

در برخی از صنایع مانند خدمات مالی و بهداشت، قوانین و مقررات بیرونی‌ای وجود دارند که برنامه‌های کاربردی باید با آن‌ها انطباق پیدا کنند. به دلیل اینکه قابلیت پیگیری در پروژه‌های بزرگ ضرورت دارد، تیم‌های بزرگ چابک، معمولاً از ابزاری برای ایجاد قابلیت پیگیری در محصولات استفاده می‌کنند، بااین‌حال کافی نیست.

به‌طور مثال پیاده‌سازی پرتال یک شرکت بزرگ دارویی را در نظر بگیرید؛ یک پرتال داخلی است که داده‌هایی را از اطلاعات دارویی درج می‌کند؛ نه‌تنها دستیابی به داده‌ها بلکه ساخت خود پرتال دستیابی به داده‌ها در معرض بازرسی است. باید همزمان که مصنوعات را باهم مرتبط می‌کنید، بر توسعه‌ی برنامه کاربردی و وب‌سایت نیز بسیار متمرکز باشید؛ وقتی بازرسان پیدا می‌شوند، شدیداً به قابلیت پیگیری و مستندات نیاز است.

مانند مستندسازی نگهداری در مستندسازی بازرسی نیز، نمی‌توان اطلاعاتی را که بازرسان نیاز دارند در یک مستند همه‌منظوره یافت. افزودن قابلیت پیگیری به پروژه نیز کافی نیست. بازرسان به مستنداتی نیاز دارند که مطابقت داشته باشند.

مستندسازی چابک

وقتی تیم‌های چابک با دو معیار سنجش مجزا، یعنی تیم‌های نگهداری و نیاز به بازرسی بیرونی، روبه‌رو می‌شوند، چارچوب تحویل منظم چابک (Disciplined Agile Delivery) و روشی مانند استمرار مستندسازی (Document Continuously) مناسب هستند. می‌توان با رویکرد مشابهی که در مستندسازی تحویل کد به کار گرفته می‌شود، خلأ بین مستندات تولیدشده برای پروژه‌های چابک و نیازهای تیم‌های نگهداری و بازرسان را رفع کرد. به جای ایجاد مستنداتی که قرار بود هر کسی بتواند از آن استفاده کند اما هیچ کس نمی‌تواند، به فکر ایجاد امکان نگهداری و مستندات بازرسی به‌صورت اقلام تحویلیِ مجزائی باشید که در بخشی از تکرارها تولید می‌شوند.

می‌توانیم با این رویکردها، نیازمندی‌ها، داستان‌ها و مصنوعات را از یکدیگر تکفیک می‌کنیم و اقلام تحویلی دیگری را که بخشی از تکرارها هستند ارائه دهیم. برحسب نیاز به نگهداری، چارچوب‌هایی را به این منظور می‌سازیم. اقلام را از نیازمندی‌های چابک به‌دست می‌آوریم و خلأهای موجود را در نظر می‌گیریم. بخشی از کار هر تکرار، برای پر کردن این فاصله‌ها است.

برای مستندسازی بازرسی، تمام نیازها را در نظر می‌گیریم و آن‌ها را بخشی از داستان می‌کنیم. در واقع داستان، تا زمانی که ما به گزارش و معیارهای سنجش انطباقی می‌پردازیم کامل نمی‌شود. شاید با تکامل مقررات بازرسی و نیاز به مطابقت، نیاز به افزایش قابلیت پیگیری و استمرار آن تشخیص داده شود.

برنامه‌ای برای هم‌راستایی

نیازمندی‌ها، مصنوعات، مستندات نگهداری و اسناد بازرسی، همگی می‌توانند باهم وجود داشته باشند و هم‌راستا باشند، نه الزاماً در یک محل، اما به‌عنوان بخشی از تکرارهای مختلف. با اتخاذ رویکردهای «دقیقاً به‌موقع» و «مناسب برای هدف»، تولید مستندات آسان‌تر می‌شود؛ در تمام این رویکردها می‌توان از بخش‌هایی که قابلیت استفاده‌ی مجدد دارند برخوردار بود.

به علاوه، وقتی دستورالعمل‌های نگهداری یا رهنمون‌های مدیریتی را بررسی می‌کنید، متوجه می‌شوید که بعضاً شامل مؤلفه‌های مستندسازی طراحی هستند. بخشی از پیگیری‌های موردنیاز بازرسان، با مواردی که تیم‌های پروژه خواهان آن هستند بسیار شباهت دارند، مانند پیوندهای مابین نیازهای تجاری، موردهای کاری توسعه و حتی آزمون‌های پذیرش.

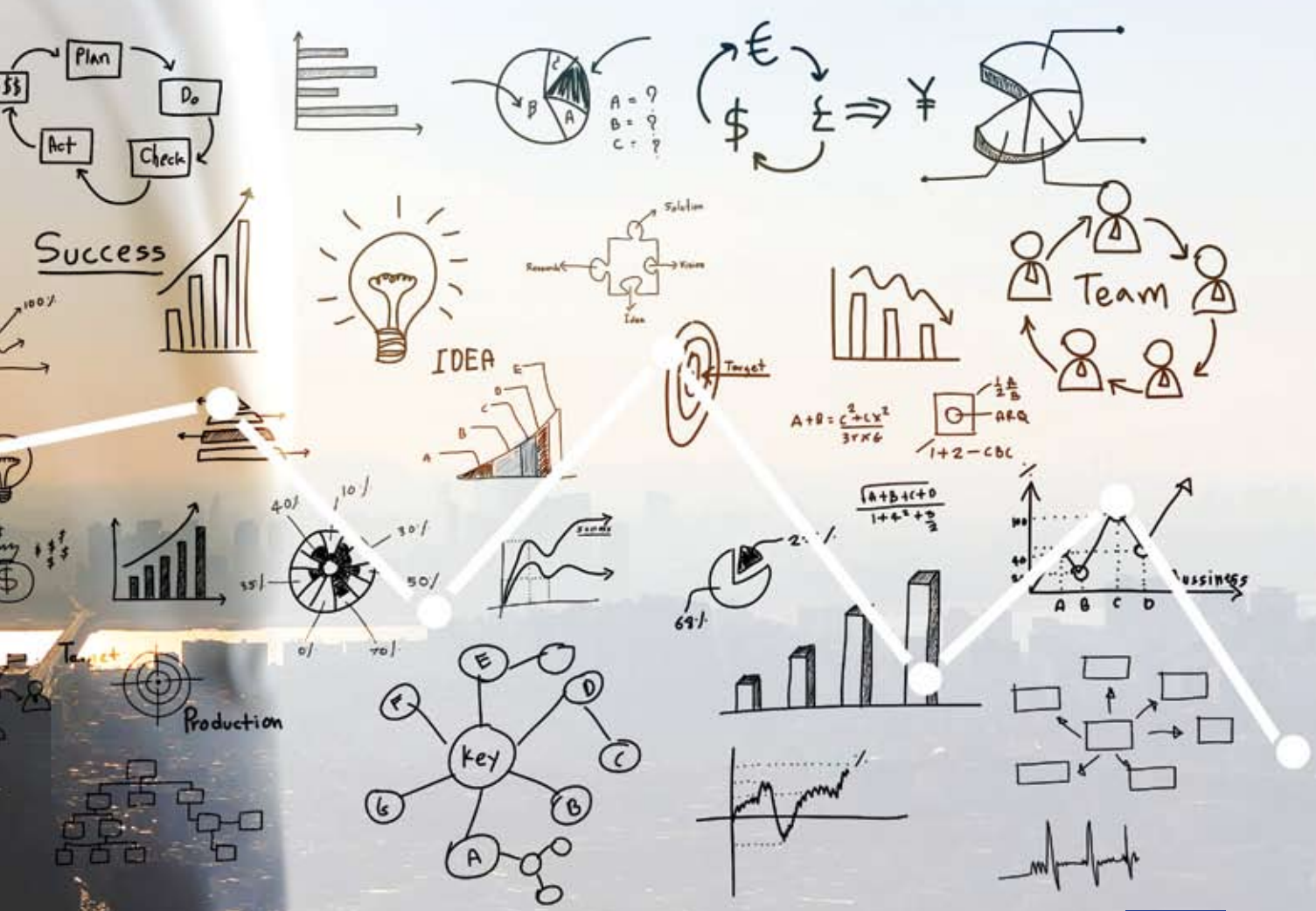
در برخی از پروژه‌های چابک، این تشابهات را تشخیص می‌دهیم، یک پوسته (shell) منتشر می‌کنیم و یک الگوی آماده (temple) با یک طرح کلی (outline) می‌سازیم که شامل خلاصه‌ای از جریان فرایند تجاری یا نمودار معماری است. برخی از اشیایی که در سایر ابزار می‌سازیم، می‌توانند در این دستورالعمل وارد شوند. نتیجه این رویکرد ایجاد اسناد کوچک (mini-document) است. از لحاظ نگهداری، این اسناد کوچک، حتی بیشتر از اسناد طراحی بزرگ و قدیمی‌تر محبوبیت پیدا کرده‌اند زیرا تیم‌های نگهداری می‌توانند مستقیماً به فضایی که عیب در آن تشخیص داده شده است تمرکز کنند. به‌این‌ترتیب مجبور نیستند که با اسناد بزرگی که مرور آن دشوار است، سروکار داشته باشند.

نتیجه‌گیری

مستندات برنامه‌های کاربردی‌ای که تیم‌های چابک برحسب نیازمندی‌ها، مصنوعات و داستان‌ها ایجاد می‌کنند، از لحاظ معیارهای سنجش تیم‌های نگهداری و بازرسی کافی نیستند. ارائه مستندات مناسب برای اهداف مختلف در صورتی ممکن است که یک رویکرد چابک برای سنجش اتخاذ کنید. همچنان که پیش می‌رود، متوجه نیاز به مستندسازی باشید و مستندات مختلفی را که در تکرارهای شما تولید می‌شوند اضافه کنید و راهکار مناسبی را برای مستندسازی به کار بگیرید که از نظر تمام ذی‌نفعان، از جمله نگهدارنده‌ها و بازرسان مناسب باشند.

چهار کلید برای درک چرخه حیاتی مدیریت پروژه

یک پروژه با تاریخ شروع و پایانش تعریف می‌شود اما میزان موفقیت یک پروژه را مدیریت درست چرخه حیاتی آن تعیین می‌کند. امروزه بیشتر اوقات مدیریت پروژه‌ها با نرم‌افزار مدیریت پروژه صورت می‌گیرد و ابزار و برنامه‌های کاربردی بسیاری هستند که به شما کمک می‌کنند وظایف را تا تکمیل به‌پیش ببرید، اما این ابزار واقعاً به شما در تمام فازهای پروژه کمک نمی‌کنند و صرفه‌نظر از تفاوت و کوچک و بزرگ بودنشان، همگی یک الگوی مشابه را دنبال می‌کنند. چرخه حیاتی اساساً شامل فازهای مختلفی از یک پروژه است. برای هر پروژه ۴ فاز وجود دارد: شروع، برنامه‌ریزی، اجرا، و بستن. البته هر کدام از این فازها را می‌توان به وظایف کوچک‌تری تقسیم کرد. به‌هرحال اگر قصد دارید پروژه شما پیشرفت کند، لازم است جزئیات هر فاز چرخه حیاتی یک پروژه را بدانید.



شروع

۱

آنچه در شروع یک پروژه اولویت دارد، داشتن یک مدیر پروژه برای اجرای همه امور است. احتمالاً کار شما همین است، اما اگر کار شما نیست، یک شخص اصلی برای هدایت همه امور پروژه تعیین کنید. حال بخش دشوارتر شروع می‌شود. لازم است که موافقت پروژه را اخذ کنید. برای این کار یک منشور پروژه و شرح کسب‌وکار می‌نویسید که در آن برای اثبات معقول بودن پروژه، بازگشت سرمایه را به تصویر می‌کشید و اهداف پروژه را تعریف می‌کنید. سپس محدوده پروژه خود را شناسایی می‌کنید و به همه موارد، بودجه‌ی موردنیاز را اختصاص می‌دهید. به‌علاوه در حین این اقدامات عملی، باید در مورد خطرات پروژه جلوتر فکر کنید و اینکه اگر این خطرات واقعاً رخ بدهند، چگونه آن‌ها را حل می‌کنید. در نهایت فهرستی از تمام اقلام تحویلی ارائه می‌دهید. اما این فاز چرخه حیاتی، فقط همین نیست. اکنون باید بر تیم تمرکز کنید. با تعیین یک حامی مالی پروژه و جمع کردن یک تیم شروع کنید. در این نقطه نیاز دارید که تمام ذی‌نفعان را بشناسید و یک نمودار سازمانی برای تعریف نقش و مسئولیت هر کس تعریف کنید که شامل موجودی‌ها، ایجاد قرارداد برای آن‌ها و توسعه‌ی یک برنامه‌ی ارتباطی و تعیین برنامه‌ی زمان‌بندی جلسات است. آخرین بخش از اولین قدم پروژه‌ی شما، انتخاب ابزار نرم‌افزاری مدیریت پروژه است. سپس برای تیم، و سیستم‌های اسناد آنلاین، حساب‌هایی را تعیین می‌کنید و برای ایجاد زمینه فاز بعدی، برنامه‌ریزی پروژه، لیست وظیفه و زمان‌بندی را می‌سازید.

برنامه‌ریزی

۲

برنامه پروژه، سندی است که پروژه حول آن مدیریت می‌شود. در اینجاست که فازهای اصلی برای تکمیل پروژه توصیف می‌شوند و از محدوده‌ی پروژه -که بخشی از منشور پروژه نیز هست- آغاز می‌شود. متعاقب آن، معرفی برنامه‌ریزی پروژه صورت می‌گیرد. لازم است که پروژه را به نقاط رویداد (مایلستون‌ها) که اساساً هر رویداد اصلی در پروژه هستند تقسیم کنید؛ از زمان جمع کردن تیم برای انجام وظایف مربوط به فاز برنامه پروژه شروع می‌شود. به‌علاوه، فازهای چرخه حیاتی خود را در این فاز تعیین می‌کنید، چرا که لازم است همه چیز در برنامه‌ریزی محسوب شود. این فعالیت‌ها را می‌توان به مجموعه‌ای از وظایف که نیاز به تکمیل آن‌ها است تقسیم کرد. طبیعتاً مرحله بعدی، تعریف این وظایف و تلاش لازم برای انجام آن‌ها است.

برای تکمیل هر وظیفه‌ای که تشخیص داده‌اید، باید منبعی را تخصیص دهید و درنهایت، لازم است که زمان‌بندی هر فاز و فعالیت‌هایی را که پیش از این تعریف کردید خلاصه کنید. بدین ترتیب، تقریباً وارد کار شده‌اید که اجرای پروژه نامیده می‌شود.

اجرا

۳

حال باید اقلام تحویلی را که قبلاً شناسایی کردید به‌دست بیاورید. در ایفای این وظایف، لازم است که بر خروجی پروژه، کنترل و نظارت بیشتری داشته باشید؛ نیاز به اعمال مدیریت کیفیت برای اطمینان از پیشروی در چارچوب استانداردِ تعیین‌شده است.

به‌علاوه لازم است به نظارت بر تیم ادامه بدهید و مطمئن شوید که از زمان‌بندی عقب نیفتاده‌اید یا به دلیل وابستگی وظایفِ اعضای تیم انسداد ایجاد نمی‌شود. مدیریت زمان به معنای چگونگی بهره‌ور نگه‌داشتن تیم است. یک تیم ممکن است به شیوه‌های بسیاری، زمان خود را تلف کند. نظارت و واکنش سریع شما ضرورت دارد.

البته شما نه تنها تیم بلکه تمام ارکان پروژه را مدیریت می‌کنید. در اینجا لازم است برای اطمینان از تمام نشدن پول و اینکه هیچ چیز شما را در پروژه غافلگیر نکند، هم مدیریت زمان و هم مدیریت ریسک را به کار بگیرید. این مشکلات با ریسک‌ها تفاوت دارند و تفاوت آن‌ها، نیاز آن‌ها به توجه فوری است.

در طی این فرایندها شما مسئول گزارش به حامیان مالی و ذی‌نفعان و نیز اعضای تیم هستید. در این فاز از چرخه‌ی حیاتی پروژه است که برنامه ارتباطی شما بیشترین اهمیت را دارد.

بستن

۴

کار پُرمشغله‌ای است. وقتی که یک پروژه را کامل می‌کنید، نمی‌توانید بدون مواجهه با تمام نکات نیمه‌تمام به پروژه بعدی بروید. به‌این ترتیب باید تمام قراردادها را تکمیل کنید و مطمئن شوید که به تیم و فروشندگان، پرداخت شده و امضا و فایل‌بندی کاغذها به پایان رسیده است.

کار دیگری که باید انجام شود گذشته‌نگری با تیم است. از تیم بازخورد بگیرید. بررسی کنید که چه چیز جواب داده و چه چیز جواب نداده است، این گونه می‌توانید موفقیت‌ها را تکرار کنید و از گودال‌ها در مسیر پروژه‌ی بعدی در امان بمانید. بررسی یک پروژه راه فوق‌العاده‌ای برای تبدیل تجربیات به عملکردهایی است که به شما کمک می‌کند بیاموزید و رشد کنید. سعی کنید به تیمتان کمک کنید، چه بسا افرادی باشند که دوباره با آن‌ها کار می‌کنید. جشن گرفتن را فراموش نکنید. به‌سختی کار کردید، همین‌طور تیم شما. شایسته زمانی برای خوشحال بودن برای موفقیت پروژه هستید؛ این مهم است. این فقط یک جشن نیست بلکه راهی برای ایجاد وفاداری تیمی و اعتماد به توانایی خود و پرداخت سهم تمام افرادی است که دوباره با آن‌ها کار می‌کنید.

محمدرضا اسماعیل طهرانی

فرهنگ‌سازی، عامل ماندگاری

یکی از چالش‌هایی که سازمان‌ها در استقرار دفتر مدیریت پروژه با آن روبه‌رو می‌شوند، ناآگاهی کارکنان سازمان با شرح وظایف این واحد است. اما وظایف این واحد چیست؟

- ایجاد یک بانک اطلاعاتی از کل پروژه‌های شرکت؛
- ایجاد یک بانک اطلاعاتی کامل از ارتباطات پروژه به تفکیک ذی‌نفعان و پروژه‌ها؛
- شفاف‌سازی روند پیشبرد پروژه‌ها؛
- بسترسازی جهت انتقال تجربیات پروژه‌ها به اعضای تیم‌ها؛
- تهیهٔ گزارش‌های سنجش عملکرد تیم‌ها؛
- نظارت و اعلام وضعیت بودجهٔ مصرفی و باقیمانده به تفکیک پروژه‌ها؛
- بهبود و تسهیل ارتباطات میان اعضای تیم و ذی‌نفعان؛
- کاهش خطای ناشی از انتخاب مسیر نادرست در پیشبرد پروژه‌ها؛
- تعیین روش‌شناسی پیشبرد پروژه‌ها و آموزش به تمام اعضای تیم‌ها؛
- مدیریت ریسک در سطح کلان سازمان؛
- جایگاه راهبری و رایزنی مابین مدیران تیم‌ها.

رسیدن به یک واحد PMO با وظایف فوق پروژه به پروژه دستخوش تغییر خواهد شد؛ کمی دشوار است و نیاز به همراهی فرهنگ قوی سازمانی دارد. این فرهنگ قطعاً و صرفاً توسط مدیر کل مجموعه قابل اجرا و استقرار کامل خواهد بود و انتظار اینکه به‌صورت ناگهانی همه موارد فوق به‌صورت کامل و صددرصدی برقرار شود، کاملاً اشتباه است. توجه شود که موارد بیان‌شده تنها بخشی از مهمترین وظایف دفتر مدیریت پروژه بود و باید تمرکز اصلی عملکرد این واحد، اتمام پروژه‌ها با کیفیت عالی و به‌نحوی باشد که منافع ذی‌نفعان و تیم پروژه حفظ شود. برای رسیدن به این موضوع باید انعطاف‌پذیری را بیشتر کنیم.

پرینس۲ یا چابک؟
PRINCE2 or Agile?

از زمان عرضهٔ مانیفست توسعهٔ نرم‌افزار چابک در سال ۲۰۰۱ جنگ بین پرینس۲ و چابک به‌تدریج شدت گرفته است. هر دو روش طرفداران و پشتیبانانی دارند.
می‌گویند: «چابکی بسیاری در پرینس۲ هست» یا «پرینس۲ درحال مرگ است»؛ «اصلاً چرا طرف یکی را بگیریم؟»؛ «پرینس۲ آبشاری (Waterfall) است، پس چابک نیست.»

مانیفست توسعهٔ چابک نرم‌افزار

ما درعینِ اجرای توسعهٔ نرم‌افزار و کمک به دیگران، راه‌های جدیدی را برای توسعهٔ نرم‌افزار کشف می‌کنیم. درحینِ این کار به چنین ارزشی رسیده‌ایم:

- افراد و تعامل‌ها برترند از فرایندها و ابزار
- نرم‌افزار درحال کار برتر است از مستندهای کلیتگرا
- همکاری با مشتری برتر است از مذاکره دربارهٔ قرارداد
- پاسخگویی به تغییر برتر است از تبعیت از یک برنامه

هرچند موارد طرف راست ارزش دارند، برای موارد طرف، بیشتر ارزش قائلیم.
اخیراً این جنگ از پشت‌پرده‌های مدیریت پروژه به محیط‌های عمومی‌تر کشانده شده؛ این اظهار نظرات و مانند آن در کانال‌های رسانه‌های اجتماعیِ مربوط به کسب‌وکار مطرح می‌شوند.
با تلاش شرکت‌ها برای عقب نیفتادن و رقابت در جنگلِ بی‌رحمِ کسب‌وکار، روش‌های جدید کاری پیشَرفت کرده‌اند و روش‌های نوینی برای مدیریت پروژه به‌وجود آمده‌اند.
تردیدی نیست که شرکت‌ها برای استمرار حیات خود باید در پاسخ به نیازهای مشتری چابک‌تر باشند؛ اما آیا در پشت پردهٔ مدیریت پروژه، پایانِ حیاتِ پرینس۲ صحت دارد؟ آیا مرگی در کار است و از این پس، پروژه‌ها و مدیریت تغییرات تنها باید طبق رویکرد چابک انجام بگیرند؟
در سواحل آب‌های شمالی، جمعیت هلندی Agile کارگاهی را برای بررسی بیشتر این موضوع در هلند برگزار کردند. بیش از ۶۰ چابک کار حاضر در این کارگاه، پاسخ برخی از این سؤال‌ها را دریافتند؛ تاریخ این روش‌ها را بیشتر بررسی کردند؛ و نمونه‌های مناسبی را از هردو روش کندوکاو کردند. این گروه برای این سؤال پاسخ بالقوه‌ای یافتند: آیا پروژه‌ها را همچنان باید با پرینس۲ انجام داد، یا خیر؟

اما پاسخ چه بود؟

در کمال تعجب، در گروهی که بیشتر آن متشکل از توسعه‌دهندگان نرم‌افزار، متخصصان DevOp، مالکان محصول و مانند آن‌ها بود، فقط ۲۰ درصد از گروه باور داشتند که چابک واقعاً بهتر از پرینس۲ است، و ۸۰ درصد به دلایل مختلفی بی‌طرف بودند. جای تعجب ندارد که کسی معتقد نبود که پرینس۲ بهتر از چابک باشد...

کدام‌یک بهتر است: پرینس۲ یا چابک؟

آیا حیات پرینس۲ به‌پایان رسیده یا می‌رسد؟

جالب‌ترین نتیجهٔ جلسه، پاسخ به این سؤال بود که آیا حیاتِ پرینس۲ به‌پایان رسیده یا درحالِ به‌پایان‌رسیدن است؟ نتیجه دقیقاً نصف‌نصف بود!
بخش آخر این جلسه به این سؤال اختصاص یافت: «چرا بازهم به کار با پرینس۲ ادامه بدهیم؟»
پاسخ‌ها به این سؤال به‌صورت مزایا و معایب بر دیواری نوشته شد. بدیهی بود که هرچند چابک همچنان عنوانِ «بهترین روش برای توسعهٔ نرم‌افزار» را حفظ خواهد کرد، هنوز، درمجموع، استفاده از پرینس۲ مزایایی دارد، به‌خصوص زمانی که برای پروژه و کسب‌وکار شما مناسب باشد.

9 ابزار ضروری برای تیم‌های چابک (Agile) توسعهٔ محصول

برخی از کارآفرینان با خلق فناوری‌هایی توانسته‌اند بازی رقابت را به نفع خود تغییر دهند؛ اما ابزار این کارآفرینان چیست؟

بی‌شک، یک تیم پربازده و مؤثر توسعه. بهترین ویژگی کسب‌وکارهای نوپا (استارتاپ)، برای رقابت با غول‌های کسب‌وکار، شتاب و توانمندی آن‌ها در به بازار رساندن ایده‌هاست. از طرفی، کسب‌وکارهای نوپا معمولاً با بودجه‌های کمتر و منابع ناچیز کار می‌کنند؛ ازاین‌رو این عملکرد تیم توسعه است که تعیین‌کنندهٔ شکست و پیروزی آن‌هاست. درهرصورت، همیشه «تحویل» مرکز تلاش‌های آن‌هاست. برای کسب بیشترین بازدهی از تیم پروژه نیاز است که بر ابزار درستی سرمایه‌گذاری شود. این ابزار پلتفرم‌های دیجیتالی هستند که تیم توسعهٔ محصول از آن‌ها برای ابتکار و خلاقیت و عرضهٔ محصولات استفاده می‌کنند.

توسعهٔ محصول

ارتباطات و مدیریت پروژه‌ها امور دشواری هستند اما کار را باید، دیر یا زود، انجام داد. درادامه چندی از ابزار موردنیاز تیم توسعه، برای انجام کارهای روزانه، معرفی می‌شوند.

Sublime Text Editor

ازآن‌جهت که توسعه‌دهندگان بیشتر وقت خود را صرف کد نوشتن می‌کنند، یک ویرایشگر متنی، که عملکرد مناسب و قابلیت سفارشی‌سازی دارد، فناوری باارزشی است. به‌موجب ابزار Guard در Sublime Text، تیم می‌تواند با افزایه‌هایی که عملکرد و سرعت کدنویسی را بهتر می‌کند کدهای زیبا و کاربردی بنویسد.

New Relic

نرم‌افزار شما دنیا را تغییر نمی‌دهد، مگر اینکه کاربران شما تجربهٔ خوبی داشته باشند. New Relic برای تیم‌ها امکان پیگیری اجرای بلادرنگ نرم‌افزارشان را فراهم می‌کند. به‌این‌ترتیب می‌توانند مشکلات را رفع کنند و راه‌هایی برای پیشرفت روزانهٔ نرم‌افزار بیابند.

GitHub

با GitHub تیم توسعه می‌تواند به‌آسانی کد را در یک ابر (cloud) ذخیره کند. این ابزار برای بررسی‌ها و بحث‌های عمیق دربارهٔ کد نیز به‌کار برده می‌شود.

Wercker

برای اجتناب از تگناها در فرایند توسعهٔ محصول، لازم است که زودهنگام و به‌دفعات محصول را آزمون کنید. با Wercker تیم توسعه می‌تواند آزمون‌های خودکار انجام بدهد و ویژگی‌های جدیدی را توسعه دهد.

Trello

با ترلو، که از روش کانبان در شرکت تویوتا گرفته شده، می‌توان روند پروژه‌ها را، از ایده تا عرضه، به‌طور تصویری پیگیری کرد. Trello کارکردهای بسیاری برای تیم‌های توسعه‌ای که سعی در تحویل سریع‌تر ویژگی‌ها دارند فراهم می‌کند. یکی از این ویژگی‌ها یک ریسهٔ ارتباطی (thread) مختص پروژه است. تیم توسعه‌ای که از هیچ نرم‌افزاری برای پروژه‌ها استفاده نکند، نمی‌تواند آن‌قدر که باید مؤثر باشد.

Google Docs

با Google Drive استارتاپ شما راه آسانی برای ساختن ویکی‌های داخلی می‌یابد. تیم توسعه یک روال عملیاتی استاندارد برای آزمون ویژگی‌ها قبل از عرضهٔ آن‌ها به کاربران دارد. با این روال مجبور نیستند که بارها و بارها دربارهٔ روالِ آزمون سؤال‌های تکراری بپرسند؛ درنتیجه به‌روزرسانی‌ها سریع‌تر عرضه می‌شوند.

مدیریت گردش کار

آیا تیم شما پربازده است و کارها را به انجام می‌رساند؟ یک تیم مؤثر توسعه برای مدیریت پروژه‌ها یک روند تعریف می‌کند و با اطمینان می‌داند که ویژگی‌ها به‌موقع و طبق مشخصات تحویل داده می‌شوند.

ارتباطات

ارتباطات همیشه یک چالش است، خصوصاً وقتی که اعضای تیم توسعه از راه دور کار می‌کنند. مادام اینکه ابزار ارتباطی صحیح و بجایی دارید، تیم شما ذره‌ای از کار عقب نمی‌افتد.

Slack

رشد شگفت‌آورِ Slack بی‌دلیل نیست. استفاده از آن آسان است و به ارتباطات آسانِ تیم کمک می‌کند. صرف‌نظر از اینکه اعضای تیم کجا کار می‌کنند، می‌توانند به‌آسانی و به‌سرعت ایده‌ها را باهم درمیان بگذارند و اگر دربارهٔ باگ یا ویژگیِ بخصوصی به مشکلی برخوردند، درخواست کمک کنند.

Screen Hero

این صفحه‌نمایش در کارهای نرم‌افزاری به‌خوبی با Slack سهیم می‌شود (Slack اخیراً مالک آن شده است). با Screen Hello اعضای تیم می‌توانند صفحه‌نمایش خود را به‌طور بلادرنگ به‌اشتراک بگذارند و امکان چندین نشانگر بر صفحه وجود دارد تا اعضای تیم بتوانند برنامه‌ها را در دستگاه یکدیگر جفت کنند.

با تیم توسعهٔ خود دربارهٔ ابزار موردنیاز برای کار مؤثر و ارتباط بهترشان با یکدیگر صحبت کنید. یکی از بهترین سرمایه‌هایی که می‌توانید در کسب‌وکار خود از آن بهره بگیرید اطمینان از کاراییِ ابزار موردنیاز مهندسان است.

10 راهکاربرای مالکان محصول درچارچوب اسکرام

چارچوب اسکرام (Scrum Framework) چارچوب سبکی است که در آن سه نقش (Role)، سه مصنوع (Artifact) و پنج رویداد (Event) تعریف می‌شوند؛ این چارچوب در توسعه و نگهداری محصولات پیچیده در محیط‌های پیچیده کاربرد دارد. اسکرام کارهای الزامی زیادی را تجویز نمی‌کند، و روش‌ها و مستندات و جلسات گسترده‌ای- که تبعیت از آن اجباری است- ندارد. بااین‌حال در راهنمای اسکرام (Scrum Guide) چند قانون وجود دارد. سعی نکنید چارچوب اسکرام را تغییر بدهید؛ تا چارچوب اسکرام را به شیوهٔ ذکرشده در راهنمای اسکرام پیاده‌سازی نکرده‌اید، از هیچ مؤلفه‌ای چشم‌پوشی نکنید. در این مقاله به ۱۰ راهکار دربارهٔ چارچوب اسکرام می‌پردازیم.

۱. هدفِ اسپرینت (Sprint Goal) داشته باشید

مسئولیت اصلی شما، به‌عنوانِ مالک محصول، بیشینه کردن ارزش محصول (محصولاتی) است که به مشتریان و کاربران تحویل داده می‌شود. برای بیشینه کردن ارزش، چند تصمیم‌گیری دشوار، دربارهٔ این که چه بسازید و چه نسازید، پیشِ رو دارید. همچنین باید دربارهٔ زمان ساخت نیز تصمیم بگیرید. بک‌لاگ محصول را طوری تنظیم کنید که فرصت‌ها به حداکثر و ریسک‌ها به حداقل برسند و تیم اسکرام متمرکز بماند. تمرکز بر معیارهای اصلی اسکرام است و واقعاً ضرورت دارد. حفظ تمرکز تیم اسکرام به شما کمک می‌کند که بهره‌وری خروجی و ارزش را افزایش بدهید؛ به‌علاوه بهره‌وری تیم را نیز بالا می‌برد و آن را اثرگذارتر و خودسامان‌ده‌تر می‌سازد. مالک محصول با افزایش تمرکز تیم اسکرام می‌تواند بر موفقیت یک اسپرینت جداً اثرگذار باشد. یکی از راه‌های ایجاد تمرکز تعیین یک هدف اسپرینت (Sprint Goal) برای هدایت تیم است. با هدف اسپرینت، تیم مقصد و جهتش را درمی‌یابد و به‌وضوح می‌بیند که آیا اسپرنیت موفق خواهد بود یا خیر. بسیاری از مالکان محصول در کارهای روزانه‌شان هدف اسپرینت ندارند؛ که این موضوع اصلا خوب نیست.

۲. جزء تیم اسکرام باشید

تیم اسکرام شامل یک مالک محصول، یک اسکرام‌مستر، و یک تیم توسعه است. در بسیاری از تیم‌های اسکرام شاهد هستیم که مالک محصول مثل یک عضوِ واقعیِ تیمِ اسکرام عمل نمی‌کند. در اسپرنیت حضور ندارند، با تیم اسکرام مشارکت نمی‌کند، در پیشروی‌های پیوسته پشتیبان تیم نیست و مانند آن. به‌عنوانِ یک مالک محصول، باید عضوی از تیم اسکرام باشید؛ زیرا این تیم است که به شما در بهینه کردن ارزش محصول یاری می‌کند. اگر شما به تیم اسکرام کمک کنید آن‌ها نیز در عوض کمک خوبی برای شما خواهند بود!

۳. تمرکز تیم اسکرام را حفظ کنید

همان‌طور که در راهنمایی ۱ ذکر شد، متمرکز کردنِ تیم اسکرام ضرورت دارد. علاوه‌بر تعیین یک هدف اسپرینت برای تیم، با انتخاب اقلام مربوط‌به بک‌لاگ محصول، می‌توانید تمرکز بیشتری بر اسپرنیت ایجاد کنید. به‌طورمثال در تنظیم بک‌لاگ محصول، اقلامی از بک‌لاگ محصول را، که نیازمندِ کار بر بخش معینی از یک سیستم یا بر همان فرایندند، گروه‌بندی کنید.

۴. اسکرام روزانه را «مدیریت» نکنید

اسکرام روزانه یک رویداد مربوط‌به تیم توسعه است که اعضای تیم توسعه در آن به پیشرفت فعلی رسیدگی می‌کنند و برنامه‌شان را برای ۲۴ ساعت آینده انطباق می‌دهند. حضور مالک محصول و اسکرام‌مستر طی اسکرام روزانه، برای پاسخ دادن به سؤال‌های تیم، و راهنمایی آن‌ها، درصورت تمایل یا درخواستشان، مفید اما نه ضروری است. بااین‌حال درعمل مالکان محصول و اسکرام‌مسترهای زیادی را می‌بینیم که اسکرام روزانه را «مدیریت» می‌کنند؛ وظایف را میان اعضای تیم تقسیم می‌کنند؛ می‌پرسند که افراد در روز قبل چه کردند و قرار است چه کنند؛ بک‌لاگِ اسپرینت را به‌روزرسانی می‌کنند و غیره. از اکنون به بعد، دیگر چنین نکنید! اسکرام روزانه رویدادی مربوط‌به تیم توسعه است، نه یک جلسهٔ به‌روزرسانی وضعیت که تیم توسعه به اطلاع اسکرام‌مستر و مالک محصول می‌رساند! وقتی از مدیریت اسکرام روزانه دست بردارید، می‌بینید که تیمِ توسعه خودسامان‌دهی بیشتری می‌یابند؛ ازاین‌رو شما، به‌عنوانِ مالک محصول، زمان و فضای بیشتری برای تمرکز بر ذی‌نفعان و نقشه‌های درازمدت می‌یابید.

۵. زمانی را به بهسازی بک‌لاگ محصول (Product Backlog Refinement) تخصیص دهید و آن را تنظیم کنید

با بهسازی بک‌لاگ محصول می‌توانید بک‌لاگ محصول را مرتب کنید یا مجدداً تنظیم کنید. این کار به جداسازی اقلام بک‌لاگ محصول نیز کمک می‌کند و اقلامِ بک‌لاگ محصول را شفاف‌تر می‌سازد. پیشنهاد می‌شود که ۱۰ درصد از ظرفیت تیم‌های توسعه برای بهسازی صرف شود؛ اما این فقط یک پیشنهاد است. بنابراین باید بر میزان زمانی که صرف تنظیم می‌شود نظارت داشته باشید و آن را تنظیم کنید. به‌این‌ترتیب روند بهسازی، یک ریتم و وزن می‌یابد. مثلاً می‌توان یک جلسهٔ بهسازی با اسکرام‌مستر (و درصورت‌نیاز ذی‌نفعان)، دوبار در هفته، مثلاً در سه‌شنبه‌ها و پنج‌شنبه‌ها، به‌مدت یک یا دو ساعت، تشکیل داد. ریتم تشکیل جلسات بهسازی باعث کاهش پیچیدگی و افزایش توانمندیِ تیم اسکرام و ذی‌نفعان در پیش‌بینی امور می‌شود. علاوه‌بر ایجاد ریتم، بررسی کنید چقدر زمان برای بهسازی لازم است. برای برخی از تیم‌ها ۵ درصد و برای برخی ۱۰ درصد و برای برخی دیگر ۲۰ درصد؛ بنابراین بررسی کنید چقدر زمان لازم است. پیشنهاد می‌شود که برای این کار، هفتان را به بک‌لاگ محصولی با اقلام «آماده» برای دو یا سه اسپرینت بعدی قرار بدهید.

۶. در انتهای هر اسپرینت یک مورد «انجام‌شده» یا Done تحویل بدهید

تیم‌های اسکرام زیادی را مشاهده می‌کنیم که در یک اسپرینت، متحملِ کارهای بسیار زیادی می‌شوند؛ که غالباً نتیجه‌اش نرساندن به هیچ «انجام‌شده‌ای» (Done) است. مطمئناً تیم کارهای زیادی کرده است، بی‌شک کار بر اقلام بسیاری را آغاز کرده، اما در آخر اسپرینت تمام کار (یا بخشی از آن) به اسپرینت بعدی محول می‌شود. اگر متوجه این موضوع شده‌اید این راهنمایی را به‌خوبی می‌فهمید. لازم است تعداد اقلامی را که تیم در اسپرینت درنظر می‌گیرد کاهش دهید. زیرا می‌خواهید ارزش محصول را بیشینه کنید؛ هدفتان خروجی دادن تا حد ممکن نیست، بلکه نتیجه دادن تا حد ممکن است. بنابراین تیم ممکن است خروجی‌های زیادی تحویل بدهد؛ مثلاً آیا ۲۰ اقلام تحویل بک‌لاگ (PBI) که ۸۰ درصد آن «انجام‌شده» یا Done است فوق‌العاده به‌نظر می‌رسد؟ خیر! زیرا ۲۰ ارزش صفر دارید! اگر «انجام‌نشده» باشند، هیچ ارزشی به مشتری‌ها و کاربران تحویل داده نمی‌شود. بنابراین مرجح است که ۱۰ اقلام داشته باشید که ۱۰۰ درصد تکمیل شده باشند؛ زیرا ۱۰ قلم ارزش دارید نه صفر ارزش.

۷. به معنای «انجام‌شده» یا Definition of Done احترام بگذارید

تحت فشارهای خارجی و داخلی (مثل موعدهای تحویل، رویدادهای بازاریابی یا دیگر فرصت‌ها) بسیاری از مالکان محصول تیم پروژه را به تحویل سریع ترو امی دارند. اما محدودهٔ تحویل را کاهش نمی‌دهند؛ معمولاً تیم را به این وامی‌دارند که محدودهٔ یکسانی (یا بیشتری) را در زمان کمتری تحویل بدهند. این کار غالباً (نه همیشه) به افت کیفیت منجر می‌شود. وقتی تیم اسکرام از معنای «انجام‌شده» تبعیت نکند، کیفیت نامناسبی تحویل می‌دهد. در کوتاه‌مدت به‌نظر نمی‌رسد که بد باشد، اما در درازمدت منجربه کار انجام‌نشده و بدهی فنی می‌شود. در بسیاری از سازمان‌ها شاهد هستیم که از ۱۰ تیم حدود ۴ تا ۶ تیم فقط بر حل مسئله، برطرفی باگ، مدیریت مشکلات و پرداخت بدهی فنی کار می‌کنند. همیشه بهتر کردن کیفیت بعد از راه‌اندازی محصول از باکیفیت ساختن از همان ابتدا پرهزینه‌تر است. پس قائل به معنای انجام‌شده یا Definition of Done باشید و بگذارید تیم توسعه محصولات کیفی تحویل بدهند.

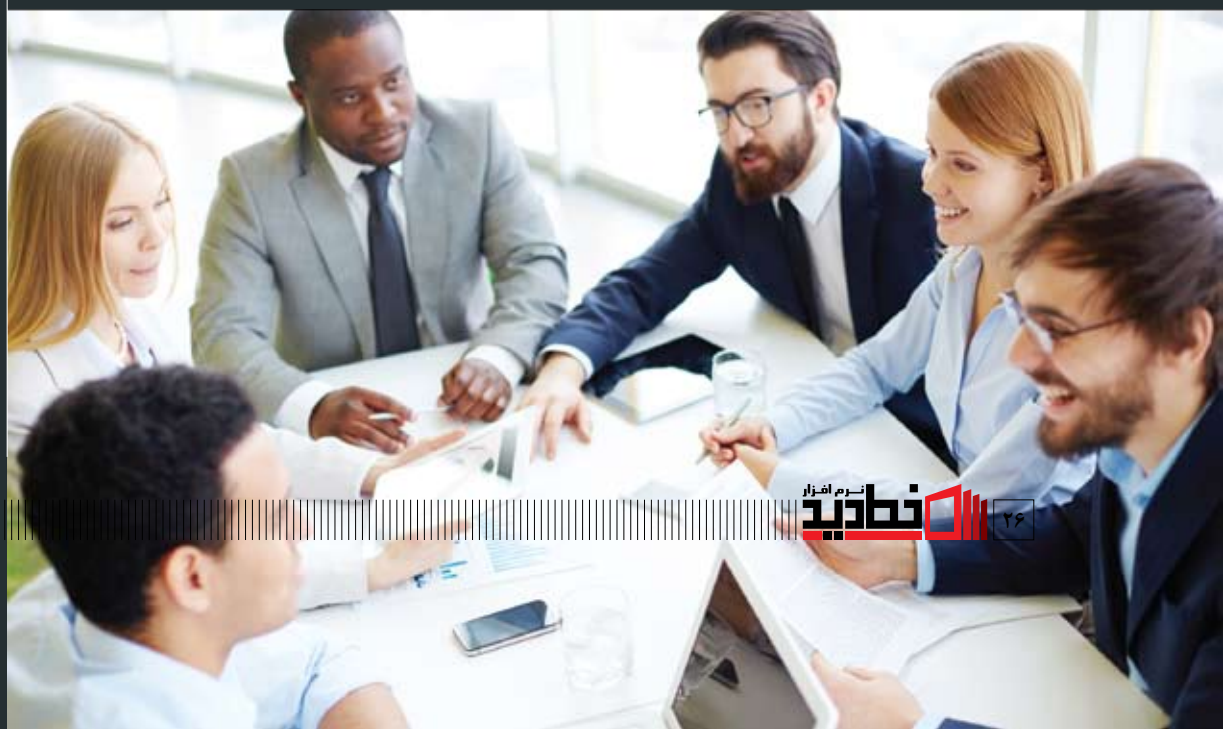
دین فنی یا Technical Debt مفهومی در توسعهٔ نرم‌افزار است که بیانگر هزینهٔ احتمالی دوباره‌کاری اضافی است که به‌دلیل انتخاب یک راهکار آسان‌تر درعوض راهکار بهتری که زمان بیشتری می‌گرفت ایجاد می‌شود. اگر دِین فنی پرداخته نشود، «بهره» پیدا می‌کند و بعداً پرداخت آن دشوارتر می‌شود. بی‌توجه ماندن به دِین فنی می‌تواند موجب «درگاشت (آنتروپی) نرم‌افزار» شود. دِین فنی الزاماً بد نیست بلکه گاهی در اثبات مفهوم (proof of concept) برای پیشبرد اولیهٔ پروژه لازم است. با این حال برخی از کارشناسان معتقدند که استعارهٔ «دین فنی» تأثیر آن را کم جلوه می‌دهد و نتیجه، جهت‌گیری ناکافی برای تصحیح آن است.

۸. بررسی اسپرینت (Sprint Review) را با دمو (demo) اشتباه نگیرید

در بسیاری از سازمان‌ها بررسی اسپرینت یا Sprint Review را یک دمو درنظر می‌گیرند. دمو دادنِ افزایش محصول (Product Increment) بخشی از بررسی اسپرینت است؛ اما تنها هدف آن نیست! هدف «بررسی اسپرینت» تعامل با ذی‌نفعان به‌منظور فیدبک گرفتن دربارهٔ افزایش محصول، بررسی بازار، بررسی نقشهٔ راه محصول، و نیز همکاری برای قدم‌های بعدیِ موردنیاز برای بیشینه کردن ارزش محصول است. معنایش این است که «بررسی اسپرینت» بخشی از تعاملات مربوط‌به کار است، نه یک جلسهٔ به‌روزرسانی وضعیت که تیم توسعه (با استفاده از پاورپوینت) به اطلاع ذی‌نفعان یا بدتر، به اطلاع مالک محصول می‌رسانند. جلسه‌ای است که به تیم اسکرام کمک می‌کند که ذی‌نفعان را مطلع نگه دارند، و بررسی را با آن‌ها انجام بدهند و ایده‌ها و بازخوردهایی را مبادله کنند و تصمیم بگیرند که قدم بعدی برای محصول چیست.

۹. با اسکرام‌مستر، روزانه، همکاری کنید

به‌عنوان مالک محصول، اسکرام‌مستر (یا مربی چابک) دوست شماست. او می‌تواند به شما کمک کند، مربی شما باشد، با شما مشورت کند، شما را آموزش بدهد و در هر نوع چالشی ناظر شما باشد. با این حال مالکان محصول بسیاری را دیده‌ایم که به‌ندرت با اسکرام‌مستر خود همکاری می‌کنند. تأسف‌آور است! اسکرام‌مستر می‌تواند در هدایت ذی‌نفعان، تسهیل رویدادهای مربوط‌به ذی‌نفعان، و بهبود مهارت‌های مالک محصول در مدیریت بک‌لاگ محصول و ذی‌نفعان و مدیران و غیره کمک کنند. اسکرام‌مستر فقط برای تیم توسعه نیست، اسکرام‌مستر برای شما و سازمان نیز هست. بنابراین برای تحویل ارزش بیشتر همکاری کنید.



۱۰. مشارکت خود را ادامه بدهید و به حفظ پیشرفت پیوسته متعهد باشید

در بسیاری از تیم‌های اسکرام، مالکان محصول طی اسپرینت قطع ارتباط می‌کنند و در دسترس نیستند و یا در پیشرفت پیوسته مشارکتی ندارند. تیم‌های اسکرام بسیاری هستند که گذشته‌نگری اسپرینت (SprintRetrospective) برای آن‌ها رویدادی است که اسکرام‌مستر و تیم توسعه اجرا می‌کنند و مالک محصول حضور ندارد، یا فقط در چند اسپرنیت حضور داشته است. این سوءتفاهم بزرگی دربارهٔ گذشته‌نگری اسپرینت و کمبود بزرگی برای تیم اسکرام است. اگر مالک محصول هستید، شکی نیست که باید در گذشته‌نگری اسپرینت حضور داشته باشید. مالک محصول، که عضو تیم اسکرام به‌حساب می‌آید، باید در پیشروی پیوسته مشارکت داشته باشد. به‌عنوان مالک محصول، شاید اموری باشد که شاید شما بهتر قادر به انجام آن باشید، شاید در همکاری شما و تیم توسعه پیشرفت‌هایی حاصل شود، شاید تصمیمات مهمی را بتوان اتخاذ کرد، پس باید در این جلسه حضور داشته باشید. به‌علاوه مگر نمی‌خواهید ارزش را بیشینه کنید؟ لازم است از پیشرفت‌های تیم اسکرام دربارهٔ فرایندها، تعاملات و ارتباطات و غیره مطلع بمانید. زیرا هرچه تیم مؤثرتر شود شما می‌توانید ارزش بیشتری به مشتریان و کاربران تحویل بدهید.

امیدواریم این ۱۰ راهنمایی برای موفقیت مالکان محصول در چارچوب اسکرام مفید واقع شوند. با تغییر دیدگاهتان به وظایف مالک محصول، می‌توانید در تحویل ارزش بهینه به مشتری‌ها و کاربران موفق شوید.

چند استاندارد سنجش عملکرد اجرایی پروژه

PMOها چگونه عملکرد اجرایی پروژه را بسنجند؟

تقریباً هر مدیر پروژه‌ای حداقل یک یا چندبار در زندگی حرفه‌ای‌اش به مشکل برمی‌خورد. به‌طورمثال مدیر ارشد، مسئولیت گزارش سطح اجرایی پروژه را به او محول می‌کند؛ مدیر پروژه ساعت‌ها مشغول وارد کردن جزئیات فاز در یک نرم‌افزار مدیریت پروژه می‌شود؛ سپس درصد تکمیل را تخمین می‌زند؛ در چندین مورد، گزارش وضعیت باید روزانه باشد؛ برای مدیر پروژه کارهای همین سناریوی معمولی دشوار می‌شود. درنهایت می‌فهمد که انتقال گزارش‌های دقیق عملکرد روزانه چقدر می‌تواند سخت باشد. این دشواری به این علت است که او بیشتر درگیر بهبود اجرای پروژه است؛ ازاین‌رو بعد از یک روز سخت کاری، به‌ندرت وقتی برای گزارش‌ها می‌یابد.

هر شرکتی وقتی پول و زمان و منابعش را در یک پروژهٔ بخصوص سرمایه‌گذاری می‌کند، نگرانی اصلی‌اش بازگشت سرمایه (ROI) است، که از مدیر پروژه و تیمش انتظار دارد. این مدیر پروژه است که باید از تکمیل پروژه طبق زمان‌بندی اطمینان حاصل کند. باید مطمئن شود که پروژه در محدودهٔ بودجهٔ فعلی‌اش پیش می‌رود. مدیر پروژه تاحد زیادی سنجش اجرای پروژه را شفاف می‌کند. اهمیت این شفافیت برای اطمینان از پیشروی پروژه طبق برنامه است. باید نظارت کند که برنامهٔ اصلی پروژه طبق انتظارات مدیریت مرکزی انجام شود. این شفافیت عملکردی مثل یک سیستم هشداردهنده‌ای عمل می‌کند که اگر پروژه ازلحاظ خط زمانی یا بودجه خوب پیش نرود هشدار می‌دهد. دراین‌صورت مدیر پروژه باید با نظارت جدی، تیم را به مسیر برنامه برگرداند.

مدیر پروژه یا دفتر مدیریت پروژه (PMO) مسئول است که اجرای پروژه را در سطح رضایتمندانه‌ای استمرار بدهد. به‌علاوه باید مراقب باشد که هر فاز اجرای پروژه با دقت گزارش و ثبت شود. باید امکان دسترسی به گزارش‌های عملکرد را ممکن سازد. اما پیگیری فرایندها باید آسان و ساده باشد تا برای PMO به بار اضافی بدل نشود. یک PMO باید اندازهٔ بستهٔ کاری مناسب در WBS (ساختار شکست کار) را با دقت فراوان تعریف کند. برای انجام این مسئولیت نیاز به تبعیت از چند مجموعهٔ استاندارد است. لازم است که بهترین استانداردی را که برای نوع پروژهٔ شما و برحسب نیازهای پروژه مناسب است انتخاب کنید.

به‌علاوه لازم است که PMO از یک استاندارد ثابت کاری، که به عملکرد تیم اعتبار می‌دهد، تبعیت کند.

استانداردهای مناسب برای موقعیت‌های معمولی عبارت‌اند از:

SV (واریانس زمان‌بندی)- اگر مقدار SV صفر باشد، پروژه عنوان «طبق زمان‌بندی» یا «به‌موقع» را دارد. اگر مقدار بیش از صفر داشته باشد، عنوان «جلوتر از زمان‌بندی» را می‌گیرد؛ چرا که ارزش بیشتری از برنامهٔ اصلی دارد. اگر کمتر از صفر باشد عنوان «عقب‌تر از زمان‌بندی» یا «تأخیرافتاده» را می‌گیرد.

CV (واریانس هزینه)- اگر مقدار CV صفر باشد، پروژه عنوان «مطابق با بودجه» را می‌یابد. اگر مقدار بیش از صفر باشد، با عبارت «کمتر از بودجه» آن را توصیف می‌کنیم، زیرا ارزش بیشتری از برنامهٔ اصلی کسب می‌کند. اگر کمتر از صفر باشد با عبارت «بیش از بودجه» آن را توصیف می‌کنیم زیرا ارزش کمتری کسب می‌کند.

SPI (شاخص اجرای زمان‌بندی)- اگر مقدار SPI=1 باشد پروژه «مطابق با زمان‌بندی» است. اگر SPI>۱ باشد، پروژه «عقب‌تر از زمان‌بندی» است و اگر SPI<۱ پروژه «جلوتر از زمان‌بندی» است. هر پروژه‌ای که عملکرد و کارکرد کاملی داشته باشد، مقدار SPI آن نزدیک ۱ است.

CPI (شاخص اجرای هزینه)- اگر مقدار CPI=1 باشد پروژه «در محدودهٔ بودجه» است. اگر CPI>۱ باشد پروژه «بیش از بودجه» است و هزینه‌ها باید کنترل شوند. اگر CPI<۱ پروژه «کمتر از بودجه» است. پروژه‌ای که اجرا و کارکرد کاملی داشته باشد باید مقدار CPI نزدیک به ۱ داشته باشد.

باید خودتان آستانهٔ این مقادیر را، که درصورت هر تغییری برحسب آن تغییر می‌کنید، از همان ابتدای کار تعیین کنید. چند مثال را بررسی کنیم. مثلاً تعریف می‌کنید که اگر SPI=0.9 باشد، زمان‌بندی پروژه به وضعیت زرد می‌رسد، که به این معناست که پروژه از زمان‌بندی عقب است. اگر این کاهش ادامه بیابد به وضعیت قرمز می‌رسیم. همچنین اگر خطاهایی را در زمان‌بندی یا بودجهٔ پروژه یافتید، می‌توانید مقدار کسب‌شدهٔ اضافی را محاسبه کنید. به‌این‌ترتیب تصویر واضح‌تری از مشکلات احتمالی پروژه‌تان می‌یابید. این موضوع ساده را به‌خاطر داشته باشید که سنجش عملکرد پروژه به یک شیوهٔ مؤثر بخش بسیار مهمی از کل فرایند مدیریت پروژه است.

5 راهنمایی برای برنامه‌ریزی پروژه

چگونه به اهداف دست یابیم؟

یک کارآفرین همیشه عهده‌دار پروژه‌ها می‌شود. چه خود کارآفرین بداند و چه نداند، کارِ کارآفرین را می‌توان همیشه به گروه‌های مجزایی از وظایف، حول یک اهداف معین، تفکیک کرد؛ به این گروه‌ها پروژه می‌گوییم. به‌طور ایدئال، انجام وظایف باید به تکمیل پروژه‌ها منجر شود و انجام پروژه‌ها به دستیابی به اهداف تجاری می‌انجامد. اما گاهی کارآفرین، خصوصاً آن دسته‌ای از کارآفرینان که تازه کارند، پروژه‌ها را به کفایت اولویت‌بندی نمی‌کنند؛ بلکه کل کار را به یک‌باره انجام می‌دهند. این ایدهٔ خوبی نیست؛ استرس را افزایش می‌دهد؛ احتمال تکمیل کار را کم می‌کند؛ و مانع پیشبرد مؤثر پروژه به‌سمت اهداف تجاری می‌شود. اگر نمی‌خواهید چنین آخر و عاقبتی داشته باشید، ۵ راهنمایی زیر را هنگام شروع و آماده‌سازی و مدیریت پروژه‌هایتان درنظر بگیرید:

۱

پروژه‌هایتان را تعریف کنید

اولین و مهم‌ترین موضوع در برنامه‌ریزیِ پروژه فهمیدن این است که با چه پروژه‌ای سروکار دارید. پس اولاً پروژه چیست؟ به‌سادگی می‌توان گفت هر کاری است که تعداد وظایفش از چند وظیفهٔ ساده بیشتر است. آیا باید نرم‌افزار فروش جدیدی را پیاده‌سازی کنید؟ این یک پروژه است. شبکه‌تان را باید گسترش بدهید؟ بله، این هم یک پروژه است. می‌خواهید کسب‌وکارتان را توسعه دهید؟ این مجموعه‌ای از پروژه‌هاست!

برخی از پروژه‌ها کاملاً بدیهی‌اند اما برخی دیگر در کمون و ناشناخته هستند. برای شناسایی تمام پروژه‌ها هر کاری را که اخیراً انجام می‌دهید یا می‌خواهید انجام بدهید یادداشت کنید. سپس برحسب اهداف معینتان، این لیست را گروه‌بندی کنید.

با این کار به لیست پروژهٔ خود دست می یابید. شاید در این لیست، تعداد ۱۰ تا ۲۰ (یا حتی بیشتر) پروژه داشته باشید؛ اما برای بازدهی بیشتر فقط ۲ یا ۵ پروژه را، که از همه مهم‌ترند، عهده‌دار شوید. در ادامهٔ این راهنمایی همین چند پروژه را درنظر بگیرید.

۲

برای پروژه‌ها تاریخ شروع و پایان تعیین کنید

باید بدانید یا تصمیم بگیرید که هر پروژه چه‌موقع شروع می‌شود و چه‌موقع به‌پایان برسد. به‌این‌ترتیب می‌توانید توالی پروژه‌ها را برحسب تاریخ شروع یا پایان هریک اولویت‌بندی کنید. به‌علاوه با تاریخ‌های شروع یا پایان می‌توانید برحسب چرخه‌های تجاری خود برنامه‌ریزی کنید و مانع شوید که پروژه‌ها بیش از نیاز پیشروی کنند.

۳

یک برنامهٔ یک‌صفحه‌ای تعیین کنید

فقط کمی برنامه‌ریزی در ابتدای کار کمک بزرگی به موفقیت پروژه خواهد بود. برای تعیین یک برنامهٔ یک‌صفحه‌ای، پاسخ‌های هریک از سؤال‌های زیر را یادداشت کنید:

- پروژه چه مشکلی را حل می کند؟
- چه مزایایی برای سازمان دارد؟
- به چه کارکنان یا بودجه‌ای نیاز است؟
- رویدادگاه‌ها (مایلستون‌های) اصلی کدام‌اند؟
- چه موانع بالقوه‌ای وجود دارد؟
- این پروژه چه ریسک‌هایی دارد؟
- برنامهٔ کاری من چیست (لیست وظایف و موارد فرعی چه‌موقع باید تکمیل شود)؟
- معیارهای مشخص کنندهٔ موفقیت کدام‌اند؟
- چگونه می‌فهمید پروژه انجام شده است؟

توجه کنید که اگر بتوانید به این سؤال‌ها پاسخ‌های مختصر و مفید بدهید، می‌توانید هماهنگی برنامه را حفظ کنید.

۴

ارتباطات را روتین کنید

برای هر پروژه‌ای که چندین عضو تیم پروژه دارد، ارتباطات را از قبل به‌صورت روتین درآورید. آیا کسی را برای گزارش هفتگی درنظر گرفته‌اید؟ یا گزارش ماهانه؟ آیا ذی‌نفعان برای بررسی فراخوانده می‌شوند؟ چه کسی مسئول نظارت بر پروژه است، اینکه پروژه چه‌موقع به رویدادگاه‌هایش (مایلستون‌هایش) می‌رسد؟

۵

بدانید چه‌موقع لازم است کنار بکشید

گاهی لازم است که دوشاخه را از برق بیرون بکشید؛ هرچند بهتر است کاری را که شروع می‌کنیم به‌پایان برسانیم. اگر تمام پروژه‌های شما منظم هستند و تاریخ‌های شروع و پایان و یک برنامهٔ شفاف یک‌صفحه‌ای دارند، که تمام رویدادگاه‌ها (مایلستون) و دستورالعمل‌های ارتباطی در آن لحاظ شده است، از کل اطلاعات لازم برای تشخیص زمانِ دست‌کشیدن از پروژه برخوردارید.

بی‌تردید دست‌کشیدن همیشه آسان نیست، خصوصاً اگر زمان و انرژی و عاطفهٔ زیادی صرف کرده باشید. اما اگر یک پروژه ممتد شد، با چند سؤال زیر می‌توانید بفهمید که باید متوقف شود یا خیر:

- آیا هدف این پروژه هنوز برای سازمان من مهم است؟
- نرخ بازگشت سرمایه یا ROI برنامه‌ریزی‌شده برای این پروژه چقدر خواهد بود؟ اکنون ROI چیست؟
- هزینهٔ احتمالی ادامهٔ پروژه چیست؟ آیا اگر بر دیگر پروژه‌ها وقت صرف کنم بیشتر به اهدافم دست نمی‌یابم؟
- ادامهٔ این پروژه چه تلفات عاطفی و معنوی‌ای به‌همراه خواهد داشت؟

مدیر پروژه کیست؟

مدیران پروژه نقش هدایت برنامه‌ریزی، اجرا، نظارت، کنترل و اتمام پروژه‌ها را به‌عهده دارند. در این مقاله نقش مدیر پروژه، مسئولیت‌ها، مدارک مرتبط و حقوق موردانتظار را بررسی می‌کنیم و راهنمایی‌هایی برای کاریابی عرضه می‌کنیم.

نقش هدایتگر به موفقیت پروژه

یک مدیر پروژه کیست؟

مدیران پروژه نقشی کلیدی در برنامه‌ریزی، اجرا، نظارت، کنترل و اتمام پروژه‌ها بازی می‌کنند. آن‌ها مسئول کل محدودهٔ پروژه، تیم پروژه، منابع و موفقیت و شکست پروژه هستند. اگر جویای حرفه‌ای در فناوری اطلاعات هستید و جایگاه مدیریت پروژه را برای خود بررسی می‌کنید، با یک نقشهٔ راه مناسب به‌سمت مدیریت پروژه‌های آی‌تی حرکت کنید. همچنین تحقیق کنید که برای یک مدیر پروژهٔ عالی شدن به چه چیز نیاز است.

مسئولیت‌های مدیر پروژه

یک مدیر پروژه، با کمک تیمش، عهده‌دار چندین مسئولیت مربوط به پنج فاز چرخهٔ حیاتی پروژه (تکوین، برنامه‌ریزی، اجرا، نظارت، و اتمام) است. فازهای مدیریت پروژه با ۱۰ حوزهٔ دانش هم‌پوشانی دارد.

حوزه‌های این دانش عبارت‌اند از یکپارچه‌سازی، محدوده پروژه، زمان، هزینه، کیفیت، منابع انسانی، ارتباطات، کارپردازی ریسک و مدیریت ذی‌نفعان.

۱. فاز آماده‌سازی:

۱,۱ مدیریت یکپارچگی: توسعهٔ منشور پروژه

۱,۲ مدیریت ذی‌نفعان: شناسایی ذی‌نفعان

۲. فاز برنامه‌ریزی:

۲,۱ مدیریت یکپارچگی: توسعهٔ برنامهٔ مدیریت پروژه

۲,۲ مدیریت محدوده: تعریف و مدیریت محدوده، ساخت یک ساختار شکست کار (WBS) و مجموعهٔ نیازمندی‌ها

۲,۳ مدیریت زمان: برنامه‌ریزی، تعریف، توسعهٔ زمان‌بندی، فعالیت‌ها، تخمین منابع و طول‌مدت فعالیت

۲,۴ مدیریت هزینه‌ها: برنامه‌ریزی و تخمین هزینه‌ها و تعیین بودجه‌ها

۲,۵ مدیریت کیفیت: برنامه‌ریزی و شناسایی نیازمندی‌های کیفی

۲,۶ مدیریت منابع انسانی: برنامه‌ریزی و شناسایی نیازهای منابع انسانی

۲,۷ مدیریت ارتباطات: برنامه‌ریزی ارتباطات

۲,۸ مدیریت ریسک: برنامه‌ریزی و شناسایی ریسک‌های بالقوه، اجرای تحلیل‌های کیفی و کمی ریسک و برنامه‌ریزی راهبردهای کاهش ریسک

۲,۹ مدیریت کارپردازی: برنامه‌ریزی و شناسایی امور و تدارکات موردنیاز

۲,۱۰ مدیریت ذی‌نفعان: برنامه‌ریزی برای انتظارات ذی‌نفعان

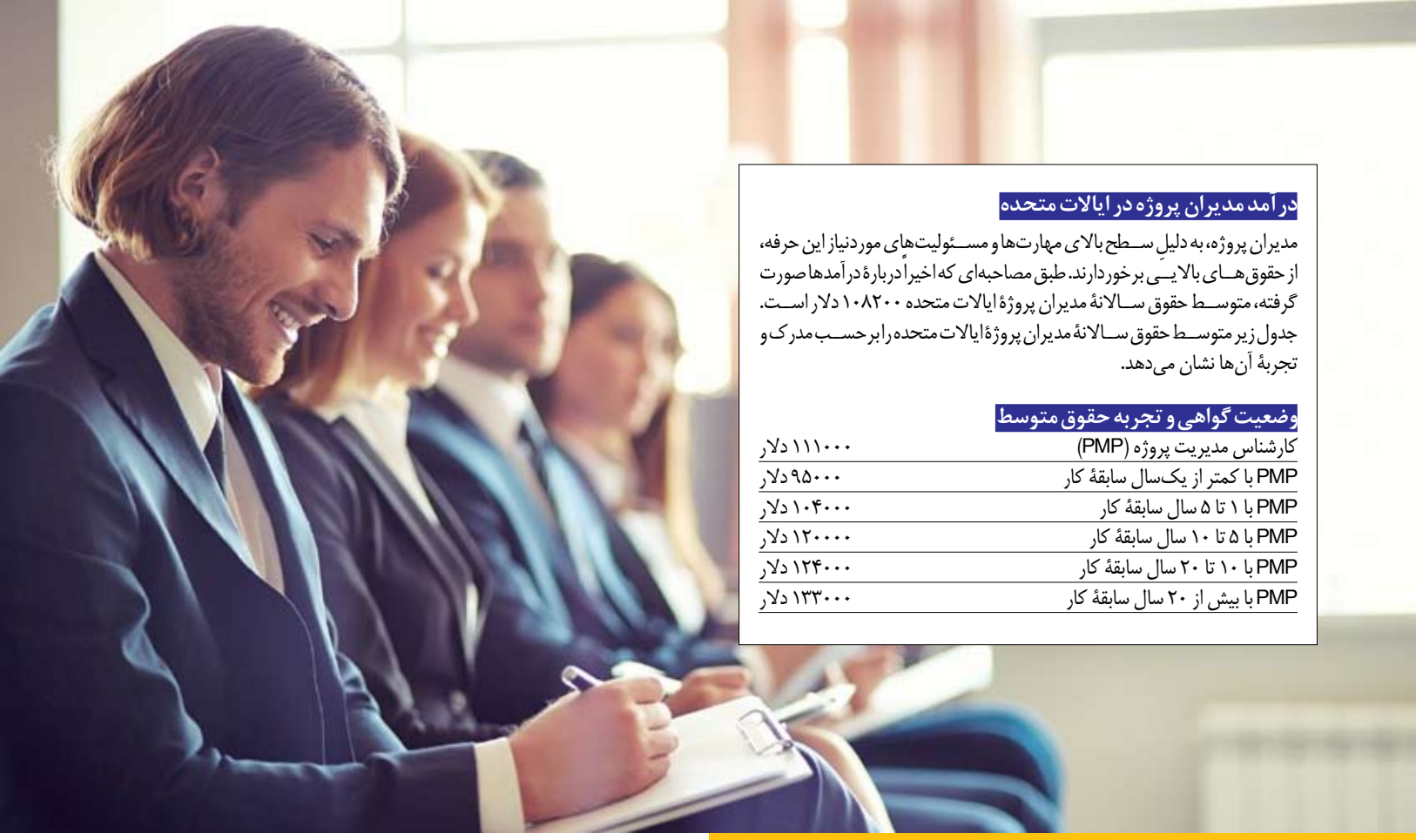
۳. اجرا:

۳,۱ مدیریت یکپارچه‌سازی: هدایت و مدیریت تمام کار پروژه

۳,۲ مدیریت کیفیت: اجرای تمام جنبه‌های مدیریت کیفیت

۳,۳ مدیریت منابع انسانی: انتخاب، توسعه و مدیریت تیم پروژه

۳,۴ مدیریت ارتباطات: مدیریت تمام جنبه‌های ارتباطات



درآمد مدیران پروژه در ایالات متحده

مدیران پروژه، به دلیلِ سطح بالای مهارت‌ها و مسئولیت‌های موردنیاز این حرفه، از حقوق‌های بالایی برخوردارند. طبق مصاحبه‌ای که اخیراً دربارهٔ درآمدها صورت گرفته، متوسط حقوق سالانهٔ مدیران پروژهٔ ایالات متحده ۱۰۸۲۰۰ دلار است. جدول زیر متوسط حقوق سالانهٔ مدیران پروژهٔ ایالات متحده را بر حسب مدرک و تجربهٔ آن‌ها نشان می‌دهد.

وضعیت گواهی و تجربه حقوق متوسط

کارشناس مدیریت پروژه (PMP)	۱۱۱۰۰۰ دلار
PMP با کمتر از یک‌سال سابقهٔ کار	۹۵۰۰۰ دلار
PMP با ۱ تا ۵ سال سابقهٔ کار	۱۰۴۰۰۰ دلار
PMP با ۵ تا ۱۰ سال سابقهٔ کار	۱۲۰۰۰۰ دلار
PMP با ۱۰ تا ۲۰ سال سابقهٔ کار	۱۲۴۰۰۰ دلار
PMP با بیش از ۲۰ سال سابقهٔ کار	۱۳۳۰۰۰ دلار

مدارک مربوط به مدیریت پروژه

داشتن مدرک یا گواهی در مدیریت پروژه می‌تواند درها را بر فرصت‌های شغلی بیشتر و شغل‌های پردرآمدتر بگشاید. اگر دوره‌های باز و رایگان مدیریت پروژه را می‌پسندید، می‌توانید چندی از این دوره‌ها را در سایت‌های مثل کورسِرا بیابید تا دانش و مهارت‌های مربوط به پروژه را در خودتان تقویت کنید. بسیاری از کسب‌وکارها رَوِشگان (متدولوژی) چابک (Agile) را ترجیح می‌دهند که به موجب آن، کارکنان به کَسب دانش در این حوزه نیاز پیدا می‌کنند. با مدارک بین‌المللی مربوط به چابک می‌توانید حرفهٔ خود را به سطح بعدی برسانید.

۳,۵ مدیریت کارپردازی: امن‌سازی کارپردازی‌های موردنیاز
۳,۶ مدیریت ذی‌نفعان: مدیریت تمام انتظارات ذی‌نفعان

۴. نظارت و کنترل:

۴,۱ مدیریت محدوده: نظارت و کنترل کار پروژه و مدیریت تغییرات لازم
۴,۲ مدیریت محدوده: اعتبارسنجی و کنترل محدودهٔ پروژه
۴,۳ مدیریت زمان: کنترل محدودهٔ پروژه
۴,۴ مدیریت هزینه‌ها: کنترل هزینه‌های پروژه
۴,۵ مدیریت کیفیت: کنترل کیفیت اقلام تحویلی
۴,۶ مدیریت ارتباطات: کنترل تمامی ارتباطات تیم و ذی‌نفعان
۴,۷ مدیریت کارپردازی: کنترل کارپردازی‌ها
۴,۸ مدیریت ذی‌نفعان: کنترل مشارکت ذی‌نفعان

۵. بستن (اتمام):

۵,۱ مدیریت یکپارچه‌سازی: اتمام تمام فازهای پروژه
۵,۲ مدیریت کارپردازی: اتمام تمام کارپردازی‌های پروژه

علاوه بر دانش فنی ذکرشده مدیر پروژه به مهارت‌های پرمطالبهٔ دیگری نیاز دارد که «مهارت‌های نرم» یک نوع از این مهارت‌هاست که به ایفای این نقش به بهترین شکل کمک می‌کند. با کسب این مهارت‌ها مدیران پروژه می‌توانند خود را از دیگر همکارانشان متمایز کنند.

شغل‌های مدیریت پروژه

مهم‌ترین نکته در مدیریت پروژه این است که تقریباً هر بخش صنعت در سرتاسر دنیا به مدیران پروژه با تخصص‌های مختلف نیاز دارد و همین مدیریت پروژه را به گزینهٔ حرفه‌ای مناسبی تبدیل کرده است. شرکت‌ها در زمینهٔ بهداشت عمومی، هوانوردی، فناوری، توسعهٔ نرم‌افزار، مهندسی، ساخت‌وساز، املاک، انتشارات، امور مالی و دارائی، بازاریابی، تولید، آموزش، بیمه، امور دولتی و زمینه‌های متعدد دیگر به مدیران پروژهٔ شایسته نیاز دارند و در جست‌وجوی آن‌ها هستند. درواقع ادارهٔ آمار کار ایالات متحده چشم‌انداز روشنی را برای مدیران پروژه پیش‌بینی کرده است. اساساً هر کسب‌وکاری که پروژه دارد به مدیران پروژه نیازمند است. امید است که همزمان با پیشرفت حرفهٔ مدیریت پروژه در دنیا، روزبه‌روز در ایران نیز شاهد توسعهٔ این حرفه، و درنتیجه، نظام‌مندی و بهینه‌سازی تلاش‌های کسب‌وکارهای بومی‌مان باشیم.

مدیریت پروژه

7 قدم برای تحویل هدف‌محور پروژه، درمحدودهٔ زمان‌بندی و بودجه

مدیران پروژه‌ای که این ۷ راهنمایی پیشرفته را دنبال می‌کنند، می‌توانند پروژه‌های سطح‌بالا را در محدودهٔ بودجه و زمانِ تعیین‌شده و برحسب هدف تحویل دهند و متعاقباً با افزایش نرخ‌های موفقیت پروژه‌هایشان، چشم‌انداز شغلی بهتری را رقم بزنند. طبق گزارش جهانی سال ۲۰۱۷ با عنوانِ «ضربان حرفه» (Pulse of the Profession)، سازمان‌ها، به‌ازای هر یک میلیارد دلاری که سرمایه‌گذاری می‌کنند، ۹۷ میلیون دلار به‌دلیل اجرای ضعیف پروژه از دست می‌دهند. هرچند شاهد کاهش ۲۰ درصدی نرخ شکست پروژه‌ها بوده‌ایم، این ارقام همچنان تکان‌دهنده است. مهترین و پربسامدترین علت‌های شکست پروژه چه هستند؟ از سال ۲۰۰۶ سالانه مصاحبه‌ای، در سطح جهانی، از کاروَرانِ مدیریت پروژه انجام می‌شود و هدف از این مصاحبه ایجاد چشم‌اندازی از گرایش‌های کنونی و بعدی در مدیریت پروژه است.

این بررسی بر بازخوردها (فیدبک) و بینش‌های کسب‌شده از مدیران پروژه، برنامه، و سبد پروژه (پورتفولیو) تأکید دارد. به‌علاوه تحلیلی از داده‌های شخص ثالث حاصل می‌شود که مسیر پیشِ روِ مدیران پروژه را برای موفقیت روشن می‌کند. در این بررسی نرخ‌های موفقیت برحسب دو نوع مجری پروژه تعریف می‌شوند: برنده‌ها و کم‌کارآمدها. متوسط نرخ موفقیت این‌دو به‌قرار زیر است:

پارامترها	برنده‌ها	کم‌کارآمد
به‌موقع بودن پروژه	۸۸ درصد	۲۴ درصد
در محدودهٔ بودجه بودن پروژه	۹۰ درصد	۲۵ درصد
برآوَردن اهداف یا قصد تجاری	۹۲ درصد	۳۳ درصد
تجربهٔ خزش محدوده (scope creep)	۲۴ درصد	۶۸ درصد
پروژه‌های شکست‌خورده	۶ درصد	۲۴ درصد
تلفات بودجه به‌علت پروژه‌های شکست‌خورده	۱۴ درصد	۴۶ درصد
بلوغ در شناخت مزایا	زیاد	کم

جای تعجب ندارد که نرخ‌های موفقیت پروژهٔ برنده‌ها، در مقایسه با ۳۳درصد کم‌کارآمدها، بسیار بالاتر است (تا ۹۲ درصد).
درگذشته وقتی پروژه‌ها ناموفق می‌شدند توجه‌ها بیشتر به عوامل مربوط به پروژه معطوف می‌شد. معمولاً از این واقعیت چشم‌پوشی می‌شد که نرخ‌های موفقیت و شکست پروژه عمدتاً ریشه در عوامل ریشه‌دار سازمانی است. این عوامل تقریباً باعث می‌شد که کارشناسان پروژه عملاً نتوانند به اهداف پروژه، که یا هم‌راستا و یا واضح نبود، دست یابند.
در این مقاله ۷ قدم کلیدی برای تحویل اهداف پروژه، طبق بودجه و زمانِ تعریف‌شده، معرفی می‌شود. اما توجه کنیم که اگر رهبران پروژه و شرکت می‌خواهند که ذی‌نفعان به نتایج مستمر و مطلوبشان برسند، باید در تمام این حوزه‌ها همگام و هماهنگ باشند.

۱

۱. اهداف راهبردی شرکتی را با EPMO (دفتر سازمانی مدیریت پروژهٔ) تنظیم کنید

راهبرد را نقشهٔ راهی در نظر بگیرید که هدایتگر تمام اقدامات شرکت است. بدون این نقشهٔ راه، تیم‌ها در سازمان‌ها کورکورانه ریسک می‌کنند، و همین اهداف درازمدت را در خطر می‌اندازد. ازاین‌رو بیشتر شرکت‌ها دفترهای سازمانی مدیریت پروژه یا EPMOها را با آغوش باز می‌پذیرند. EPMO یک کارکرد راهبردی مرکزی در سطح تجاری دارد و هدف آن پشتیبانی شرکت در مسائلی چون راهبری، اتخاذ بهترین روش‌ها، آموزش، نظارت، شناسایی، به‌کارگیری ابزار و فنون، و استانداردسازی فرایندهای تجاری است. طبق گزارش ذکرشده، «دربین سازمان‌هایی که در این مصاحبه PMO داشتند، نیمی گزارش می‌دهند که دفتر سازمانی مدیریت پروژه دارند.
و مصاحبه‌شوندگانی که EPMOشان با راهبردشان هم‌راستاست (مثلاً یک EPMO راهبردی دارند) گزارش می‌کنند که پروژه‌هایشان ۳۸ درصد بیشتر اهداف و قصد تجاری را برآورده کرده‌اند و ۳۳ درصد کمتر پروژه‌های محکوم به‌شکست داشتند.»

۲

۲. مدیریت شناخت مزایا را تمرین کنید

علاوه بر خط‌مشی‌های شرکت و EMPO، شناسایی و اجرا و استمرار یک «برنامهٔ شناخت مزایا» اهمیت دارد.
در گزارش ذکرشده آمده: «مدیریت شناخت مزایا (BRM) راه قدرتمندی برای هم‌راستایی پروژه‌ها، برنامه‌ها و سبدهای پروژه (پورتفولیو) با راهبرد کلی سازمان است. اما این شیوه برای بسیاری وحشت ایجاد می‌کند، زیرا توافق نظری دربارهٔ یک فرایند BRM واحد وجود ندارد. بااین‌همه بیشتر سازمان‌ها برای تعیین روند شناسایی مزایا و نظارت بر پیشبرد آن در چرخهٔ حیات پروژه تا اکتسابش اقداماتی انجام می‌دهند. درواقع ۳۱ درصد از سازمان‌ها در این گزارش از بلوغ در شناخت مزایا خبر می‌دهند.
با مدیریت شناخت مزایا «چگونگی» ارتباط پروژه با راهبرد را تعیین می‌کنیم. برای دست‌یابی به این مهم، یک «راهنمای چارچوب مدیریت شناخت مزایا» تهیه شده که سازمان‌ها با آن می‌توانند:
۱. مزایا را شناسایی کنند تا بتوان تعیین کرد که آیا پروژه‌ها، برنامه‌ها و سبدهای پروژه قادر به حصول نتایج تجاری مطلوب هستند.
۲. مزایا را به اجرا درآورند، به‌طوری که ریسک‌های مزایای آینده کمینه شوند و شانس دستیابی به مزایای اضافی بعدی، بهینه.
۳. مزایا را حفظ کنند تا ارزش‌افزایی ماحصلِ برنامه یا پروژه استمرار بیابد.

۳

۳. پلی بین فرمول‌بندی راهبردها و اجرای آن‌ها بسازید

موضوع بعدی که باید به آن رسیدگی شود چگونگی اجرای راهبردهای شرکتی است. در گزارش مشخص می‌شود که در ۱۲ ماه گذشته، فقط ۶۰درصد از سازمان‌ها مطلع‌اند که چگونه بین شناخت راهبرد و اجرای آن پلی بسازند. منظور از پل‌سازی هر کار لازم است؛ از شناسایی استعدادها و منابع و جلب موافقت حامیان اجرایی، تیم‌ها و ذی‌نفعان گرفته تا انتخاب رَوِشگان‌ها (متدولوژی) و ابزار مناسب. تعیین اجرای راهبرد قدم مهمی است. در این مرحله خوب است که شتاب امور را کم کنیم و بررسی کنیم که معیارها همگی در نظر گرفته شده و به‌دقت برنامه‌ریزی شده باشند. بی‌توجهی به این قدم بعدها عواقب پرهزینه‌ای دارد.

۴

۴. جلب و حفظ حمایت اجرایی (executive sponsorship)

نقش حامی اجرایی پشتیبانی از پروژه‌های شرکت و مرتبط نگه داشتن پروژه‌ها و اهداف شرکت است. حامیان اجرایی از پروژه‌ها دفاع نیز می‌کنند؛ مشارکت‌ها را تقویت می‌کنند؛ به حل‌وفصل کشمکش‌ها و رفع مشکلات کمک می‌کنند؛ هدایتگری می‌کنند؛ و بودجه و منابع موردنیاز برای موفقیت را تعیین می‌کنند.
به‌علاوه فرهنگ تعامل نیز در همین سطوح اجرایی شکل می‌گیرد و فرهنگ تیمی و سطوح همکاری تعیین می‌شوند. نبودن حمایت کافی در پروژه موجب تأثیرات منفی می‌شود. در گزارش نشان داده شده که «حامیان اجرایی‌ای که مشارکت فعال دارند مهترین پیش‌بران پروژه به‌سمت دستیابی به اهداف و قصد تجاری‌شان هستند. این واقعیت در مصاحبه با پاسخ‌دهندگان پوشیده نمانده؛ مصاحبه‌شوندگان نشان دادند که نسبت به سال پیش، درصد مشارکت فعالانهٔ حامیان در پروژه‌هایشان افزایش یافته است؛ به ترتیب ۶۲ درصد در مقایسه با ۵۹ درصد.

۵

۵. استعدادهای مناسب را استخدام کنید و نگه دارید

پروژه‌ها عمدتاً به دلیل افرادی که آن را اجرا می‌کنند موفق هستند. گزینش، آموزش و ارتقای استعدادها و نظارت بر آن‌ها در چگونگی تعامل تیم‌ها و افراد، و اجرای پروژه‌ها ضروری است. درصد اولویت‌بندیِ شرکت‌های برنده عبارت‌اند از: برای توسعهٔ مهارت‌های فنی (۷۶درصد در مقایسه با ۱۹درصد کم‌کارآمدها)، مهارت‌های رهبری (۷۶درصد در مقایسه با ۱۶درصد کم‌کارآمدها) و مهارت‌های مدیریت تجاری و راهبردی (۶۵درصد در مقایسه با ۱۴درصد کم‌کارآمدها).

تمام این حوزه‌ها ضروری هستند. با سرمایه‌گذاری در افراد شانس جلب موافقت بیشتر می‌شود. درنتیجه افراد باور می‌کنند که مشارکت آن‌ها کلیدِ دستیابی به اهداف است و امکانِ اینکه مهارت‌ها و دانش خود را کاملاً به کار بگیرند بیشتر می‌شود.

۶

۶. چابک (Agile) را به کار بگیرید

شناسایی و گزینش و به‌کارگیریِ بهترین روش پروژه برای دستیابی به نتایج مطلوب ضرورت دارد. درواقع گزارش ۱۲ ماه گذشته نشان می‌دهد که ۲۰ درصد از پروژه‌ها با استفاده از چابک اجرا شدند و «۷۱ درصد از سازمان‌ها گزارش می‌دهند که گهگاه و با بسامد بیشتری نسبت‌به گذشته از رویکردهای چابک استفاده می‌کنند.». این گزارش حاکی است که «۵۵ درصد برنده‌ها (در مقایسه با ۴۲ درصد از کم‌کارآمدها) تمرکز شدیدی بر به‌کارگیری رویکردهای چابک در پروژه‌هایشان دارند.»

۷

۷. اختلال‌های تجاری مربوط به فناوری را مدیریت کنید

در گزارش گفته شده: «انقلاب ارتباطات همچنان با وسایل الکترونیک دستی مثل گوشی‌های هوشمند و رایانه‌های تبلتی شدت می‌گیرد و حتی به کشورهای فقیر و دورافتاده نیز سرایت کرده و قریب است که به ازدیاد پروژه‌های توزیع‌شده در ملل و صنایع گوناگون بینجامد.» با شتابی که این روند گرفته، شرکت‌ها تحت فشار خواهند بود که از فناوری‌هایی استفاده کنند که در برآوردن تقاضاهای روبه‌رشدِ ذی‌نفعان از خود چابکی نشان بدهند. مقاومت در برابر تغییر ممکن است که شرکت‌ها را به وضعیت رکود برساند و مانع چابک شدن و مؤثر بودن تیم‌های پروژه شود. طبق گزارش، «رهبران سازمان‌ها راهبردهای خود را با هدف واکنش مناسب به اختلالات تجاری تنظیم می‌کنند. برای بسیاری این راهبردها انتخاب بین «پذیرفتن یا مردن» است. مدیران اجرایی دریافته‌اند که برای هم‌ارتعاشی با بازار باید این گذار را هدایت کنند، نه اینکه فقط با آن هدایت شوند.»

چه کنیم که بودجهٔ یک پروژهٔ آی‌تی را به باد ندهیم

بسیاری از دوستان، همکاران و مشتریان پیش‌بینی می‌کنند که بودجه‌های آی‌تی در سال ۲۰۱۷ همچنان افزایش بیابند؛ اما مدیران ارشد فناوریِ بسیاری در یک مصاحبهٔ اخیر (در بریتانیا) گفته‌اند که مدیریت و به‌کارگیری این بودجه‌ها دشوارتر می‌شود. دلیلی دارد. بودجه‌های بزرگ به معنی مسئولیت‌های بزرگ است و هر سازمانی بیشترین بازگشت سرمایه ممکن را می‌خواهد. در مصاحبهٔ Apptio با مدیران ارشد فناوری، ۶۷درصد از آن‌ها و دیگر مدیران ارشد اجرایی آی‌تی انتظار دارند که بودجه‌هایشان یا افزایش بیابد یا در طول سال ثابت بماند؛ اما بیشتر از نیمی گفته‌اند که این بودجه‌های زیاد چالش‌هایی را با خود به‌همراه می‌آورند. از هرکسی که تابه‌حال کاهش بودجه داشته بپرسید؛ تخصیص بیشتر دلیلی برای خوشحالی است نه ترسیدن! درادامه هفت راهبرد ارائه می‌کنیم که هیچ‌وقت بودجهٔ یک پروژهٔ آی‌تی را به باد ندهید.

۱

برای تغییر، یک بودجهٔ جدا تخصیص بدهید

در این راهبرد ساده، برای هر تغییرِ تأییدشده، یک بودجهٔ ذخیره (Reserve Fund) ایجاد می‌شود. علت به دردسر افتادن بسیاری از پروژه‌ها این است که هرگاه تغییری تأیید می‌شود مدیریت پروژه باید برای آن، بودجه پیش‌بینی کند و موافقت آن را اخذ کند. به‌این ترتیب از پیش با تأمین وجه موافقت می‌شود. برای فراهم کردن بودجه برای تغییر باید درخواست‌های تغییر را برحسب تجربه و آگاهی‌تان از پروژه و منابع، طی حیات پروژه، تخمین بزنید. وقتی که تأیید شد معمولاً به مدیر پروژه بستگی دارد که بودجه را تسهیم کند. بااین‌همه بعضی از پروژه‌ها که فقط یک بودجه برای اتفاقات احتمالی (Contingency Fund) تخصیص می‌دهند، در تلهٔ سرگردانی بودجهٔ تغییر گرفتار می‌شوند و به‌زودی تمام شدن آن را نظاره می‌کنند. ارتباطات نیز مثل دیگر جنبه‌های پروژه پراهمیت است. اگر پروژه از خارج از سازمان است، هربار که برای هزینه‌ای، از بودجهٔ تغییر استفاده می‌شود باید به اطلاع مشتری برسد؛ و اگر پروژه داخلی است به اطلاع ذی‌نفعان برسد. به‌این ترتیب در انتهای هر پروژه هرکسی می‌داند چقدر از بودجه استفاده شده و چقدر در چنته مانده است.

۲

بودجه را دائماً بررسی کنید

بودجه‌های پروژهٔ آی‌تی مانند رژیم‌های غذایی می‌مانند. اگر دائماً مراقب هر لقمهٔ خود نباشید شکست منتظر شما می‌نشیند. اگر مدیر پروژه‌ای بداند که چطور رژیم بگیرد، می‌داند که چطور پروژه‌ها را لقمه‌لقمه کند! ذهنیت رژیم لاغری بر «سالم‌خواری و ورزش بیشتر در همین امروز» استوار است؛ می‌توان این ذهنیت را در مدیریت پروژه‌های آی‌تی به‌کار برد. بارها شده که با رویکرد «همین امروز» سه سایز کم کردند! در نظارت بر بودجه نیز این رویکرد درست است. منابع را باید تفکیک کرد و برحسب اندازهٔ هر لقمه، بودجه تخصیص داد و با بررسی (حداقل) هفتگی، بودجه را به لقمهٔ بعدی هدایت کرد.

آیا کار آسانی است؟ اگر وزن کم کردن آسان باشد، این کار نیز آسان است. اما وقتی که عادت‌سازی می‌کنیم، عادت‌هایی که با اهداف ما هم‌راستا هستند، چه با خوردن سالاد باشد و چه با از پله بالا رفتن یا پیش‌بینی هفتگی میزان مصرف بودجه، دشوار برای ما آسان می‌شود.

پروژه را با راهبرد تجاری هم‌راستا کنید

۳

به‌نظر می‌رسد ۹۰ درصد پروژه‌هایی که با مشکلات بحرانی بودجه روبه‌رو می‌شوند با مأموریت تجاری سازمان هم‌راستا نیستند. در حالی که ۹۹ درصد از پروژه‌هایی که در محدودهٔ بودجه یا زیر بودجه تحویل داده می‌شوند هم‌راستایی مناسبی دارند. شاید این رقم به تحقیق بیشتری نیاز داشته باشد، اما تجربهٔ بیشتر مدیران پروژه این رقم حسی را تأیید می‌کند. وقتی یک پروژه با خط‌سیر مأموریت کسب‌وکار شرکت هم‌سو است، احتمال مصرف بیش‌ازحد بودجه کمتر می‌شود. شاید دلیلش آن است که همه حس مشترکی از سفر پیش‌رو شرکت دارند و ارزش پایین نگه‌داشتن هزینه‌های سفر را فهمیده‌اند؛ شاید راهبردهای هوشمندانه بستری برای راهبردهای هوشمندانهٔ بعدی فراهم می‌کنند؛ و شاید وقتی که پاسخگویی افراد افزایش می‌یابد، بینش بهتری از هدایتگری و مسئولیت‌پذیری می‌یابند... هرکدام می‌تواند دلیلی باشد! اما شکی نیست که پروژه‌های IT‌ائی که مورد تجاری ندارند، محتمل‌ترین پروژه‌هایی هستند که در بیابان کمبود بودجه سرگردان می‌شوند؛ زیرا هیچ راهبردی برای سنجش آن‌ها تعیین نشده است.

۴

کمک بگیرید

جای تشویق دارد که بیشتر از نیمی گفته‌اند که مدیریت بودجه‌های بزرگ‌تر برای آن‌ها چالش بیشتری دارد. حداقل همین که چالش را انکار نمی‌کنند عالی است.

خجالت‌آور نیست که اعتراف کنیم که گاهی شرح کار مدیریت پروژه جزء خصوصیات قدرتمند ما نیست. مادام اینکه ضعف‌های خود را انکار کنیم راهی برای برطرفی آن نداریم. سال پیش یک مدیر پروژه در بخش عمومی بریتانیا با شجاعت اعتراف کرد که بودجه‌های افزوده‌شده و سبد پروژه از ظرفیت تخصیص بودجهٔ او فراتر رفته است، بنابراین متحمل بار کار اضافی شد تا درخواست کند چند استعداد جدید برای تیم استخدام کنند. یک حرفه‌ای را به کار گرفتند که در کمال تعجب در تخصیص بودجه حرفه‌ای بود! نکته اینجاست که می‌توانید استعدادهایی را برای جبران نقاط ضعفتان به کار بگیرید و کمبود بودجه فرصت فوق‌العاده‌ای برای این بازبینی است. اگر نمی‌توانید تعداد کارکنان را زیاد کنید، با مدیریت پروژه به‌طور خدماتی (Project management as Service) (برون‌سپاری دفتر مدیریت پروژه) می‌توانید از راهکارهای دیگر افراد برای پایان دادن به مدیریت پروژه، و درنتیجه، به مدیریت بودجه استفاده کنید. به‌علاوه با ارزیابی یک دفتر مدیریت پروژه، که هم‌چون چشمان تیزبین همیشه مراقب عملیات شماست، می‌توان با راهکارهای ساده، مشکلات مربوط به بودجه را برطرف کرد.

۵

طی چرخهٔ حیات پروژه، میزان دسترسی و نیاز به منابع را پیش‌بینی کنید

سال‌ها پیش مدیر افسانه‌ای فوتبال، برایان کِلاگ، دربارهٔ راهبردهایش صحبت می‌کرد؛ اینکه چگونه تیم ناتینگهام فارست را به پیروزی دو جام اروپا رساند. یکی از راهبردهایش دانستن این بود که هر بازیکن باید در چه فازی از بازی قرار می‌گرفت.

این فن بسیار ساده را مدیران پروژه‌های آی‌تی نیز می‌توانند تمرین کنند تا به پیروزی دست بیابد. اگر در دسر بخش کردن پروژه به تکه‌های ریز و تخصیص بودجه به هر فاز را تحمل کنید، قدم بعدی ساده‌تر است: بررسی دسترسی به منبع. با این کار نه‌فقط با اطمینان از پوشش یافتن تمام امورِ پایه‌ای فراغ بال می‌یابید، بلکه درصورت وجود خلأ، متوجه می‌شوید و با اولویت‌بندی مجددِ مابقی سبد پروژه یا با مدیریت پروژه به‌طور خدماتی (PMaaS) این خلأها را پر می‌کنید.

۶

مدیریت محدوده

همه می‌دانیم که اگر محدودهٔ مدیریت پروژهٔ خود را مدیریت نکنیم، بعید نیست که بعداً برای مدیریت بودجه به دردِ سر بیفتیم. علی‌رغم اینکه همه این را می‌دانیم، خزش محدوده (Scope Creep) هنوز یکی از قاتلان پروژه‌هاست. متأسفانه با افزایش ابعاد پروژه و میزان بودجه، خطرات خزش محدوده نیز افزایش می‌یابد. در این مقاله نمی‌توانیم یک دورهٔ آموزشی از نحوهٔ تعریف، کنترل و مدیریت محدوده ارائه کنیم؛ اما یکی از خطرناک‌ترین و نادیدنی‌ترین اشکال خزش محدوده، که بسیار با آن مواجهیم، ارزش ذکر کردن دارد. این نوع خزش محدوده از خودمان حاصل می‌شود؛ فرض کنید هرروز شما و تیم بر پروژه کار می‌کنید که بعد از مدتی درمی‌یابید راه‌های بهتری برای انجام کارها و بهبود اقلام تحویلی وجود دارند که می‌توان آن‌ها را با یک ترفند ساده به کار برد. شاید این کار را با نیت خیر انجام بدهید؛ اما لازم است کسی نظارت داشته باشد وگرنه ممکن است ناگهان امور از کنترل خارج می‌شود. دقت کنید که بودجه، منابع و زمان‌بندی توأم با محدودهٔ پروژهٔ شما رشد می‌کنند و اگر شاهد رشد دائمی آن‌ها نبودید، هر خزش محدوده‌ای را که ممکن است بر بودجهٔ شما اثر بگذارد بررسی کنید.

۷

چه کسی مسئول بودجه است؟ جواب را حدس می‌زنید: همه!

یکی از بهترین راه‌هایی که مدیران ارشد فناوری و رهبران پروژه می‌توانند بودجه‌ها را مدیریت کنند، ایجاد حس مسئولیت پذیری نسبت به بودجه در افراد تیم است. منظور این نیست که مسئولیت را واگذار کنید؛ اسم شما بر سر در پروژه نوشته شده! اما با فهماندن اینکه راهبرد تجاری چیست و چگونه پروژهٔ شما به ایفای آن کمک می‌کند، نظر جمع را به مأموریت همگانی بیشتر جلب می‌کنید. اگر این کار را به‌درستی انجام بدهید خواهید دید که اعضای تیم مواظبت می‌کنند که خرجشان با انتظارات سازمان در خصوص بودجه هم‌خوانی داشته باشد. اگر تیمتان را کاملاً از پیش‌بینی‌های بودجهٔ پروژه مطلع نگه دارید، می‌فهمید که وقتی وضعیت فعلی بودجه را برای آن‌ها به‌روز نگه می‌دارید، آن‌ها مسئولانه‌تر عمل می‌کنند و بر هزینه‌های مربوط به حوزهٔ کاری‌شان در پروژه نظارت می‌کنند. مدیریت مناسب بودجه نیز مثل دیگر جنبه‌های مدیریت پروژه به هدایتگری مناسب، راهبرد هوشمندانه و روش‌های کارا بستگی دارد. ایده‌های گفته‌شده را امتحان کنید، ایده‌های خودتان را نیز به محک تجربه بگذارید. در سال ۲۰۱۷ و سال‌های پیش رو پرتوان باشید؛ برای بودجه‌های بیشتر باید جشن گرفت، نه اینکه از آن‌ها فراری بود!

مقایسهٔ کانبان با اسکرام کدام بهتر است؟

برای مدیریت پروژه باید از کانبان (Kanban) یا اسکرام (Scrum) استفاده کرد؟ سؤال فریبنده‌ای است زیرا بستگی دارد به پروژه، به طبیعت سازمان، و به مسیری که بهترینِ برای تیم شماست؛ بی‌گمان دو روش ارزش‌های خودشان را دارند. درواقع رَوِشگان‌های (متدولوژی) بسیاری وجود دارند که می‌توانیم برای پروژه‌ها به کار بگیریم؛ اما معمولاً دربارهٔ کانبان و اسکرام بحث می‌شود. بنابراین برای این بحث، خوب است که این دو را در یک حلقهٔ مربعی قرار بدهیم تا شاید بهترین روشگان حاصل شود!

کدام در صحنهٔ نبرد مدیریت پروژه پیروز است؟

اسکرام یا کانبان

اسکرام

- تکرارها با جعبه‌های زمانی (Timebox)
- میان‌کارکردی بودن
- وظایف اسپرنیت
- اسکرام‌مستر

کانبان

- رویدادمحوری
- اختیاری بودن تخمین‌زنی‌ها
- اشترک‌گذاری با همه
- بدون نقش معین

PM

در یک گوشه: اسکرام

اسکرام از نیمهٔ دههٔ ۱۹۸۰ به کار گرفته شد و از سال ۲۰۰۱ به یک رَوِشگان فرعیِ (sub-methodology) چابک (Agile) تبدیل شده است. این اتفاق وقتی افتاد که کِن شوابر و مایک بیدِل در این‌باره کتابی را با عنوان «توسعهٔ نرم‌افزار چابک با اسکرام» نوشتند. اسکرام یک چارچوب «فرایندی» است که برای ساختاردهی کارِ به‌خوبی تعریف شده است و معمولاً تیم‌های کوچک‌تر در دوره‌هایی که اسپرینت نامیده می‌شوند از آن استفاده می‌کنند؛ اسپرینت‌ها از وظایف با طول مدتِ کوتاه‌تر تشکیل شده‌اند تا پروژه را نسبت به تغییرات انعطاف‌پذیرتر و انطباق‌پذیرتر کنند. چرخه‌ها (سیکل) تقریباً طول دوهفته‌ای دارند و یک اسکرام‌مستر (Scrum Master) آن را هدایت می‌کند؛ که می‌تواند –نه الزاماً– جایگاه CSM (اسکرام‌مستر گواهی‌دار ScrumAlliance) را داشته باشد. اسکرام‌مستر کارشناس فرایند تیم است و برای هدایت تیم در بهینه‌سازیِ کارکرد اسکرام در پروژه نقش کلیدی دارد. این مهم با جلسات حاصل می‌شود؛ جلسات اسکرام، روزانه تشکیل می‌شوند و اسکرام‌مستر تقریباً مثل یک معلم در روند پروژه از تیم پشتیبانی می‌کند و به آن‌ها جهت می‌دهد. هرچند اسکرام به گونه‌ای است که برای سازمان‌ها و پروژه‌های بزرگ‌تر مقیاس‌پذیر و تعمیم‌پذیر است، ریشهٔ آن در توسعهٔ نرم‌افزار چابک (Agile) است و در چنین محیط‌های چابک‌تر و کوچک‌تری کارکرد منسجمی دارد. اما این ویژگی باعث نشده که کاروران (به کارگیرندگان) در صنعت‌های مختلف، از تدارکات و خرده فروشی گرفته تا برنامه‌ریزی رویدادها، دست از اسکرام بردارند.

کانبان در گوشهٔ دیگر مربع

زادگاهِ کانبان کارخانه‌های شرکت خودروسازی تویوتا در ژاپن است. کانبان یک شیوهٔ تصویری برای مدیریت گردش کار است. بی‌دلیل نیست که اسم آن به ژاپنی به‌معنای آگهی‌نما (بیلبورد) است؛ زیرا در این فرایند وظایف بر تابلوِ (بورد) کانبان قرار می‌گیرند. بدین‌ترتیب پیشروی پروژه به‌طور شفاف و به‌سادگی در تمام امور ترسیم می‌شود.

کانبان که نسبت به اسکرام ساختاریافتگی کمتری دارد، اصلاً یک چارچوب فرایندمحور نیست؛ مدل تابلوِ کانبان (کانبان بورد) و کارت‌های رویِ آن جنبه‌های پروژه را به‌صورت تصویری نشان می‌دهند. کار در تابلوِ کانبان به‌صورت ستون‌های عرضه‌شده (released)، آماده برای عرضه (ready for release) و آزمون (testing) سازمان‌دهی می‌شود. از آن برای مدیریت پیشروی کار استفاده می‌شود. یکی از فوائد استفاده از این روش آسان بودن شناسایی ناکارآمدی‌ها در زمان‌بندی پروژه و رسیدگی به آن‌ها قبل از ایجاد در کود است. این مزایای کانبان به دلیل برنامه‌های کاربردی (اپ‌های) تصویری آن برای برنامه‌ریزی است که می‌توانند به برنامه‌ریزی بک‌لاگ و داستان‌های کاربری مصور (استوری‌بورد) در آن کمک کند.

کانبان، مانند اسکرام، در تیم‌های چابک (Agile) به کار برده می‌شود؛ اما این روش برای تعداد زیادی از برنامه‌های کاربردی قابل استفاده است. درواقع تقریباً هر سازمان یا فردی می‌تواند از تابلوِ کانبان بهره ببرد. در هر فاز پروژه، کارت‌ها می‌توانند موعد تحویل پروژه، اعضای تیم، و حتی ایده‌ها یا هر چیز ممکن را نمایش بدهند.

آیا آمادهٔ رقابتید؟

این دو روش رقابت جالبی دارند؛ زیرا شباهت‌های بسیاری بین اسکرام و کانبان وجود دارد که درنهایت آن‌ها را باهم منطبق می‌سازد. از شباهت‌های این دو این است که هم برای پروژه‌های چابک و هم برای ناب (لین) (Lean) ایدئال هستند، کار را در فرایند محدود می‌کنند و بیشتر طرفدار گردش‌های پیوسته زمان‌بندی هستند تا اینکه یک زمان‌بندی را بپذیرند و اجرا کنند.

هردو با تمرکز بر تحویل سریع و متعدد، در یک فرهنگ شفاف برای کمک به بهبود فرایند کار می‌کنند و پروژه را به بخش‌های کوچک‌تر و قابل‌مدیریت‌تر تقسیم می‌کنند. تیم‌های خودسازمان‌ده، که دائماً برمبنای داده‌های پروژه کار را بهینه‌سازی می‌کنند، پذیرای این دو روش هستند.

چرا اسکرام بهتر است؟

اسکرام ساختاریافته است. اگر بر پروژه‌ای کار می‌کنید که به روندها و نقش‌های معین‌تری نیاز دارد، اسکرام برای آن پروژه مناسب است. اسکرام به‌دلیل چنین چارچوبی مزایایی دارد.

به‌طورمثال در اسکرام شفافیت و وضوح بیشتری نسبت‌به کانبان و دیگر روشگان‌ها (متدولوژی‌ها) وجود دارد. وقتی که می‌خواهید همه را باهم هماهنگ کنید می‌توانید مثل یک تیم کاملاً یکپارچه عمل کنید.

مزیت دیگر اسکرام افزایش مسئولیت‌پذیری تیم‌های اسکرام است. به دلیل اینکه سریعاً عمل می‌کنید، جلسات شما حداقل روزانه است. این ویژگی طبیعتاً به شفافیت پروژه کمک می‌کند اما علاوه‌برآن موجب مسئولیت‌پذیری افراد تیم نیز می‌شود. به‌این‌ترتیب می‌توانید به کسانی که عملکرد خوبی دارند پاداش بدهید و برعکس. تغییر معمولاً در پروژه‌های بزرگ مشکل‌ساز است؛ چراکه مثل کشتی‌های بزرگ رفت‌وبرگشت آن زمان زیادی را می‌گیرد. اما با اسکرام چنین نیست. این فرایند چابک‌تر طراحی شده تا بتواند هدایت را به‌عهده بگیرد. درواقع تغییر خواه است و تغییرات را آسان می‌کند. اسکرام به‌دلیل ساختار و سرعت و متناسب بودنش با تیم‌های کوچک حافظ پول آن‌هاست. این برای کسب‌وکارهای نوپا (startup) و دیگر مؤسسه‌هایی که حد سود (bottom line) را به‌دقت بررسی می‌کنند (واقعاً کجا بررسی نمی‌کنند؟!) مزیت بزرگی است.

معایب اسکرام

اسکرام در مبارزهٔ خود قدرتش را اثبات کرده است اما شکاف‌هایی هستند که می‌توانند موجب آسیب‌هایی شوند، اول‌ازهمه اسکرام‌مستر. باید تجربهٔ زیاد و سطح‌بالایی داشته باشید که اسکرام را به پیشروی وادارید. درغیر این‌صورت با خطر خزش محدوده (Scope creep) روبه‌رو می‌شوید.

نه‌تنها اسکرام‌مستر باید درحد عنوانش مهارت داشته باشد بلکه تیم نیز به سطحی از تجربه نیاز دارد. اگر اسکرام‌مستر یا تیم شما در سطح قابل‌قبولی نباشند با مشکلاتی روبه‌رو خواهید شد. همچنین اگر وظایف به‌خوبی تعریف نشده باشند، امکان بی‌دقتی‌هایی وجود دارد. البته این در هر پروژه‌ای صادق است اما در محیط اسکرام سریع‌تر و شدیدتر خودش را نشان می‌دهد.

چه‌موقع استفاده از کانبان بهتر است

شاید اسکرام روش مناسبی باشد اما کانبان نیز شایستهٔ توجه است. یک مزیتش این است که به تجربه و آموزش موردنیاز اسکرام نیاز ندارد. آموزش و شناخت آن آسان است. این یک امتیاز مثبت برای آن محسوب می‌شود؛ می‌توانید با انرژی دست‌به‌کار شوید.

کانبان برای بهبود گردش کار و کمینه کردن چرخهٔ زمانی عالی است اما انعطاف‌پذیری محصول را نیز افزایش می‌دهد. اما اگر در جست‌وجوی رَوِشگانی (متدولوژی) هستید که با بادهای تغییر خم می‌شود و نمی‌شکند، کانبان مناسب شماست.

با کانبان در فرایند اجرای پروژه‌تان اتلاف کمتری خواهید داشت، زیرا فرایند را زودتر مشاهده می‌کنید و می‌توانید هرچه را که نیاز ندارید کنار بگذارید. همچنین به جریان تحویل نیز کمک می‌کند، پس می‌توانید سریع‌تر از پروژه نتیجه بگیرید.

کانبان این کار را با کم کردن چرخه‌های زمانی پروژه انجام می‌دهد که مجدداً بر کارآمدی گردش کار تأثیر دارد. کانبان را هرکسی در هر پروژه‌ای می‌تواند به کار بگیرد، هم اعضای تیم‌های عمومی و هم اعضای ماهرتر می‌توانند تمام کار را بر یک تابلوِ کانبان یکسان دنبال کنند. با تابلوهای کانبان به هیچ جعبهٔ زمانی (تایم باکسی) نیاز نیست، بنابراین هیچ اسپرینتی نداریم. با پیشروی در پروژه لازم نیست که تابلوِ کانبان را بازآرایی (reset) کنید. تا جایی که پروژه نیاز دارد ادامه می‌یابد.

مشکلات کانبان

کانبان ساده است، بنابراین مشکلاتی که رخ می‌دهد معمولاً به‌دلیل استفادهٔ نادرست از تابلوِ کانبان است. مثلاً یک تابلوِ کانبانِ به‌روزنشده در فرایند توسعهٔ شما مشکلاتی را ایجاد می‌کند. مسئلهٔ دیگر وقتی است که تیم‌ها تابلو را زیاده پیچیده می‌کنند. شهرت کانبان به سادگی آن است، پس با پیچیده کردن پروژه، خلافِ منطقِ کار با کانبان عمل کرده‌اید و درنتیجه پروژهٔ شما آسیب می‌بیند. مزیت نداشتن چارچوب زمانی می‌تواند به یک عیب تبدیل شود. بدون یک موعد تحویل سفت‌وسخت بعید نیست که یک وظیفه یا یک فاز به رکود برسد و زمانی بیش از آنچه نیاز دارد بگیرد.

برنده کیست؟

مشخصاً برنده‌ای وجود ندارد. این رقابت واقعاً مورد تا مورد تفاوت دارد. حتی مواردی هست که پتانسیل استفاده از هردو رَوِشگان برای پروژه‌های متفاوت و جنبه‌های مختلف پروژه مناسب است. هردوی این روشگان مدافع بهبود پیوسته هستند و ازاین‌رو هردو می‌توانند برای مدیریت هر نوع کاری مناسب باشند. به شما بستگی دارد که تصمیم بگیرید که کدام برای نیازهای تیم و پروژهٔ شما مناسب است. خوب است با سلیقه‌های شخصی‌مان بی‌انصافی نکنیم و پروژه را صادقانه و با شفافیت به‌پیش ببریم!

چگونه با MS Project یک محیط چندپروژه‌ای را نجات بدهیم؟

بهترین نرم‌افزار برنامه‌ریزی پروژه برای مدیریت هم‌زمان چندین پروژه چیست؟

این سؤالی است که مدیران سطح C، مدیران پروژه و مدیران اجرائی تصمیم‌گیرنده را، که در جست‌وجوی بهترین راهکار مدیریت پروژه برای نظارت بر پروژه‌های خود هستند، مشغول خود می‌کند. Microsoft Project پرکاربردترین برنامهٔ کاربردی مدیریت پروژه است؛ اما آیا برای مدیریت چندپروژه‌ای کفایت می‌کند؟ شرکت GetApp Lab که در زمینهٔ تحقیقات و گرایش‌های اخیر دربارهٔ ابزار نرم‌افزاری کسب‌وکار کار می‌کند، گزارش می‌کند که ۶۷درصد از پاسخ‌دهنده‌ها به مصاحبهٔ سال ۲۰۱۶، برای برنامه‌ریزی پروژه‌ها و نیازهای مدیریتی‌شان از MS Project استفاده می‌کنند.

مصاحبهٔ PricewaterhouseCooper با عنوانِ «بلوغ مدیریت پروژه» نشان می‌دهد که فقط ۲٫۵درصد از شرکت‌ها ۱۰۰درصدِ پروژه‌هایشان را با موفقیت تکمیل می‌کنند. مؤسسهٔ مدیریت پروژه در گزارشی تخمین می‌زند که برای هر یک‌میلیارد دلاری که در سال ۲۰۱۶ در ایالات متحدهٔ آمریکا سرمایه‌گذاری می‌شود، ۱۲۲ میلیون دلار به دلیل اجرای ضعیف پروژه هدر می‌شود. با درنظرگرفتن آخرین آمار مربوط به تأخیرهای مکرر پروژه و افزایش هزینه‌ها، به‌نظر می‌رسد که مدیریت پروژه در مرز شکست است. آیا کمبود ابزار مناسب مدیریت پروژه دلیل خروجی‌های اسف‌بار ذکرشده است؟ تیم مدیریت پروژهٔ Epicflow با سی‌سال تجربهٔ ترکیبی در مدیریت پروژه، بررسی کرده است که درواقع ویژگی‌های MS Project نادیده گرفته شده‌اند. این تیم راهکاری برای ۲۰۱۷ عرضه کرده است.

چرا MS Project برای مدیریت چندپروژه‌ای کافی نیست

Epicflow وقتی ویژگی‌های نرم‌افزاری‌ای را که باید در سال ۲۰۱۷ برای برطرف‌ساختن نیازهای فعلی مدیران پروژه توسعه بیابند بررسی می‌کرد، دریافت که تقریباً تمام ابزار مدیریت پروژه برای محیط‌های تک‌پروژه‌ای طراحی شده‌اند. تقریباً هر راهکار مدیریت پروژه‌ای از جمله MS Project، Jira، Treimergo، TOPdesk، و Primavera برمبنای یک فلسفهٔ پروژه‌ای واحدی ساخته شده‌اند و به برنامه‌ریزی و بازبرنامه‌ریزیِ (replanning) دستی نیاز دارند. این برنامه‌های کاربردی هرچند که همکاری‌های مستمر را ترغیب می‌کنند، فاقد ویژگی‌های نرم‌افزاری برای تعیین اولویت‌ها، ایجاد بارکارهای تلاش‌محور مکفی، و نهایتاً قابلیت جلوگیری از اضافه‌بار و سربارِ برنامه‌ریزی در محیط‌های چندپروژه‌ای هستند. پس مدیران پروژه چگونه می‌توانند نیاز فعلی بازبرنامه‌نویسی را برطرف کنند؟ چه ابزار دیجیتالی برای پیگیری عملکرد پروژه مطلوب مدیران است؟ چگونه می‌توان MS Project را با این محیط‌های چندپروژه‌ای هماهنگ کرد؟

ابزار را تغییر ندهید، فلسفه‌ای را که پشت آن نهفته است تغییر بدهید

استفاده از MS Project در محیط‌های چندپروژه‌ای طاقت‌فراست، زیرا فاقد تحلیل‌های موردنیاز برای تخصیص مؤثر منابع است. اما چه خوب بود اگر می‌توانستید دیدی از اینکه فرایندها کجا گیر کرده‌اند داشته باشید و تنگناهای منابع را در کانال داده‌های MS Project پیش‌بینی کنید!

آلبرت پونستین و ویلِم ترومپ، دانشمندان هلندی، این کار را با تلفیق الگوریتم هوشمند و راهکارهای مبتنی بر وب و متناسب با محیط‌های چندپروژه‌ای، برای بسیاری از حوزه‌ها، مانند حوزه‌هایی مثل مهندسی، ساخت‌وساز، توسعهٔ نرم‌افزار و غیره ممکن ساخته‌اند. این متخصصان خبرهٔ مدیریت پروژه، با ترکیب سی‌سال تجربهٔ ۴۰۰۰ کارشناس کسب‌وکار، مدیرعامل و مدیرپروژه به این نتیجه رسیدند که برای تعیین اولویت‌ها، کاهش اضافه‌بار، و برنامه‌ریزی سربار راهکاری وجود دارد.

مدیریت چندپروژه‌ای از چشم‌انداز دسترسی به منابع

پونستین و ترومپ معتقدند که آنچه واقعاً موجب سرسام مدیران پروژه می‌شود، نیاز به جریان‌بخشی، ترتیب‌دهی و سادگی در مدیریت تعداد زیادی از پروژه‌ها، حل مشکلات ظرفیت، تعیین اولویت‌ها و تخصیص مناسب منابع است. یکی از اهداف Epicflow این است که با رویکرد مدیریت پروژه از لحاظ دسترسی‌پذیری منابع و توزیع وظایف، مانع هدر رفتن زمان و انرژی صرف‌شده برای برنامه‌ریزی اضافی شوند.

در مدیریت چندپروژه‌ای، سرعت و ماندگاری و انعطاف‌پذیری را انتخاب کنید

آیا راهکار قابل‌اعتمادی در مدیریت پروژه وجود دارد که بتواند به‌سادگی یکپارچه شود و با MS Project به کار گرفته شود؟ بله، وجود دارد. اکنون امکان ارتقای MS Project برای محیط‌های چندپروژه‌ای وجود دارد و با افزودن امکان تجزیه‌وتحلیل به برنامه‌های کاربردی ایستای مدیریت پروژه، می‌توان بارِ برنامه‌ریزی را از شانه‌های مدیران پروژه برمی‌داشت.

افزونهٔ نرم‌افزاری Epicflow یک بسته از ویژگی‌های مفید مدیریت پروژه عرضه می‌کند و داده‌های MS Project و تقریباً هر ابزار مدیریت پروژهٔ مورداستفاده مدیران پروژه را یکپارچه می‌کند.

Epicflow به این منظور طراحی شد که با جمع‌آوری و تحلیل به‌روزرسانی‌های گردش‌کار و تصویرسازی‌های این تجزیه‌وتحلیل‌ها به‌صورت چند نموداری که شفافیت کاملی از منابع، وظایف و پروژه‌ها فراهم می‌کند، مانع تنگناها شود. این افزونه نظم‌وترتیب و سادگی را برای محیط‌های پویای کاری به‌همراه می‌آورد و تمرکز را افزایش می‌دهد.

تیم Epicflow و عرضهٔ SaaS (نرم‌افزار به‌عنوان یک سرویس)

بااینکه نگرش مبتنی بر وب Epicflow اخیراً مطرح شده است، پیشاپیش کارآیی خود را اثبات کرده است؛ بسیاری از شرکت‌ها این نرم‌افزار مدیریت پروژه را کلید حل مشکلات اصلی مدیریت پروژهٔ خود می‌دانند. نسخهٔ اولیهٔ Epicflow، Flow MPM، یک برنامهٔ کاربردی آفلاین بود که به بسیاری از کسب‌وکارها در کاهش زمان عرضه، رسیدن به تاریخ تحویل و بهبود کیفیت خروجی‌های تجاری کمک می‌کرد. Flow MPM برای کارخانهٔ کابل TKF که کابل‌های اقیانوسی تولید می‌کند، کارآمدی خود را اثبات کرد. هانز دِبوئر، مدیر R&D در TKF اذعان می‌کند که با Flow MPM طی دوسال، خروجی شرکت تا ۲۰۰درصد افزایش و زمان عرضه (lead time) تا ۵۰درصد کاهش یافت و توأمان تعهدات واقع‌بینانه‌ای به مشتریان داده شد. این برنامهٔ کاربردی مدیریت پروژه که پیشگام Epicflow است، با موفقیت توانست به Trimergo متصل شود تا بتواند ابرکشتی‌های تفریحی‌ای مانند ونوس را برای استیوجابز بسازد. نسخهٔ مبتنی بر وب آن، Epicflow، انعطاف‌پذیرتر است. Epicflow، همراه با MS Project، قدرت فوق‌العاده‌ای را برای موفقیت در محیط‌های چندپروژه‌ای فراهم می‌کند و به گردش کارهای کافی و بی‌نقص و حل تضاد و تداخل منابع کمک می‌کند. Epicflow نه‌فقط یک محیط دوستانه برای تعاملات برد-برد ایجاد می‌کند، بلکه نشان می‌دهد که کجا شریان‌های پروژه شما دچار گرفتگی شده‌اند. از اکنون به بعد، MS Project با برخورداری از مهارت فوق‌العادهٔ تحلیلی جدید Epicflow بازسازی می‌شود. پس با راهکاری که در دست داریم، پیشروی مدیران پروژه در محیط‌های چندپروژه‌ای متوقف نمی‌شود.

برگزاری دوره‌های کارآموزی مدیریت پروژه در بریتانیا

■ برگزارکننده: Parallel و Interserve

انتخاب مدیریت پروژه به‌عنوان شغل اول هر روز رواج بیشتری می‌یابد. این تغییر مهمی است زیرا مدیریت سنتی پروژه یک شغل ثانویه محسوب می‌شد. گزارش شرکت بریتانیایی مدیریت پروژهٔ Arras People مؤکد همین موضوع است. در این مصاحبه از مدیران پروژه پرسیدند که آیا مدیریت پروژه حرفهٔ اول یا دوم آن‌هاست. نتایج هیجان‌انگیز است؛ بیشتر افراد هنوز مدیریت پروژه را شغل دوم خود می‌دانند. بااین‌حال در ادامه مشاهده می‌کنید که معمولاً افراد جوان‌تر مدیریت پروژه را شغل اول خود محسوب می‌کنند.

نتیجه این است که اگر افراد پاسخ‌دهندهٔ بریتانیایی را برحسب سن درنظر بگیریم، مثل شکل زیر، مشاهده می‌کنیم که درصد بیشتر افرادی که مدیریت پروژه را شغل اول خود می‌دانند، زیر ۴۲ سال دارند. اما اگر تعداد کلی را در نظر بگیریم، محاسبهٔ آن‌ها از ۴۳ درصد از پاسخ‌دهنده‌ها کمتر می‌شود.

مشاهده می‌کنیم که افراد زیر ۳۴ سال طیف وسیع‌تری دارند و تعداد واقعی آن‌ها بیشتر است؛ ۵۴ درصد از پاسخ‌دهنده‌ها را تشکیل می‌دهد. بنابراین شاید بتوان گفت که انتخاب مدیریت پروژه به‌عنوان شغل اول رو به فزونی است.

در این گزارش این پاسخ‌ها مربوط به اولین دفعه‌ای بود که از آن‌ها پرسیده شد و تفاوت پاسخ‌ها در چند سال بعد نیز جالب است.

انتظار می‌رود این روند در بریتانیا با برنامه‌های آموزشی جمعیت مدیران پروژه بریتانیا (APM) مانند «دورهٔ کارآموزی پیشگامان مدیریت پروژه» شتاب سریع‌تری بگیرد. این دوره یک استاندارد معیار برای تمام کارآموزان مدیریت پروژه تعیین می‌کند و توسعهٔ دانش و اتکا به مهارت‌ها و رفتارهای موردانتظار مدیران پروژه در دستور کار این دوره است.

دو شرکت مدیریت پروژه بریتانیایی Parallel Project Training و Interserve Learning and Employment باهم ترکیب شدند تا یک دورهٔ کارآموزی ساختاریافتهٔ مدیریت پروژه عرضه کنند. این برنامه برای تازه‌کارها ۲۴ماهه و برای کسانی که مهارت‌های قبلی دارند ۱۸ماهه است و شامل سه بخش اصلی است: دانش اصول مدیریت پروژه، کارگاه‌های توسعهٔ مهارت‌ها، و کاربرد عملی مهارت‌ها در محل کار.

دانش اصول مدیریت پروژه

دانش مدیریت پروژه با یک امتحان نهایی در ۱۲ ماه اول سال ارزیابی می‌شود. این امتحان با تکمیل مدرک مدیریت پروژهٔ انجمن مدیریت پروژه بریتانیا (APM PMQ) صورت می‌گیرد که شامل یک امتحان سه‌ساعته است که شرکت‌کنندگان باید به ۱۰ سؤال انتخابی از ۱۶ سؤال پاسخ بدهند. آمادگی برای این امتحان طبق نظام آموزشی Parallel صورت می‌گیرد که متشکل از کارگاه‌ها، آموزش مجازی، پادپخش‌ها (پادکست) و وبینارهای متخصصان مدیریت پروژه است. از کارآموزان انتظار می‌رود که سؤال‌های تکلیف‌های ماهانه و امتحان آمادگی را قبل از امتحان نهایی انجام بدهند.

کارگاه‌های توسعهٔ رفتاری و مهارت‌ها

علاوه بر تئوری، کارآموزان می‌آموزند که چگونه در کارگاه‌ها از ابزار و فنون مدیریت پروژه استفاده کنند. برحسب پروژهٔ شبیه‌سازی‌شده، این کارگاه‌ها مهارت‌های عملی مدیریت پروژه را تقویت می‌کنند، مانند مدیریت ذی‌نفعان، برنامه‌ریزی، تخصیص بودجه، کنترل تغییر، گزارش‌دهی و مدیریت ریسک.

به عمل در آوردن تئوری‌ها

آخرین بخش این کارآموزی، به‌کارگیری مهارت‌های توسعه‌یافته در کارگاه‌ها و بخش‌های تئوری دوره در یک پروژهٔ کاری است. این بخش را یک مربی کارآموزی پشتیبانی می‌کند اما برای دستیابی کارآموز به یک پروژهٔ مناسب به پشتیبانی کارفرما نیاز است.

ارزیابی مرحلهٔ پایانی

وقتی کارآموز تمام این بخش‌ها را تکمیل کرد، می‌تواند سبد پروژه (پورتفولیو) خود را برای ارزیابی نهایی تحویل بدهد؛ که نشان می‌دهد چه آموخته است و همچنین یک مصاحبه‌ای هست که باید در آن اثبات کند که به استانداردهای تعیین‌شدهٔ APM رسیده است.

هوش‌تجاری و کاربردهای آن

در مقالهٔ شمارهٔ قبلی مجله، با عنوان «گزارش تحلیلی»، گفتیم شرکت‌هایی موفق هستند که نیازهای خود را بهتر و سریع‌تر تشخیص بدهند و درنتیجه سؤال‌های بهتری بپرسند و جواب‌های دقیقی بیابند. گفتیم که چنین قابلیتی به هوش تجاری نیاز دارد.

هوش تجاری (BI) چیست؟

هوش تجاری به‌منظور تبدیل داده‌ها به یک «هوش کاربردی» از خدمات و نرم‌افزارها استفاده می‌کند؛ این هوش کاربردی تصمیمات تجاری تاکتیکی و راهبردی سازمان را مشخص می‌کند. با ابزار BI امکان دسترسی به مجموعه‌داده‌ها و تحلیل آن‌ها فراهم می‌شود و گزارش‌ها و خلاصه‌ها و داشبوردها و گراف‌ها و نمودارها و نقشه‌ها و تحلیل‌هایی ارائه می‌شود که در اختیار کاربران قرار می‌گیرد؛ این کاربران دربارهٔ وضعیت تجاری، هوش جزء‌گرا دارند.

هوش تجاری (BI) با تحلیل تجاری (BA) چه تفاوتی دارد؟

«هوش تجاری» تحلیل‌های توصیفی نیز نامیده می‌شود؛ زیرا وضعیت فعلی و قبلی را توصیف می‌کند. مایکل اِف گورمن، پروفسور مدیریت عملیات و دانش تصمیم‌گیری در دانشگاه دِیتونِ اُهیو، معتقد است که «[هوش تجاری] به شما نمی‌گوید چه کنید؛ به شما می‌گوید چه بوده و چه هست.» این توضیح هوش تجاری را با تعریف تحلیل تجاری (BA) مقایسه کنید: یک فرایند فناوری‌محوری که نرم‌افزار با آن داده‌ها را به‌منظورِ پیش‌بینی آنچه رخ می‌دهد (تحلیل‌های پیش‌بینی‌کننده) یا آنچه با یک رویکرد معین ممکن است رخ بدهد (تحلیل‌های تجویزی) تحلیل می‌کند. تحلیل تجاری را گاهی تحلیل‌های پیشرفته می‌نامیم.

اما کارکرد هوش تجاری چگونه است؟

هوش تجاری به کاربران تجاری نمی‌گوید که اگر عمل معینی را انجام بدهند، چه اتفاقی می‌افتد و باید چه کنند. به‌علاوه هوش تجاری فقط به معنای تولید گزارش نیست؛ بلکه هوش تجاری به افراد راهی برای بررسی داده‌ها و گرایش‌ها و بینش‌یابی می‌دهد. کریس هاگانز، نائب‌رئیس عملیات شرکت مشاورهٔ WCI، می‌گوید: «بسیاری از افراد در کسب‌وکار برای بهتر انجام‌دادنِ کارشان به داده‌ها نیاز دارند.» هاگانز خاطرنشان می‌کند که ابزار هوش تجاری به تلاش‌های افراد برای جست‌وجو و ادغام و پرسمان داده‌ها و دستیابی به اطلاعات لازم برای اخذ تصمیمات تجاریِ صحیح کمک می‌کند. هاگانز اظهار می‌کند که به‌طور مثال اگر شرکتی بخواهد زنجیرهٔ تأمینش را بهتر مدیریت کند، نیاز دارد که قابلیت‌های هوش تجاری‌اش را برای تعیین اینکه تأخیرها کجا رخ می‌دهند و موارد تغییرپذیر درون فرایند حمل‌ونقل کجا هستند تقویت کند. این شرکت همچنین می‌تواند قابلیت‌های هوش تجاری خود را برای کشف محصولاتی که بیشتر از همه تأخیر می‌یابند یا شیوه‌های نقل‌وانتقالی‌ای که بیشتر از همه در تأخیرها دخیل بوده‌اند به کار بگیرد. در اینجاست که متوجه ضرورت هوش تجاری در کسب‌وکارها می‌شویم. شما چگونه هوش تجاری را در کسب‌وکار خود به‌کار می‌گیرید؟ تیم مجلهٔ تخصصی خط دید، مشتاق تحلیل‌های کاربردی و برخاسته از تجربه‌های شماست.

نگاهی به موقعیت‌های شغلی فناوری اطلاعات در سال 2017

ناحیهٔ پاسیفیک، کالیفرنیا

منطقهٔ پاسیفیک که منزلگاه درهٔ سیلیکون و دیگر مناطق سرشناس فناوری است، بزرگ‌ترین و رقابتی‌ترین بازار کار فناوری ایالات متحده است. همین باعث می‌شود که این منطقه مزایا و معایبی داشته باشد؛ همه دنبال بهترین و مبتکرترین استعدادها هستند. درواقع استعدادهای زیادی هستند که باید با آن‌ها رقابت کرد؛ چراکه خوره‌های فناوری در این منطقه، که قرارگاه بزرگ‌ترین نام‌ها در فناوری‌های رایانه‌ای مانند فیس‌بوک، گوگل، اپل، مایکروسافت و آمازون است، دسته‌دسته جمع می‌شوند. گروه راهبردی توسعه و تحقیقات اقتصادیِ پراکسیس (Praxis) با تحلیل داده‌های ده‌سالهٔ ارزش شغلی برای مجلهٔ اقتصادی فوربز، سه منطقه از مناطق کلان‌شهری پاسیفیک را در بالای لیست ۱۰ منطقهٔ اول ایجادکنندهٔ شغل‌های مربوط به فناوری در سال ۲۰۱۷ قرار داد. سان‌فرانسیسکو، اوکلند، هی‌وُردِ کالیفرنیا، با بیش از ۲۲۰۰۰۰ شغل مربوط به فناوری در ۲۰۱۶ و ۹۰ درصد رشد صنعتی فناوری‌محور از سال ۲۰۰۶ تا ۲۰۱۶ در بالای این لیست قرار دارند. مناطق کلان‌شهر سان‌خوزه، سانی‌ویل و سانتاکلارای کالیفرنیا (که به درهٔ سیلیکون شناخته می‌شوند) با ۱۷۶۰۰۰ شغل مربوط به فناوری در سال ۲۰۱۶ و ۷۹٫۶ درصد رشد صنعتی فناوری‌محور از سال ۲۰۰۶-۲۰۱۶ در مکان چهارم قرار گرفتند. و ناحیهٔ مناطق سیاتل، تاکوما و بلویو واشینگتون با حدود ۱۴۶۰۰۰ شغل مربوط به فناوری در سال ۲۰۱۶ و ۴۷٫۷ درصد رشد صنعتی در این دورهٔ ده ساله، به مقام هشتم رسید.

پاتریک سیرکلی، یک مدیر استخدام در ناحیهٔ سان‌فرانسیسکو در آژانس Mondo، معتقد است که ویژگی‌های منحصربه‌فرد صحنهٔ فناوری ساحل پاسیفیک وضعیت استخدام نیرو را در این ناحیه نسبت به دیگر ناحیه‌ها متفاوت ساخته است. او معتقد است که بسیاری از شرکت‌های ساحل غربی در جست‌وجوی فن‌آورانی هستند که پرشور و ابتکاری و خلاق باشند تا کسانی که فقط هم‌خوانی فنی و فرهنگی برای جایگاه‌های شغلی معین دارند. او می‌گوید: «این حرفه‌ای‌های مبتکر و ماهر حوزهٔ فناوری هستند که انتخاب می‌کنند کجا بروند.»

درنتیجه سیرکلی معتقد است که شرکت‌ها غالباً مجبورند که به دنبال متقاضیان منفعل باشند؛ یعنی کارکنان استخدام‌شده‌ای که در جست‌وجوی جایگاه‌های شغلی جدید نیستند اما مایلند که این شغل‌های جدید را در نظر داشته باشند. سیرکلی می‌گوید که با توجه به رقابتی که برای استعدادها وجود دارد، بسیاری از کارکنان مایل‌اند که متقاضیانی را جلب کنند که چندان ایدئال نباشند؛ زیرا دسترسی به این متقاضیان بیشتر است؛ اما برخی از شرکت‌ها امکان صبر کردن برای یافتن کارمند مناسب را دارند و گاهی واقعاً ماه‌ها صبر می‌کنند. برای مهندسان نرم‌افزار و توسعه‌دهندگان وب همیشه تقاضا هست؛ همین‌طور برای متخصصان امنیت و دانشمندان داده‌ها.

استیفن فلیسان، مدیرعامل Edmunds، که یک پلتفرم خرید و اطلاعات خودکار در سانتامونیکای کالیفرنیاست، اظهار می‌دارد که در جست‌وجوی کارکنانی است که از مجموعه‌ای از مهارت‌های فنی و نیز تجاری برخوردار باشند. او می‌افزاید که به‌ویژه رقابت تنگاتنگی برای استعدادهای DevOp (توسعه و عملیات) وجود دارد. فلیسان می‌گوید که استخدام آسان نیست، اما به‌دفعات شاهد فرم‌های تقاضایی از دیگر مناطق بوده است که درخواست نقل‌مکان به ایالت‌های پاسیفیک را دارند، و همین به گسترش خزانهٔ استعدادها کمک می‌کند. او می‌گوید: «این متقاضیان برای آب‌وهوا به اینجا می‌آیند؛ به آن‌ها خوش می‌گذرد و انرژی زیادی در اینجاست.»

مقایسهٔ حقوق آی‌تی در سطح ملی با ناحیهٔ پاسیفیک

متوسط حقوق و مزایای کل در ۲۰۱۷ (حقوق پایه + مزایا)

منبع: مصاحبهٔ Computerworld دربارهٔ حقوق شغل‌های آی‌تی در سال ۲۰۱۷

عنوان شغلی	حقوق و مزایا در سطح ملی	حقوق و مزایا در پاسیفیک
مدیر ارشد اطلاعات (CIO) – نائب‌رئیس آی‌تی	۱۷۹۶۱۲ دلار	۱۹۰۰۵۳ دلار
مدیر ارشد امنیت شبکه	۱۷۳۳۰۷ دلار	۱۶۱۰۰۰ دلار*
مدیر آی‌تی	۱۲۷۲۴۸ دلار	۱۴۳۴۳۱ دلار
مدیر توسعهٔ برنامه‌های کاربردی	۱۲۵۳۹۹ دلار	۱۲۳۹۹۰ دلار
مدیر امنیت اطلاعات	۱۱۹۰۷۵ دلار	۱۴۲۲۸۶ دلار
مدیر آی‌تی	۹۹۰۵۳ دلار	۱۰۵۵۳۳ دلار
توسعه‌دهندهٔ برنامه‌های کاربردی	۹۷۱۳۵ دلار	۱۰۶۳۰۶ دلار
تحلیلگر تجاری	۹۲۹۹۳ دلار	۱۱۲۷۰۰ دلار
مدیر پایگاه‌داده	۱۰۴۸۶۰ دلار	۱۲۳۶۴۷ دلار
مدل‌ساز/توسعه‌دهنده/معمار پایگاه‌داده	۱۱۶۹۸۷دلار	۱۱۵۰۵۷ دلار
معمار سازمانی	۱۳۳۸۳۰ دلار	۱۳۲۲۳۶ دلار
متخصص پشتیبانی فنی–میز کمک (Help Desk)	۵۱۴۶۹ دلار	۶۲۶۷۶ دلار
متخصص امنیت اطلاعات	۱۰۰۰۳۲ دلار	۱۰۵۶۷۷ دلار
مدیر شبکه	۷۱۸۰۲ دلار	۷۷۷۸۲ دلار
مهندس شبکه	۹۵۵۵۶ دلار	۱۲۵۸۷۵ دلار*
تحلیلگر–برنامه‌نویس	۸۹۸۵۴ دلار	۹۲۰۳۱ دلار*
رهبر پروژه	۱۰۱۲۷۴ دلار	۱۰۴۶۶۷ دلار*
تحلیلگر ارشد سیستم‌ها	۹۹۰۵۱ دلار	۱۰۲۴۴۲ دلار
مهندس نرم‌افزار	۱۰۸۰۵۹ دلار	۱۲۹۹۹۵ دلار
مدیر سیستم‌ها	۷۵۵۵۳ دلار	۷۴۴۹۵ دلار
تحلیلگر سیستم‌ها	۷۴۷۷۲ دلار	۷۸۷۵۶ دلار
معمار سیستم‌ها	۱۱۷۱۹۰ دلار	۱۲۹۱۲۴ دلار
کاردان (تکنسین)	۵۳۶۱۴ دلار	۵۹۵۵۳ دلار

* کم‌تر از ۱۰ پاسخ (در مصاحبه)

بهترین ناحیه‌های پاسیفیک برای کار در حوزهٔ آی‌تی در ۲۰۱۷

چهارده کارفرما از ناحیهٔ پاسفیک در لیست ۱۰۰ محل برتر برای شغل‌های آی‌تی در سال ۲۰۱۷ نام برده شده‌اند. نام این شرکت‌ها عبارت‌اند از:

- **Applied Materials در سانتاکلارا، کالیفرنیا**
- **Avanade در سیاتل**
- **Cedars-Sinai Health System در لس‌آنجلس**
- **Dignity Health در سان‌فرانسیسکو**
- **Edmunds.com در سانتامونیکا، کالیفرنیا**
- **Genentech در سان‌فرانسیسکوی جنوبی، کالیفرنیا**
- **Illumina در سان‌دیگو**
- **Informatica در ردوود سیتی، کالیفرنیا**
- **Jet Propulsion Laboratory در پاسادنا، کالیفرنیا**
- **Kaiser Permanente در اوکلند، کالیفرنیا**
- **Nutanix در سان‌خوزه، کالیفرنیا**
- **Portland State University در پورتلند، اورگن**
- **Sharp HealthCare در سان‌دیگو**
- **VMware در پالو آلتو**

چرا مدیر پروژه به انعطاف‌پذیری نیاز دارد؟

ترغیب می‌کنید و متعهد می‌سازید، یا اجبار می‌کنید و می‌جنگید؟!

اگر مدیر پروژه باشید حتماً دربارهٔ مدیریت پروژه بسیار خوانده‌اید. می‌توانید هر کدام از عناوین به‌روز دنیای مدیریت پروژه باشید؛ یک «رهبر اصیل»، «یک رهبر تسهیل‌کننده» یا یک «رهبر چابک».

اگر در آمازون جست‌وجو کنید یا در قفسه‌های کتابخانهٔ محلی‌تان بگردید، ده‌ها عنوان دیگر را خواهید یافت. اما این صفات به کلیشه بدل شده‌اند. بسامد کاربرد برخی شاید بیشتر و برخی شاید کمتر باشد. بااین‌حال واقعیت این است که هیچ فرمول مسلم و بی‌چون‌وچرایی برای رهبری وجود ندارد.

اما یک جنبه وجود دارد که دربین بسیاری از نظریه‌های امروزی مدیریت مشترک است؛ اینکه بنیان مدیریت موفق بر چهار روش استوار است:

۱. داشتن یک چشم‌انداز واضح؛

۲. تسهیم آن چشم‌انداز با دیگران به شیوه‌ای شفاف و جذاب؛

۳. فراهم آوردن منابعی که به این چشم‌انداز عینیت می‌بخشد؛

۴. هم‌گام کردن منافع ذی‌نفعان، که بعضاً با هم تعارض دارند.

ویژگی پنجم که معمولاً نادیده گرفته می‌شود، نیاز رهبران به منعطف بودن و تغییر دادن سبکشان

بر حسب موقعیت است. سبک یک پروژه‌ای که عقب افتاده و در موقعیت دشواری است، تفاوت دارد با سبک پروژه‌ای که هم‌چون یک سفر آرام فقط چند پستی‌وبلندی در جادهٔ پیش روی خود دارد.

یک مدیر ارشد خدمات عمومی تعریف می‌کرد که نتوانسته بود سبک مدیریتی‌اش را با موقعیتی تطبیق بدهد. باید یک برنامهٔ ۱۲ هفتگی را تحویل می‌دادند که ۵ هفته از زمان‌بندی عقب‌افتاده بود. او در تشخیص مشکلات عالی بود اما مسئولیت تصحیح آن‌ها را به‌عهده نمی‌گرفت.

نگرش کلان‌نگر این مدیر ارشد کافی نبود. در این مرحلهٔ بحرانی پروژه، باید جزئیات بیشتری را می‌فهمید، چه دربارهٔ پروژه و چه سازوکار تیم ارشدش. حداقل فاقد سه اصل از پنج اصل رهبری بود و سبک خود را با شرایط متغیر انطباق نمی‌داد. وقتی افراد به مسندهای رهبری می‌رسند و تا مدیریت ارشد پیش می‌روند و عهده‌دار مدیریت پروژه‌های بزرگ‌تر و سخت‌تر می‌شوند لازم است مجموعه‌مهارت‌های خود را توسعه بدهند. با هر ارتقای جایگاه رهبری، تغییری در خودپندارهٔ افراد نیز ایجاد می‌شود؛ مثلاً با ارتقا به جایگاه کسی که به رهبر یک تیم کمک می‌کند یا یک مدیر پروژه تا یک رهبر سازمانی.

یک رهبر برجسته قطعاً به قابلیت‌های متعددی نیاز دارد؛ مانند ترغیب کردن و اثرگذاری، شبکه‌بندی روابط، مذاکره کردن، رهبری تیم، کار تیمی، حساسیت فرهنگی، ارزش‌گذاری به تفاوت‌ها، مربیگری، توسعه و مدیریت عملکرد، ارزیابی، مدیریت تعارض‌ها، قاطعیت، تفکر راهبردی، تخصیص منابع، تواضع و مهارت ارائه دادن. جداً لیست پرزحمتی است.

هرچه جایگاهتان در سازمان بالاتر برود، تیمی که مدیریت می‌کنید بزرگ‌تر خواهد بود و به مهارت‌های بیشتری نیاز خواهید داشت. رهبرانِ شاخص خود را متمایز می‌کنند، نه با تعصب به یک جهان‌بینی معین دربارهٔ رهبری، بلکه با شیوهٔ متفاوت رفتارشان. هرچه مخزن رفتاری‌شان پرگوهرتر باشد، قابلیت‌های رهبری را پذیراترند و رهبری‌شان اثرگذارتر خواهد بود.

خوشبختانه تغییر رفتار جزء استعدادهای هر انسانی است. برعکس شخصیت که کم‌وبیشتر با DNA تعیین می‌شود، رفتار –در اینجا به معنای آنچه می‌گویید و می‌کنید– قابل‌شکل‌گیری است؛ می‌توان تغییرش داد یا به‌طور بنیانی بازتعریفش کرد. بیشتر مواقع می‌توانید رفتار خود را کنترل کنید و رفتار انتخابی‌تان را تمرین کنید. از این رو رفتار رفتار می‌زاید و آنچه می‌گویید و انجام می‌دهید تعیین‌کنندهٔ اشکال و واکنش‌هایی است که از دیگران می‌گیرید.

در دههٔ ۱۹۷۰ گروهی از روان‌شناسان کنجکاو شروع به تحقیق رفتارهایی کردند که در تعاملات تجاری، عملگران بامهارت را از عملگران متوسط تفکیک می‌کرد. مطالعات آن‌ها برای برخی از تعاملات منجر به مدل‌های موفق رفتاری شد که بیشتر این مدل‌ها به بخشی از لیست خصوصیات رهبری اضافه شد! از آن پس گروهی از افراد و کسب‌وکارهای مشتاق به آزمون و تنظیم این طرحواره‌های رفتاری ادامه داده‌اند. تحقیق آن‌ها به رهبران کمک کرده است که با آموختن شش مهارتی که ریشه در دو سبک رایج ترغیبی دارد در ویژگی‌هایی چون اثرگذاری و ترغیب استادی پیدا کنند. این دو سبک یکی اجبار یا هول‌دادن (push) است و دیگری ترغیب یا کشیدن (pull).

هر سبک از لحاظ رفتاری متمایز است و هر کدام برای موقعیت‌های مختلفی مناسب هستند. سبک ترغیب به این طریق است:

۱. داشتن ایده یا اعتقادی که با تو به اشتراک می‌گذارم؛

۲. به تو می‌گویم که به چه دلایلی ایدهٔ خوبی است یا چرا حق با من است؛

۳. موافقت می‌کنی و جایگاه‌ات را تغییر می‌دهی.

از لحاظ رفتاری سبک اجبار با سه نوع رفتار کلامی بخصوص مشخص می‌شود:

- پیشنهاد محتوا (ارائهٔ یک ایده)؛
- اطلاعات دادن (برهان آوردن)؛
- اعلام آن (صحبت کردن با دیگران دربارهٔ آن ایده).

راهکار را یک «اثرگذار» به کار می‌گیرد و این اثرگذار است که بیشتر مکالمات را انجام می‌دهد.

سبک اجباری یا «هول دادن» برای مدیران ارشد و در شرایطی که تأثیرگذار به‌دلیل جایگاهش از اختیار تی برخوردار است بهترین کارایی را دارد. در شرایط اضطراری یا وقتی که زمان ضرورت عاجل دارد نیز مفید است. به‌طورمثال وقتی که به ضرب‌الاجل یک پروژه نزدیک می‌شوید، و مدیر پروژه به تهییج تیم به عمل کردن نیاز دارد، سبک هول دادن مؤثرتر است. و البته تقریباً برای نیمی از موارد مفید است. شاید گاهی هول‌دهنده‌های عذرخواه یا پرخاشگری باشید. وقتی که راهکارتان را زودهنگام اجبار می‌کنید، هول‌دهنده‌ای هستید که قدرت تشخیص ندارید؛ زیرا قدرت مقاومتی را که با آن روبه‌رو می‌شوید نادیده گرفته‌اید!

مثلاً یک مدیر میانی را در یک کسب‌وکار چندملیتی درنظر بگیرید. باید به تیمش جهت جدیدی را بدهد؛ یک برنامهٔ منسجم و شفاف دارد و هرکدام از اعضای تیمش را آموزش داده که چه‌موقع چه کاری بکنند. برای او همه‌چیز شفاف است و جزئیات مشخص؛ او مسئول تیم است و تیم «باید» موافقت کند. سبک هول دادن یا اجبار چندان پیچیده نیست. با این همه این مدیرِ اجبارکننده یک سؤال اساسی را فراموش کرده است: چقدر متعهد بودن افراد به برنامه برای او اهمیت دارد؟ اگر تعهد ضرورت داشته باشد، این سبک ترغیب یا کشیدن است که معمولاً کارساز می‌افتد. کِشَنده‌ها سه رفتارِ بخصوص دارند: درخواستِ پیشنهاد می‌کنند (مثلاً بهترین روش انجام این کار چیست؟) اطلاعات خواهند (چه کسی در این زمینه تجربه دارد؟)، و از همه مهم‌تر، مهارتی نادر اما شایستهٔ احترام دارند: مهارت سازندگی.

سازندگی بسط و توسعهٔ پیشنهادی است که شخص دیگری آن را مطرح می‌کند. سازندگیِ به‌قدرِ ضرورتش انجام نمی‌شود. زیرا معمولاً ترغیب‌کننده‌ها بیشتر به ایده‌های خودشان علاقه‌مندند و در مهار و بهره‌وری از پیشنهادهای دیگران به همان میزان موفق نیستند. اگر آن مدیر میانی بر ترغیب تیمش تمرکز داشت، به‌جای کاری که کرد، از سبک کشیدن یا ترغیب استفاده می‌کرد:

۱. از تیم ایده‌هایشان را می‌پرسید؛
۲. گزینه‌هایی را پیشنهاد می‌داد؛
۳. برای بررسی پیشنهادها سؤال‌هایی را می‌پرسید؛
۴. برمبنای پیشنهادهای آن‌ها عمل می‌کرد؛
۵. او و تیمش بر یک راه در پیش روی خود توافق می‌کردند.

به این شیوه، افزایش مشارکت تیم سطح تعهد تیم را بالا می‌برد. سبک تعهد را می‌توان در تعاملات و در مربیگری دیگران برای به‌کارگیری منابعشان به‌کار برد. این سبک رهبری برای مدیریت موفق پروژه حیاتی است، هم در داخل تیم، و هم به‌منظور اثرگذاری بر کسانی که در کسب‌وکار هستند و هیچ اختیار مستقیمی برای هدایت آن‌ها نمی‌تواند وجود داشته باشد. سبک کشیدن یا ترغیب شاید زمان بیشتری ببرد اما مزایای آن بر هزینه‌هایش غالب می‌آید. مثلاً مدیر پروژه‌ای را درنظر بگیرید که می‌خواهد یکی از اعضای تیمش را متقاعد کند که باید مهارتش را در ارائه دادن تقویت کند. اگر از پیش سبک اجبار یا هول دادن داشته باشد، به او مستقیماً دستور می‌دهد و علتش را هم بیان می‌کرد. اما اگر سبک کشیدن داشته باشد، بهتر درک می‌کند که چرا آن عضو تیم مهارت ارائه‌اش مطابق انتظار نیست، یعنی جدا از مشکلات حاضر، نیاز زیربنایی او چیست. سپس آن دو گزینه‌های مختلفی را بررسی می‌کنند تا علت نقص را بفهمند؛ سپس آن عضو تیم به خروجی متعهد می‌شود. واقعاً این سبک ارزش تلاشش را دارد. نقطهٔ شروع تقویت انعطاف‌پذیری رفتاری شناخت و تمرین «مهارت‌های عمومی تعامل» است. منظور از مهارت‌های عمومی تعامل رفتارهای کلامی است که از تحقیقات درخصوص کار تیمی و گروهی گرد آورده شده‌اند.

با توجه به اینکه عمدهٔ شکست پروژه‌ها ناشی از مسائل انسانی است، تقویت این مهارت‌های تعاملی ضروری است. در نتیجهٔ آن، مدیران پروژه بهتر می‌توانند جلسات را مدیریت کنند، راهکارها را توسعه بدهند، تعاملی کار کنند، بر موقعیت اثر مثبتی بگذارند، درخواست‌ها را مدیریت کنند، ارتباطات را شفاف کنند و مشارکت گروه را هدایت کنند.

تقویت انعطاف‌پذیری رفتاری فقط به معنای دانستن الگوی رفتاری صحیح و تمرین آن نیست. باید دانست چه کاری را نباید کرد و چگونه از مشکلاتی پرهزینهٔ احتمالی اجتناب کرد.

بیاموزیم که رهبران موفق چه نکرده‌اند و رهبران ناموفق چه کرده‌اند... آن‌قدر متواضع باشیم که بر رفتار خود و دیگران تعمق کنیم. به‌یاد داشته باشیم که انعطاف‌پذیری و تغییر برحسب نیاز، مهمترین رکن موفقیت یک رهبر است که هرگز نباید فراموشش کند.

دوبارهم خیلی است!

در طراحی و توسعهٔ افزایشی (incremental) یک قاعدهٔ کلیدی کارساز است؛ می توان آن را «دو بار هم خیلی است» نامید.

اساساً وقتی میزان وسوسهٔ خودمان برای برداشتن و چسباندن (copy and paste) بخشی از کد را ملاحظه می کنیم، می توانیم بفهمیم که کد ما چقدر باید عمومی باشد؛ سپس به‌جای برداشتن و چسباندن آن، یک راهکار عمومی برای آن دو نیازمندی طراحی می کنیم. هرگاه وسوسه شدیم که یک راهکار را برای دو نیازمندی کپی کنیم، می توانیم از این قاعده استفاده کنیم. مثلاً فرض کنید که یک رمزگشای صوتی (audio decoder) طراحی می کنید که در ابتدا فقط از فایل‌های WAV پشتیبانی می کند. سپس می خواهید که یک تجزیه کننده (پارسر) MP3 نیز به کد بیفزایید. مطمئناً کد parsing چه برای MP3 چه WAV یکسان است، و به‌جای کپی کردن و چسباندن کد، فوراً یک سوپر کلاس یا کتابخانهٔ سودمندی (utility library) می سازید که فقط کاری را که برای این دو پیاده‌سازی نیاز است انجام بدهند.

نکتهٔ کلیدی این است که مانع وجود دو پیاده‌سازی رقیب شدید و فوراً یک راهکار عمومی ایجاد کردید.

به‌علاوه آن را بیش‌ازحد عمومی نکردید؛ راهکار فقط از WAN و MP3 پشتیبانی می کند و اصلاً توقع دیگر فرمت‌ها را ندارد.

بخش دیگر این قاعده این است که در وضعیت ایدئال، توسعه‌دهنده هیچ‌گاه نباید مجبور باشد که یک بخش از کد را به شیوهٔ مشابه یا یکسانی با شیوهٔ تغییر بخش دیگری از کد تغییر بدهد. نباید وقتی که Class B را به‌روزرسانی می کند «به‌یاد بیاورد» که Class A را به‌روزرسانی کند. نباید بداند اگر متغیر X تغییر کرد، باید فایل Y را به‌روزرسانی کنید. به‌عبارت‌دیگر نه تنها داشتن دو پیاده‌سازی اشتباه است، بلکه دو محل داشتن نیز خوب نیست. شاید پیاده‌سازی سیستم به این شیوه شدنی نباشد، اما حداقل می توان برای آن تلاش کرد.

اگر خود را در موقعیتی می بینید که مجبورید برای چیزی دو محل داشته باشید، مطمئن باشید که وقتی آن دو «همگام» نیستند (insync) سیستم به‌شدت و به‌وضوح از کار می افتد؛ کامپایل از کار می افتد، آزمونی که همیشه اجرا می شد از کار می افتد و غیره. غیرممکن است که بتوان به آن دو اجازه داد که ناهمگام شوند.

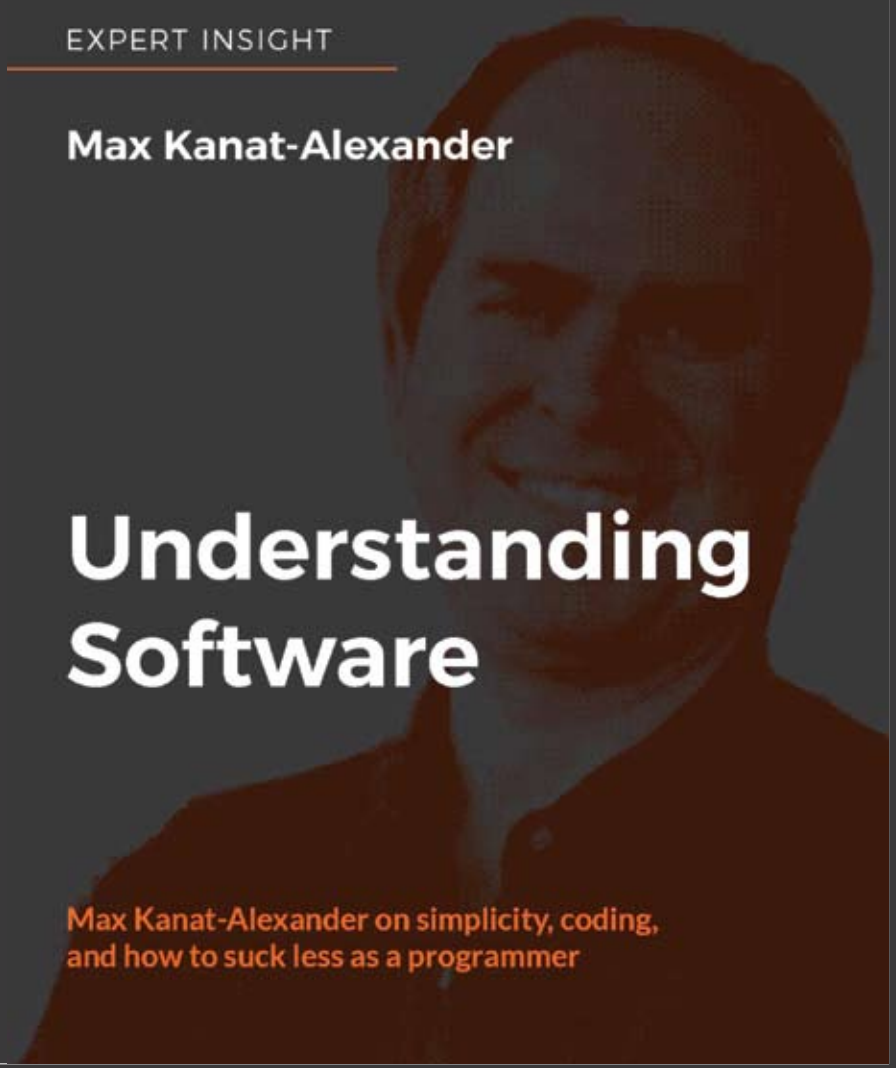
البته ساده‌ترین زیربنای این قاعده اصل کلاسیک «تکرار نکن» است. دو متغیر نداشته باشید که هردو دقیقاً نمایانگر یک چیز باشند؛ دو تابع نداشته باشید که هردو یک کار را انجام می‌دهند، و هیچ تکراری نکنید. شیوه‌های دیگری برای به‌کارگیری این قاعده وجود دارد. ایدهٔ کلی‌اش این است که وقتی می خواهید از یک مفهوم دو پیاده‌سازی داشته باشید، باید سعی کنید آن را فقط با یک پیاده‌سازی پیاده کنید. هنگام بازسازی، این قاعده به شما در یافتن مواردی که نیاز به تغییر دارند کمک می کند و شما را به چگونگی این تغییرات رهنمون می کند. وقتی در سیستم، منطق تکراری (duplicate logic) مشاهده می کنید، باید سعی کنید که آن دو محل را به یک محل تلفیق کنید. سپس اگر محل دیگری وجود دارد، آن یک محل را در سیستم عمومی جدید تلفیق کنید و این شیوه را ادامه بدهید. بدین ترتیب اگر چندین پیاده‌سازی مختلف وجود دارند که باید در یک پیادسازی تلفیق شوند، می توانید بازسازی افزایشی را با تلفیق همزمان پیاده‌سازی‌ها به‌انجام برسانید، البته مادام اینکه تلفیق، سیستم را ساده‌تر کند (و فهمیدن و نگهداری آن را آسان‌تر). گاهی باید برای بهینه‌ترین وضعیت، بهترین ترتیب تلفیق آن‌ها را بیابید؛ اما اگر نمی توانید بهترین ترتیب را بیابید نگران نباشید؛ فقط هردو را هم‌زمان تلفیق می کنید و معمولاً درآخر به یک راهکار مناسب برای تمام مشکلات دست می یابید.

نکتهٔ مهم دیگر این است که وقتی نیاز به تلفیق نیست، تلفیق نکنید. معمولاً تلفیق دو پیاده‌سازی در یک پیاده‌سازی موجب پیچیدگی بیشتر سیستم و تعدی از «اصل تک‌مسئولیتی» می شود. مثلاً اگر سیستم نمایش یک خودرو و یک شخص، مقداری کد مشابه دارند، این مسئله را با تلفیق کردن آن‌ها به صورت یک کلاس شخص-خودرو حل نکنید؛ این گونه پیچیدگی کاهش نمی یابد؛ زیرا شخص و خودرو واقعاً دو موجود متفاوت هستند و باید با دو کلاس مجزا نشان داده شوند.

قاعده‌ای که گفتیم یک قانون جهان‌شمول سفت‌وسخت و لازم‌الاجرا نیست؛ بیشتر یک راهنمایی روشنگر است که درحین توسعهٔ افزایشی به تصمیم‌گیری‌های مربوط به طراحی کمک می کند؛ اما برای بازسازی یک سیستم موروثی (legacy)، توسعهٔ یک سیستم جدید، و سادگی عمومی کد واقعاً مفید است.

چگونه می‌توان یک برنامه‌نویس فوق‌العاده شد؟

با آگاهی و ادراک و مسئولیت‌پذیری



برای یک برنامه‌نویس فوق‌العاده شدن به سه ویژگی مهم نیاز است: آگاهی، ادراک، مسئولیت‌پذیری.

مکس کانات (Max Kanat) در کتابش با نام «ادراک نرم‌افزار» موضوع ادراک را در کدنویسی بررسی و تجزیه‌وتحلیل می‌کند. شکی نیست که وقتی کاری را بهتر بفهمیم، بهتر انجامش می‌دهیم. مکس توضیح می‌دهد که چگونه این سه خصیصه یک برنامه‌نویس خوب می‌سازند.

هر کسی برای اینکه به یک توسعه‌دهندهٔ فوق‌العادهٔ نرم‌افزار تبدیل شود، علاوه‌بر ادراک به دو ویژگی دیگر نیاز دارد. در یک کلام، اگر از یک مشکل آگاه نباشید، هیچ ادراکی از آن حاصل نمی‌کنید؛ و اگر برای حل مشکل دست‌به‌کار نشوید (که اولین قدم آن مسئولیت‌پذیری است) اثرگذار نخواهید بود.

هر کدام از این سه ویژگی نکته‌های بسیاری دارد؛ اما وقتی کسی می‌خواهد توسعه‌دهندهٔ بهتری شود باید نکات کلیدی را در نظر بگیرید. در ادامه هر کدام از این ویژگی‌ها را با دقت بررسی می‌کنیم.

آگاهی

«آگاهی» شکل ویژه‌ای از ادراک است. «آگاهی» به این معناست که اساساً می‌توانید چیزی را مشاهده کنید و بدانید که وجود دارد. این ویژگی را به شیوه‌های مختلفی می‌توان در برنامه‌نویسی به کار گرفت. یک مثال بسیار ساده را بررسی کنیم: اگر از یک باگ مطلع نباشید، نمی‌توانید برطرفش کنید. گاهی آدم‌ها با بی‌اطلاع ماندن، از اقدام لازم برای حل مشکل فرار می‌کنند. به‌دلیل ناآگاهی‌شان از مشکل فکر می‌کنند لازم نیست کاری برای حل آن کنند. شاید هدفشان فرار از اقدام باشد، شاید نباشد؛ اما آنچه قطعی است وجود مشکل است، چه آن را بدانند و چه ندانند.

اگر سیستم شما باگی دارد که بر میلیون‌ها کاربر تأثیر می‌گذارد، پس باید از این باگ آگاه باشید. درغیراین‌صورت کاربران باید متحمل عواقب آن باشند. مطلع نبودن شما از باگ توجیه خوبی برای به‌دردسرانداختن کاربران نیست. اینکه شما از یک مشکلی نامطلع باشید، دلیل نمی‌شود که آن مشکل خودبه‌خود در عالم تصحیح شود. نمونه‌های بهتری برای آگاهی وجود دارد. مثلاً برخی از افراد واقعاً از پیچیدگی کد آگاه نیستند.

واقعاً نمی‌فهمند که کدی که نوشته‌اند پیچیده است.

گاهی حتی نمی‌دانند که پیچیدگی مسئله‌ساز است. یا مطلع نیستند که دیگر برنامه‌نویسانی که باید کد نوشته‌شدهٔ آن‌ها را بخوانند چه نظری

دربارهٔ آن دارند. توجه نمی‌کنند که کد توضیحات ندارد و خواندنش دشوار است؛ یا برای حل مشکل راه بهتری بود. تمام موارد ذکرشده مصداق‌های آگاهی در برنامه‌نویسی است.

ساده‌ترین راهی که می‌توانید آگاهی‌تان را به‌عنوان یک برنامه‌نویس بیشتر کنید بیرون رفتن از قالب خودتان و خواندن کدی است که خارج از حوزهٔ کنترلی شماست. شاید برای نوشتن کدِ رمزگذاری ویدئو به‌مدت یک‌روز کار کرده‌اید یا به‌طور روزانه زمان صرف کرده‌اید. اما آن ویدئو از جای دیگری می‌آید و کاربران آن را در سیستم شما بارگذاری می‌کنند. اگر آمدن ویدئوها به کدگذار شما با مشکل مواجه می‌شود، خوب است بررسی کنید که این کد واقعاً چگونه کار می‌کند.

ساده است، اما این کار، از لحاظ عاطفی، برای بسیاری از مردم دشوار است. راحت است که به خودمان بگوییم: «یک عده احمق خرابکاری می‌کنند و من باید تصحیحش کنم.» اما شما که کدشان را بررسی نکرده‌اید، از کجا می‌دانید؟ شاید چیزی را از آن بفهمید که کمکتان کند.

آگاهی می‌تواند اطلاع از وجود کتابخانه‌ها، چارچوب‌ها یا ابزاری باشد که می‌توانید در کدتان استفاده کنید. شاید نتوانید تمام ریزه‌کاری‌های جدیدترین کتابخانه‌های توسعهٔ وب را بدانید؛ اما اگر یک توسعه‌دهندهٔ وب هستید، حداقل باید از وجودشان آگاه باشید. به‌این ترتیب وقتی نیاز به تصمیم‌گیری دربارهٔ چارچوبی را دارید، می‌دانید که کدام یک را بررسی کنید. عموماً آگاهی با اشتیاق افزایش می‌یابد؛ اشتیاق به تجربهٔ زبان‌های برنامه‌نویسی جدید، الگوهای جدید طراحی، ایده‌های جدید دربارهٔ آزمون، چارچوب‌های جدید و مانند آن. مثلاً اگر می‌خواهید دربارهٔ روش‌ها و سیستم‌های نوین برنامه‌نویسی بفهمید، یکی از صفحات اولیهٔ ویکاهای مربوطه را می‌خوانید و گاهی مستندات را بیشتر کندوکاو می‌کنید. به این شیوه، حداقل می‌دانید که راه‌هایی وجود دارند، و اگر لازم شد می‌توانید بیشتر دربارهٔ این راه‌ها بیاموزید. اگر از شیوه‌های مختلف حل مشکلات آگاه نباشیم، به‌عنوان یک مهندس رایانه، ابزار بسیار کمتری را در اختیار خود خواهیم داشت. به‌این شیوه با افزایش آگاهی‌ام هنگام روبه‌رویی با یک موقعیت دشوار گزینه‌های بیشتری خواهم داشت. شاید ساده به‌نظر برسد، اما کاملاً درست است:

با بالا بردن سطح آگاهی خود از برنامه‌نویسی و برنامه‌ها، می‌توانید توانایی خود را به‌عنوان یک برنامه‌نویس تقویت کنید.

ادراک

هرچه کاری را بهتر بفهمیم، بهتر انجامش می‌دهیم. قدم اول آگاهی است، آگاهی از وجود یک مشکل. قدم بعدی فهمیدن آن مشکل است. یک مثال ساختگی را از شبه‌کد زیر مشاهده کنید:

```
function print_hello() {
clear_screen()
display_hello()
```

فکر می‌کنید این برنامه چه کاری انجام می‌دهد؟ به‌نظر می‌رسد که پایانه را پاک می‌کند و کلمهٔ Hello را چاپ می‌کند. اما این فقط حدس من است. هنوز بر مستندیا کد clear_screen یا write_hello نگاه نکرده‌ام. اگر کد را بررسی کنم، می‌بینم که clear_screen کل صفحه‌نمایش را سفید می‌کند و write_hello کلمهٔ Hello را به پائزده زبان مختلف چاپ می‌کند و بر صفحه‌نمایش رنگ می‌کند. اما بدون «بررسیِ» چیزی نمی‌توانیم آن را بشناسیم.

شاید این مثال بسیار ساده شده باشد اما همیشه در برنامه‌نویسی اتفاق می‌افتد. اکنون که می‌دانم clear_screen چه می‌کند، می‌توانم آن را هرجای دیگر، بدون نیاز به فکر کردن دربارهٔ آن یا صرف زمانی برای نوشتن آن، به‌کار گیرم. می‌توانم به‌جای یک برنامهٔ باگ‌دار، برنامه‌ای بنویسم که کاری را که قصدش را دارم انجام می‌دهد. می‌توانم برای آن یک آزمون بنویسم. سادگی و مقاومت و دیگر کیفیاتی که مختص‌به «برنامه‌نویسی درست» است، تماماً از ادراک نشئت می‌گیرند.

ویژگی ادراک به شیوه‌های دیگری نیز نمود می‌کند. اما نکتهٔ کلیدی‌ای است که از یک‌نفر یک برنامه‌نویس می‌سازد یا او را از برنامه‌نویسی ساقط می‌کند. مهارت‌های مفید دیگری نیز هستند، تایپ سریع، ارتباطات خوب، مدیریت مناسب اولویت‌بندی‌ها و دیگر مهارت‌ها؛ اما نکتهٔ کلیدی ادراک است. بدون آن می‌توانید سریع‌ترین برنامه‌نویس در دنیا باشید اما فقط سیستم‌های وحشتناک و غیرقابل‌نگهداری می‌نویسید! می‌توانید جذاب‌ترین فرد تیم باشید اما نمی‌توانید یک خط کد بنویسید. منظور این نیست که سریع بودن یا جذاب بودن خوب نیست؛ این مهارت‌ها پرفایده‌اند، اما قدرت ادراک که نباشد، یک توسعه‌دهندهٔ فوق‌العاده نخواهید بود.

مسئولیت‌پذیری

منظور از مسئولیت‌پذیری در اینجا چیست؟ تعریف معمول مسئولیت‌پذیری این است که مسئولیت کار اشتباهم را بپذیریم. این تعریف رایج مسئولیت‌پذیری است، اما در اینجا منظور ما از مسئولیت‌پذیری این نیست. بلکه مسئولیت‌پذیری در این موضوع به‌معنای «میل و اشتیاق به اصلاح مشکل است». مثلاً اگر مسئولیت اتاقم را داشته باشم، مایلم که تمیزش کنم. به این معنا نیست که باید تمیزش کنم، واقعاً مایلم چنین کنم. میل و اشتیاق به این معنا نیست که بگویم «بله، دوست دارم اتاقم را تمیز کنم» و سپس دست از کار بکشم و سه ساعت در اتاق به‌هم‌ریخته‌ام دراز بکشم. منظور میل واقعی است. مثلاً چقدر مایلید که دسر موردعلاقه‌تان را بخورید؟ به این «میل واقعی» می‌گویند. منظور این نیست که می‌توانیم دربارهٔ هرچیزی به اندازهٔ دسر موردعلاقه‌مان اشتیاق داشته باشیم. بلکه منظور این است که باید واقعاً میل داشته باشیم.

این مفهوم را چگونه می‌توانیم در برنامه‌نویسی به‌کار گیریم؟ همیشه به این مثال منفی برمی‌خوریم:

«نمی‌خواهم این کد را بازسازی کنم، چون کد من نیست.»

احمقانه است. آیا می‌توانید ویرایشش کنید؟ می‌توانید کد منبع را بخوانید؟ آیا اجازه دارید که تغییرات را به صاحب آن بفرستید؟ پس از لحاظ تئوری قادر به تصحیح آن هستید. منظور این نیست که باید بروید و تمام کدهای دنیا را تصحیح کنید؛ زیرا بر اهداف شما تأثیری ندارد. اما مردم بیشتر مواقع چنین جملات آزاردهنده‌ای را می‌گویند: «نمی‌توانم برای تصحیح کد تیم زمان بگذارم زیرا پروژه‌ام طولانی‌تر می‌شود» یا «تصحیح این کدی که کارم به آن بستگی دارد، بخشی از اهداف تیمم نیست». در این تئوری‌ها دو مشکل وجود دارد (واقعاً تئوری‌اند، واقعیت ندارند).

اول این که وقتی که کارتان به یک کد پیچیده و احمقانه و نامناسب وابسته است، معمولاً ادامهٔ کد نوشتن بر آن زمان‌گیرتر است. غالباً اینکه کد را بازسازی کنید و سپس ویژگی خود را بر آن بنویسید کمتر وقت می‌گیرد تا سعی کنید ویژگی موردنظر خود را بر یک کتابخانهٔ وحشتناک و به‌هم‌ریخته‌ای که اصلاً طوری که شما می‌خواهید کار نمی‌کند بنویسید. این کار زمان زیادی از اهداف شما می‌گیرد. علت اینکه خلافش به‌نظر می‌رسد این است که تفاوت بین «انجام کار» و

«انجام واقعی کار» را نمی‌دانید.

منظور از «انجام واقعی کار» چیست؟ به این معناست که کد کار کند، پر از باگ نباشد، بتوان به‌راحتی نگهداری‌اش کرد، زندگی کاری‌تان را از هم نپاشد، بتوانید یک قدم جلوتر بگذارید و بازدهی بیشتری حاصل کنید. تصور کنید که یک ساختمان می‌سازید و با چند آجر تقلبی شروع می‌کنید. سپس چند لایه آجر بر زمین می‌گذارید. اما وقتی به لایهٔ چهارم آجر می‌رسید، آجرهای تقلبی در لایهٔ اول شروع به شکستن می‌کنند. پس برمی‌گردید و با افزودن چند چوب تقلبی زیر آجرها را سفت می‌کنید. بعد از آن چند لایهٔ آجر دیگر می‌چینید؛ حالا دیوار فرو می‌ریزد. بنابراین چند نوار آهنی زنگ‌زده را زیر آن قرار می‌دهید و دنبال کارتان می‌روید. اگر این روش را ادامه بدهید، باید تمام زندگی‌تان را صرف حفظ دیوار کنید و دیوار به مشکل بزرگی در زندگی‌تان تبدیل می‌شود. احتمالاً آخر سر مجبور می‌شوید کنار بکشید تا شخص دیگری که اکنون مجبور است از این دیوار فاجعه‌آمیز نگهداری کند پیدا شود. اما دیگران حتی آن را نمی‌فهمند، چرا که ترکیبی است از مواد تقلبی ساختمانی که هیچ فرد عاقلی با هم ترکیبشان نمی‌کرد؛ حقیقتاً نابخردانه است!

وقتی بی‌هدف اطراف یک کد نامناسب پرسه می‌زنید و سرهم‌بندی‌اش می‌کنید، نرم‌افزارتان را مثل همان دیوار می‌سازید. شاید ظاهراً شبیه آن نباشد؛ زیرا برنامه‌ها ساختار فیزیکی بزرگی ندارند که بر سر شما آوار شوند. اما اصول ساخت یکسان است؛ وقتی یک پیچیدگی زیرساختاری را با افزودن پیچیدگی بیشتری بر آن نادیده می‌گیرید، پیچیدگی کل سیستم را افزایش می‌دهید. اما اگر پیچیدگی زیرساختاری را برطرف کنید، پیچیدگی کل سیستم را کم می‌کنید.

مهم نیست که چه کسی اول موجب این پیچیدگی بوده است. این واقعیت که «کس دیگری خرابکاری کرده» ناقض نکات ذکرشده نیست. مهم نیست که چه کسی عامل پیچیدگی است. اگر بر زیرساخت اشتباهی ادامه بدهید، فقط به پیچیدگی سیستم اضافه می‌کنید. اگر تصحیحش کنید، مانع از پیچیده‌تر شدن سیستم می‌شوید. انتخاب کنید؛ قدرت انتخاب دارید، می‌توانید مسئولیت‌پذیر باشید. گاهی مسئولیت‌پذیری با محول کردن کار تصحیح به کس دیگری نمود می‌یابد. اما تجربه اثبات کرده اغلب خودمان باید به اصلاح آن میل داشته باشیم.

همه این‌ها را می‌دانیم اما باز گاهی فکر می‌کنیم: «همین دفعه این قطعه را کمی پیچیده‌تر کنم؛ فقط همین بار که وقت نیست» و البته گاهی حق با شماست، خصوصاً زمانی که موقعیت کمی اضطراری است و مجبورید برای مدت کوتاهی همین کد را ادامه بدهید. اما اغلب موارد به یک تودهٔ بزرگی از یک دیوار شکسته می‌افزایید که برای شما و کسی که قرار است از آن نگهداری کند به یک کابوس بدل خواهد شد. همین پیچیدگی‌های کوچک، همین بی‌مسئولیت‌های ناچیز، نهایتاً به انبوه غول‌آسایی از ویرانه‌ها می‌افزاید که هیچ کسی نمی‌تواند آن را تصحیح کند. بنابراین وقتی از «مسئولیت‌پذیری» صحبت می‌کنیم، بخش اعظم تعریف آن به معنای «تمایل به اصلاح کارها خارج از حوزهٔ طبیعی شماست.» این حوزه قرار نیست نامحدود باشد. می‌توانید مرزی قائل شوید و بگویید «از این نقطه به بعد، واقعاً مشکل من نیست.» مثلاً در پروژه‌هایتان می‌توانید به این مرز قائل باشید که «من روی کدی که از بیرون از شرکت می‌آید کار نمی‌کنم چون استفادهٔ بهینه از زمانم نیست.» با این حال اگر گاهی مسئولیت برخی از کدهایی را که فرای محدودهٔ پروژهٔ شماست بپذیرید، برنامه‌نویس بسیار بهتری خواهید شد. هرچه این محدوده وسیع‌تر شود، برنامه‌نویس بهتری خواهید بود.

تا اینجا یکی از مشکلات تئوری‌های گفته‌شده را گفتیم. اما مشکل دوم چیست؟ خطای دوم این تئوری این است که شما به یک گروه بزرگ‌تر تعلق دارید، به گروه نوع بشر. تنها شما مطرح نیستید، چه عیبی دارد که به کد دیگران کمک کنید. همهٔ ما در یک تیم هستیم.

اگر برای یک شرکت کار می‌کنید با تصحیح مشکلاتی که به آن‌ها برمی‌خورید به کل شرکت کمک می‌کنید. و اگر یک توسعه‌دهندهٔ مستقل در این دنیا باشید، وقتی برخی از کتابخانه‌های پرکاربرد یا ابزار معروف یا کدهای نمونه در اینترنت را تصحیح می‌کنید، با کمک به دیگر برنامه‌نویسان، دنیا را به مکان بهتری مبدل می‌سازید، و درنتیجه، برنامه‌نویس بهتری خواهید بود. بهترین برنامه‌نویسان کسانی هستند که مایل‌اند هر چیزی را که خطا دارد بپذیرند، صرف‌نظر از اینکه چه کدی است یا چه تیمی آن را نوشته است.

به هر حال، شما که مصرف‌کنندهٔ یک کتابخانه یا API هستید، غالباً بهترین جایگاه را برای بازسازی آن را دارید؛ زیرا بهتر از مؤلف آن می‌دانید که چگونه باید از آن API یا کتابخانه استفاده کرد. حداقل باگ را فایل‌بندی کنید و مشکلی را که با آن داشتید ذکر کنید. درغیر این صورت مؤلفان چگونه خواهند فهمید که مشکلی دارند؟ شاید انتظارش را داشته باشند، اما باور کنید اغلب نمی‌دانند؛ که می‌داند شاید تجربهٔ شما برای آن‌ها باارزش باشد.

خلاصه

به‌طور کلی، اگر می‌خواهید که برنامه‌نویس بهتری باشید، از خودتان بپرسید که آیا نباید آگاهی، ادراک یا مسئولیت‌پذیری خودم را بهبود بدهم و برای مدتی بر آن متمرکز باشم؟ اگر مطمئن نیستید به کدام نیاز دارید، با آگاهی شروع کنید، سپس به ادراک بروید، و درنهایت مسئولیت بپذیرید. انجام آن به شیوه‌ای دیگر دشوار است. نمی‌توانید به‌آسانی مسئولیت چیزی را بپذیرید که درک نمی‌کنید (گیج‌کننده است و نمی‌توانید درست کار کنید) و ادراک چیزی که از آن آگاه نیستید ناممکن است. بنابراین آگاهی، ادراک و مسئولیت‌پذیری توالی درستی است. اگر باید آگاهی‌تان را بهبود بدهید، چند کد جدید بخوانید، یک وبلاگ جدید برنامه‌نویسی پیدا کنید، دنبال چند کتاب جدید بگردید، با دیگر برنامه‌نویس‌ها دربارهٔ فناوری‌های اخیر صحبت کنید؛ سعی کنید از مشکلات، راهکارها، دانش، الگوها، افراد، سازمان‌ها، اصول، یا هرچیز مؤثر در کارتان مطلع بمانید. اگر هدفتان تقویت ادراکتان است، چند مستند جدید بخوانید، زمان بیشتری برای فهمیدن نحوهٔ کار توابع صرف کنید، از برنامه‌نویسان گروهتان پرس‌وجو کنید، معنی چند کلمه را در لغت‌نامه پیدا کنید، دربارهٔ فناوری مورداستفادهٔ خود بخوانید، یا کتاب بخوانید؛ هر روشی را که به درک تمام‌وکمال شما از کارتان کمک می‌کند به کار بگیرید. درنهایت برای مسئولیت‌پذیر بودن فقط باید تصمیم بگیرید که مسئولیت بیشتری را قبول کنید. وقتی که مشکلی پیش راه شما قرار می‌گیرد، تصمیم بگیرید به‌جای به تعویق انداختنِ آن حلش کنید. وقتی خارج از تیم شما مشکلی وجود دارد، به‌جای اینکه بخشی از آن باشید، تصمیم بگیرید که راهکاری ارائه کنید. یک نوعِ ویژهٔ مسئولیت‌پذیری نیز وجود دارد؛ وقتی شخص دیگری مشکلی دارد، کمکش کنید و درصورت لزوم، اجازه بدهید خودش آن را حل کند. مسئولیت‌پذیری فقط به این معنا نیست که باید هر کاری را انجام بدهید؛ بلکه می‌تواند به این معنا باشد که به دیگران کمک کنید تا خودشان آن را انجام بدهند. توصیه‌ها فوق دشوار نیستند، مادام اینکه با قدم‌های ساده شروع کنید. لازم نیست که یک‌شبه از کدهای کل عالم آگاه شوید. لازم نیست که هر کلمه‌ای از هر برنامه‌ای که تابه‌حال نوشته‌شده را بفهمید؛ و چون این مقاله را خوانده‌اید حاضر باشید که هر قطعهٔ نرم‌افزاری را که در عرصهٔ وجود است تصحیح کنید! با یک کار کوچک شروع کنید. پس از تکمیل آن، کار دیگری را شروع کنید و بعدی و بعدی؛ به‌موقع همان برنامه‌نویس خوبی خواهید شد که می‌خواهید.

مدیران پروژه دنیا را تغییر می‌دهند

فقط خیال‌اندیشان و رؤیاسازان نیستند که دنیا را تغییر می‌دهند. هیچ‌وقت برای تغییر ایده‌داشتن کافی نیست. ایده را باید به‌عمل آورد و اجرا کرد؛ یعنی همان کاری که مدیران پروژه می‌کنند! ایده‌ها اگر عملی نشوند فایده‌ای ندارند. مردم هرروزه ایده‌ای دارند؛ اما اغلبشان ایده‌های بزرگشان را عملی نمی‌کنند. وقتی آدم‌ها ایده‌های بزرگی می‌یابند، دو حالت وجود دارد:

۱. به دوستشان می‌گویند: «یک ایدهٔ خوب دارم اما هیچ‌وقت عملی‌اش نکرده‌ام...»

۲. هیچ‌وقت آن را به کسی نمی‌گویند؛ می‌ترسند که بدزندش! بعد هم کاری برای اجرایش نمی‌کنند...

ایده‌هایی که به‌عمل نمی‌آیند همان انگاره می‌مانند. ایده به‌خودی‌خود چیزی نیست! کسی باید باشد که قصدِ متجلی کردن آن را داشته باشد. کسی باید آن ایده را به یک برنامه مبدل کند و به‌اجرا درآوردش. کسی باید:

• بفهمد که به چه نیاز است؟

• بفهمد که چگونه می‌توان ایده را به‌وجود آورد؟

• کارهای موردنیاز برای واقعی کردن ایده را هماهنگ و هدایت کند.

ایده‌های شگرف به یک رهبر و یک برنامه نیاز دارند. پروژه‌های بسیاری در زمینهٔ فناوری، بهداشت و سلامت، معماری، حمل‌ونقل، کسب‌وکار، و صنایع متعدد دیگر وجود داشته‌اند.

ایده‌های بزرگ به تیمی از افرادی نیاز دارند که با همکاری هم ایده را به یک برنامه بدل کنند و آن را اجرا کنند؛ نیاز به تیمی دارند که تمام نیازها را بشناسد و برای واقعی کردن یک برنامه دست به عمل بزند. رهبر کسی است که کارها را هماهنگ می‌کند، انتقال می‌دهد و پیشروی امور را حفظ می‌کند؛ البته الزاماً همان کسی نیست که ایده را مطرح کرده است؛ به چند دلیل:

• ایده‌دهنده مایل به رهبری نیست. به کارهای بسیاری مشغول است؛ قابل‌فهم است که کار مدیریت پروژه را به کس دیگری واگذار کند؛

• از مدیریت پروژه لذت نمی‌برد و نقاط قوتش در زمینه‌های دیگر است؛

• نمی‌داند برای متجلی کردن رؤیایش چگونه یک برنامه ایجاد کند.

فقط پروژه‌های عظیم نیستند که برای تغییرآفرینی به مهارت‌های مدیریت پروژه نیاز دارند. تغییرات کوچکی که وضعیت فناوری، سلامت و بهداشت، کسب‌وکارها و کیفیت زندگی را بهبود می‌بخشند به مدیران پروژه‌ای نیاز دارند که به موفقیت آن‌ها کمک کنند.

مدیران پروژه به رؤیاسازان کمک می‌کنند که رؤیای خود را بفهمند. یک مدیر پروژه با رؤیاساز یا ایده‌انگار کار می‌کند تا بتوانند تصویر بزرگ‌تر را بهتر بفهمند. با دقت جزئیات را بررسی می‌کند. با تیم کار می‌کند تا بتوانند برای ایجاد تغییرات یک برنامه ایجاد کنند. با فهماندن پروژه فاصلهٔ بین مخاطبان و تیم‌ها را پر می‌کند. با پشتیبانی از تیم، آن را به‌جلو می‌راند و خود به بخشی از تغییر تبدیل می‌شوند، چه این تغییر بزرگ باشد و چه کوچک. چه‌بسا یک پروژهٔ کوچک ریپلی ایجاد کند که تأثیرش از آنچه در این لحظه فکر می‌کنید بسیار بیشتر باشد.

مدیران پروژه چنین کار بزرگی دارند... آن‌ها متجلی‌کنندگان رؤیاها و ایده‌ها هستند.

کتاب جدیدی در زمینهٔ برنامه‌ریزی از

APMG

انتشارات سمت، نشر دانشگاهی، گروه‌های تحقیقاتی، گروه‌های مؤلفان و مترجمان ما نیازمند الگویی منسجم و برنامه‌ریزی‌شده برای تألیف و حتی ترجمهٔ آثار کاربردی و درعین‌حال سلیس و خواندنی هستند. با بررسی نحوهٔ تألفی کتاب جدید APM در زمینهٔ مدیریت پروژه سعی می‌کنیم که الگویی را برای تألیف کتاب‌های علمی-عملی داخلی معرفی کنیم. نائب‌رئیس انجمن مدیریت پروژه (APM)، تام تیلور، از تلاش‌هایی که برای عرضهٔ کتاب «برنامه‌ریزی، زمان‌بندی، نظارت و کنترل (PSMC)» در اگوست صورت گرفت قدردانی کرد و از گروهِ مؤلفان این کتاب خواست که تواضع بریتانیایی خود را کنار بگذارند و به جهان اعلام کنند که چه کار فوق‌العاده‌ای انجام داده‌اند. مؤلفان این کتاب با انگیزهٔ نوشتن بهترین کتابی که تابه‌حال در زمینهٔ مدیریت پروژه نوشته شده آغاز به کار تألیف این کتاب کردند.

موضوع

«راهنمای PSMC» خلأ موجود در کتاب‌های منتشرشدهٔ APM را پر می‌کند. APM کارهای غنی و فراوانی از معرفی‌ها، راهنماها و مقالات دربارهٔ مهارت یافتن در مدیریت پروژه منتشر کرده است. اما در دنیای واقعی مدیریت پروژه، اجرای برخی از اصول مدیریت پروژه با مشکلاتی روبه‌رو می‌شود. این کتاب راهنما هم راهنمای مناسبی برای اصول اولیه است و هم جدیدترین پیشرفت‌ها در چابک (Agile) و BIM را پوشش می‌دهد. APM دریافت که با یک گروه متشکل از مؤلفان از تمام زمینه‌های صنعت و چند تکنیک مرسوم سنجش کیفیت تألیف می‌تواند اثر بهتری داشته باشد.

در این کتاب سعی شده که سبک‌های مختلف آموزشی درنظر گرفته شود؛ به‌طورمثال برخی متن را ترجیح می‌دهند و برخی تصاویر را. همتایان این مؤلفان از داخل و خارج از پروژه بخش‌های اول و دوم این اثر را بررسی می‌کردند و بازخورد (فیدبک) می‌دادند تا مشخص شود کدام بخش بیشترین توجه کدام خواننده را جلب کرده است! نتیجهٔ کار تألیف این کتاب، ۱۵۰ تصویر و جدول بود، که برخی با دقت از سازمان‌های خود مؤلفان اتخاذ شده و برخی واقعی و اصیل‌اند.

محور کار این مؤلفان شفاف‌نویسی و ساده‌نویسی است. امیلی تیلور، دختر سیمون تیلور که یکی از مؤلفان است، آزمون سنجش سلیسی متن را هنگام نوشتن و بررسی و نهایی‌سازی کلمات عهده‌دار بوده است، که مؤلفان به‌مزاح، این بررسی را «تست امیلی» می‌نامند. این آزمون زبان متظاهر و پرطمطراق برخی از مؤلفان را به‌چالش می‌کشید و مؤلفان را به این وامی‌داشت که به شیوهٔ قابل‌فهم‌تر و مختصرومفیدتری بنویسند.

مطمئناً کتاب نهایی برای ده‌ساله‌ها مناسب نیست اما از آزمون خوانش‌پذیری Flesh-Kincaid، آزمونی برای نشان دادن میزان دشواری فهمیدن یک متن به زبان انگلیسی، استفاده کردند تا مطمئن شوند که نوشته بالاتر از سطح علمی کارشناسی نباشد. تقریباً تمام فصل‌ها را با این آزمون سنجیدند؛ تا برای سطوح پایین‌ترِ آموزشی مناسب باشد. هدف مؤلفان این بوده که چیزی بنویسند که برای همه مناسب و مفید باشد.

این کتاب را باید با چه معیارهایی سنجید؟

انتشارات سمت، نشر دانشگاهی، گروه‌های تحقیقاتی، گروه‌های مؤلفان و مترجمان ما نیازمند الگویی منسجم و برنامه‌ریزی‌شده برای تألیف و حتی ترجمهٔ آثار کاربردی و درعین‌حال سلیس و خواندنی هستند. با بررسی نحوهٔ تألفی کتاب جدید APM در زمینهٔ مدیریت پروژه سعی می‌کنیم که الگویی را برای تألیف کتاب‌های علمی-عملی داخلی معرفی کنیم.

می‌توان چهار معیار سنجش (متریک) برای این کتاب و مانند آن درنظر گرفت:

- آیا عملی است؟ از این لحاظ که استفاده‌کننده از آن بتواند راهنمایی‌ها را به‌کار گیرد؟
- آیا عملی است؟ از این لحاظ که مبنای رئوس مطالب درسی و مجموعه‌ای از کیفیات موردنیاز مدیران یا افراد شرکت‌کننده در آزمون‌ها باشد؟
- آیا عملی است؟ از این لحاظ که هر کسی که آن را بخواند بتواند مفاهیم درون آن را درک کند؟
- و درنهایت آیا جالب است؟

در معیار آخری احتمالاً جواب منفی است. اما حس اشتیاق و کار گروهی مؤلفان می‌تواند جالب بودن تألیف این اثر را به مخاطب آن انتقال بدهد.

این گروه مؤلفان متشکل بود از Keith Haward، Jenn Browne، Carolyn Limbert، Simon Taylor، Mike Prescott، Stephen Jones، Guy Hindley، Franco Pittoni، Ewen Mclean و بسیاری دیگر.

این راهنما را گروه‌های همتای مؤلفان، متشکل از ۳۰ نفر از داخل و خارج این پروژه، بررسی کردند و می‌توان آن را از فروشگاه APM و کتاب‌های تجاری APMG تهیه کرد. این راهنما برای مدرک APMG با عنوان Project Planning & Control™ مناسب است و در همکاری با APM تألیف شده است. گواهی و راهنمای APMG توأمان بینش مناسبی را از دیدگاه کارشناسان پروژه فراهم می‌کند؛ به‌طوری که می‌توان فهمید که چرا و چگونه پروژه‌ها شکست می‌خورند و چگونه می‌توان بهترین شیوه‌های تحویل موفق پروژه را با جزئیات شناخت و عوامل کلیدی موفقیت در مدیریت پروژه را به‌کار گرفت.

نرم‌افزار یادگیری گوگل می‌آموزد که نرم‌افزار یادگیری بنویسد

هم‌زمان که نرم‌افزارها پتانسیل به‌عهده گرفتن کار حسابداران و وکیلان را می‌یابند، در بحث‌های مربوط‌به قدرت درحال‌رشد رایانه‌ها، اتوماسیون اداری به یک لغت باب روز بدل شده است. متخصصان هوش مصنوعی گوگل سعی می‌کنند که وظایف کارمندان حقوق‌بالا را خودکار کنند.

در پروژه‌ای با نام AutoML، محققان گوگل به نرم‌افزار یادگیری ماشین آموخته‌اند که نرم‌افزار یادگیری دیگری بسازد. در برخی از مواقع خروجی این نرم‌افزارها از بهترین سیستم‌هایی که خود محققان طراحی می‌کنند قدرتمندتر و مؤثرتر است. گوگل می‌گوید که سیستم اخیراً به رکورد ۹۱درصدی در دسته‌بندی عکس‌ها برحسب محتوایشان دست یافته است. در وظیفهٔ دشوار علامت‌گذاری محل چندین شیء در یک تصویر، که وظیفهٔ مهمی برای ربات‌های خودکار است، سیستم خودتولیدکننده به امتیاز ۴۳درصد رسید. این در حالی است که بهترین سیستم‌های ساخت بشر امتیاز ۳۹درصد می‌گیرند.

اهمیت این نتیجه بسیار است زیرا ساخت سیستم‌های نوین و مدرن هوش مصنوعی با کمبود منبع مواجه است، حتی در گوگل. سال گذشته (اکتبر ۲۰۱۷) ساندار پیچای، مدیرعامل گوگل در رویدادی که برای معرفی گوشی‌های هوشمند جدید و دیگر ابزاره‌ها برگزار شده بود، مختصراً به AutoML اشاره کرد و گفت: «امروزه این سیستم‌ها دست‌ساختهٔ دانشمندان یادگیری ماشین هستند و فقط چند هزار دانشمند در دنیا هستند که قادر به ساخت آن‌ها هستند. ما می‌خواهیم صدها هزار توسعه‌دهنده را قادر به این کار کنیم.»

AutoML یک پروژهٔ تحقیقاتی باقی می‌ماند. جالب است که حتی همین‌اکنون، باید متخصصان نادری داشته باشند که کارکرد آن را خودکار کنند. اما محققان بسیاری خارج از گوگل نیز بر این فناوری کار می‌کنند. اگر هوش مصنوعی ساختِ هوش مصنوعی عملی شود، آموزش ماشین می‌تواند خارج از صنعت فناوری گسترش یابد، مثلاً در حوزهٔ سلامت‌وبهداشت و مالی.

AutoML می‌تواند به راهبرد پیچای مبنی بر «اول، هوش مصنوعی» شتاب

بدهد؛ به‌موجب این راهبرد است که شرکت از یادگیری ماشین برای ادارهٔ مؤثرتر و ساخت محصولات جدید استفاده می‌کند. به‌طور مثال محققان گروه تحقیقاتی Brain در گوگل یا آزمایشگاه تحقیقاتی لندنیِ DeepMind، که در سال ۲۰۱۴ گوگل مالک آن شد، توانستند به تفکیک صورت‌حساب‌ها در مرکز داده‌های شرکت کمک کنند و توانایی گوگل را در نقشه‌کشی شهرها بالا ببرند. AutoML می‌تواند بازدهی این متخصصان را بیشتر کند و به مهندسان کم‌مهارت‌تر کمک کند که خودشان سیستم‌های قدرتمند هوش مصنوعی بسازند.

گوگل فقط ۱۳۰۰ نفر را در وبگاه تحقیقاتی‌اش لیست کرده، که همهٔ آن‌ها در هوش مصنوعی متخصص نیستند. هزاران مهندس نرم‌افزار دیگر دارد. گوگل مسئول Alphabet است که طبق یکی از آخرین بایگانی‌های سالیانهٔ مالی‌اش، ۲۷۱۶۹ کارمند در حوزهٔ تحقیقات و توسعه دارد.

گوگل از معرفی کسی برای پاسخگویی به بحث AutoML اجتناب می‌کند. محققان خارج از این شرکت معتقدند که ایدهٔ خودکارسازیِ بخشی از کار متخصصان هوش مصنوعی به یک موضوع داغ تحقیقاتی بدل شده است؛ و مستلزم پیچیدگی بیشتر هوش مصنوعی است.

هدف بیشتر کاری که در خصوص «فرایادگیری» یا «آموختنِ آموختن» انجام می‌شود، از جمله کار گوگل، سرعت دادن به فرایند توسعهٔ شبکه‌های عصبی مصنوعی است. این تکنیک شامل خوراندن داده‌ها از طریق شبکه‌های عملیات ریاضی است و تا حدی از تحقیقات در زمینهٔ نرون‌های مغز الهام گرفته شده است.

شاید شدیداً پیچیده و هوشمند به‌نظر برسد، اما به کارگرفتن شبکه‌های عصبی در ترفندهای مفیدی چون پردازش صوت، به کار خسته‌کننده و دشواری نیاز دارد که مستلزم دستمزدهای بالاست. متخصصان باید از حس غریزی و خطاوآزمون استفاده کنند تا معماری درستی را برای یک شبکهٔ عصبی بیابند. رُبرتو کالاندرا، محقق دانشگاه کالیفرنیای برکلی، معتقد است: «بخش اعظم کار مهندسْ وظیفه‌ای بسیار خسته‌کننده است؛ او باید چند پیکربندی را بررسی کند تا ببیند کدام بهتر کار می‌کنند.» او می‌گوید که این چالش سخت‌تر خواهد شد؛ زیرا محققان شبکه‌های بزرگ‌تری می‌سازند که موجب مشکلات دشوارتر خواهد بود.

کالاندرا بعد از گذران دو هفتهٔ وحشتناک در سال ۲۰۱۳ که برای تحقیقات دکترایش باید یک ربات را به راه‌رفتن وامی‌داشت، تحقیق دربارهٔ فرایادگیری را آغاز کرد. او یک تکنیک آزمایشگاهی را برای تنظیم خودکار نرم‌افزار به کار برد؛ این تکنیک برحسب تکنیک‌های یادگیری ماشین است و پیچیدگی کمتری از شبکهٔ عصبی دارد. ماشینِ سرسخت بعداز یک هفته به راه‌رفتن افتاد.

مهریار مُهری، پرفسور و استاد دانشگاه نیویورک، می‌گوید که تولید یک سیستم شبکهٔ عصبی از ابتدا از تنظیم یک سیستم موجود دشوار است. اما این تحقیق نشان می‌دهد که به عملی‌تر شدن آن نزدیک می‌شویم.

مُهری در همکاری با محققان دفتر نیویورک گوگل، بر سیستمی با نام AdaNet کار می‌کند. وقتی به این سیستم مجموعه‌ای از داده‌های برچسب خورده داده شد، لایه به لایه یک شبکهٔ عصبی می‌سازد و علاوه بر طراحی، هر لایه را برای اطمینان از بهبود عملکرد آزمون می‌کند. Adanet می‌تواند یک شبکه‌های عصبی که قادر به انجام یک وظیفه است و نیز یک شبکهٔ استاندارد و دست‌ساخته دوبرابر بزرگ‌تر بسازد. مهری معتقد است که نتیجهٔ امیدوارکننده‌ای است؛ زیرا بسیاری از شرکت‌ها تلاش می‌کنند که با منابع محدودشان، تا می‌توانند نرم‌افزار هوش مصنوعیِ قدرتمند را در وسایل تلفن‌همراه بگنجانند.

مهریار مهری

استاد علوم رایانه در مؤسسهٔ علوم ریاضی کورانت دانشگاه نیویورک متخصص یادگیری ماشین، نظریهٔ خودکاره‌ها و پردازش زبان

آسان کردن تولید و استقرار سیستم‌های پیچیدهٔ هوش مصنوعی ممکن است که معایبی داشته باشد. تحقیقات اخیر نشان می‌دهد که می‌توان به‌سادگی سیستم‌هایی ساخت که با هدف ترویج دیدگاه‌های متعصبانه باشند، مثل توهین‌ها یا جانبداری‌های نژادی و جنسیتی. مهری معتقد است که با کاهش امور خسته‌کنندهٔ دستی موردنیاز شبکه‌های عصبی، می‌توان این مشکلات را آشکار کرد و از آن‌ها اجتناب نمود. او می‌گوید «با خودکار کردن این امور مردم فرصت بیشتری می‌یابند تا به جنبه‌های دیگر مشکلات رسیدگی کنند.» اگر گوگل AutoMl را آن‌قدر خوب راه‌اندازی کند که بتواند یک ابزار عملی برای برنامه‌نویسان باشد، تأثیرات آن فرای خود شرکت احساس خواهد شد. پیچای اشاره کرده که می‌خواهد این ابزار را خارج‌از گوگل در دسترس عموم قرار بدهد. او گفت: «می‌خواهیم این ابزار دموکرات شود». این گفتهٔ او پژواک سخنان مغرورانه‌ای او است که از ارتقای خدمات هوش مصنوعی توسط واحد محاسبات ابری‌اش خبر می‌دهد.

چگونه اعتبار رهبری بیابیم؟

اعتبار اساس رهبری است. اما در قرن ۲۱ چگونه یک رهبر می تواند اعتبار پیدا کند؟

اعتبار کیفیتی است که از راستی، اعتمادپذیری، باورپذیری و متقاعدسازی حاصل می شود. در موضوع رهبری اعتبار پراهمیت است زیرا محرک زنجیره‌ای از واکنش هاست که تعیین می کنند چقدر رهبری شما بر پیروانتان، اعضای تیمتان، ذی‌نفعان، نتایج خروجی شان و نتایج تجاری اثرگذار است. اعتبار همیشه فراتر از الگووارهٔ اگر-آنگاه می‌رود؛ یا اعتبار دارید یا ندارید؛ درواقع میزان اعتبار جای سنجش دارد، نه بود و نبود آن. می توان آن را اندازه گرفت یا کمّی کرد، ناظرش بود و بهبودش داد. اساساً اعتبار سه جزء اصلی دارد: برآوردن انتظارات، اعتماد و عدالت. کم‌وبیشی اعتبار شما به ارزیابی دیگران از این سه جنبهٔ کلیدی رفتار شما بستگی دارد:

۱- چقدر اعضای تیم معتقدند که انتظاراتشان را برآورده کرده‌اید؛

۲- چقدر به شما اعتماد می کنند؛

۳- آیا معتقدند که شما رهبر عادلی هستید؟

اعتبار ریشه در راستی دارد؛ اینکه تا چه اندازه الگویی هستید از اصول، ارزش‌ها و رفتارهایی که به آن‌ها معتقدید. با پایبندی به حرف‌هایتان، زمانی که می توانید، و با پذیرش و کنش گری و اصلاح، زمانی که نمی توانید، راستی در شما رشد می کند و ریشه‌اش عمق می گیرد. اگر راستی را محور عملکردتان قرار بدهید، به قول‌هایتان عمل می کنید، اعتماد دیگران را جلب و حفظ می کنید، و رفتار شما منطقی و منصفانه است؛ و به این ترتیب، رهبری شما نمود می یابد.

برآوردن انتظارات

معیار سنجش برآوردن انتظارات میزان عمل کردن شماست به وعده‌ووعیدهایتان، انتظارات دیگران از شما و یا تعهدات شما به آن‌ها، آن گونه که درنظر آن‌هاست.

انتظارات، هرچند غیرمستقیم و نانوشته باشند، ماهیتی شدیداً تعهدی دارند، فراگیرند، و کیفیت روابط شما را تعیین می کنند. خلف‌وعده‌ها و نقض تعهدات یا انتظارات معمولاً عواقب منفی شدیدی بر اعتماد و درنتیجه، بر اعتبار دارند.

پیروان شما پنج انتظار از شما دارند:

- صداقت
- پیش‌اندیشی
- الهام‌بخشی
- تبحر
- هوشمندی

اگر می خواهید شما را رهبری استثنایی و قادر به اصلاحات تغییرساز ببینند، قطعاً در نگاه آن‌ها باید صادق باشید. صداقت مهم‌ترین خصیصهٔ رهبران ستوده است. علاوه‌بر برآوردن انتظارات، دو عنصر اصلی اعتباریابی اعتماد و عدالت هستند.

انصاف

انصاف یعنی چقدر شما در چشمان اعضای تیمتان منصف و بی‌طرف هستید. اینکه بی‌طرف باشید تأثیر بسیاری بر طرز نگرش آن‌ها به شما، و تمایلشان به پیروی از شما، و درنهایت، کارکردشان دارد. مشاهدهٔ عینی بی‌طرفی شما بر تعهدات اعضای تیم و تلاش فراتر از تعهد آن‌ها تأثیر می گذارد.

اعتماد

اعتماد برای اعتبار یافتن ضرورت دارد؛ معتمد بودن به‌معنای میزانی از اعتماد است که اعضای تیم و دیگر ذی‌نفعان به شما دارند. اعضای تیمی که رهبری معتمد دارند، باور دارند که رهبرشان هنگام نبودنشان منافع آن‌ها را درنظر می گیرد.

اعتماد ارزیابی تیم است از مقاصد، شخصیت، راستی و تبحر شما. رهبری موفق ذی‌نفعان و تحویل نتایج استثنایی قطعاً بدون مورد اعتماد بودن شدنی نیست.

اعتماد اعضای تیم زمانی جلب می‌شود که شما، زمانی که لازم است، محترم، بازخوردخواه (پذیرای فیدبک) و پذیرای اشتباهات خود باشید. اعتماد ضامن موفقیت تیم شماست. کلیدی است که در را به‌سوی ابتکار، چابکی و پرثمری بیشتر سازمان می گشاید.

حفظ اعتماد همواره از نیازهای رهبران کسب‌وکار است؛ زمانی می توانند اعتماد دیگران را حفظ کنند که در مرکز تمام عملکردشان افراد و مشتری‌ها باشند.

راهبردهای اعتمادسازی

همیشه:

- صادقانه و با احترام و پذیرش ارتباط برقرار کنید؛
- شایسته و کاردان باشید؛
- تشخیص درست خود را نشان بدهید؛
- اطمینان بدهید که می توان به شما اعتماد کرد؛
- به قول‌های خود جامهٔ عمل بپوشانید؛
- اطلاعات مهم را با شفافیت و با صراحت انتقال بدهید؛
- وقتی به مشکل برخورد می کنید دیگران را سرزنش نکنید.

برای سنجش میزان اعتماد پیروانتان به شما، بررسی کنید که به موارد فوق از ۰ تا ۱۰ چه امتیازی می‌دهند (۰ = کم، ۱۰ = زیاد).

راهکارهایی برای عدالت‌پروری

همیشه:

- به اعضای تیمتان فرصت ابراز دیدگاه‌هایشان را بدهید؛
- در تصمیم‌گیری به توافق برسید؛
- اختلافات و تعارضات را به‌سرعت و منصفانه رفع و حل‌وفصل کنید؛
- دیگران را در قدرت و نفوذ خود شریک کنید؛
- هیچ‌گاه از خود جانبداری نشان ندهید؛
- هیچ‌گاه از قدرت خود سؤاستفاده نکنید و تبعیض قائل نشوید.
- برای سنجش انصاف خود ببینید که پیروانتان به موارد فوق از ۰ تا ۱۰ چه امتیازی می‌دهند.

تأثیر اعتبار

همان‌طور که در ابتدا ذکر شد اعتبار ضمانتی است برای اثربخشی رهبری و این دو اثر متقابل دارند، مثل یک واکنش زنجیره‌ای. اعتبار بر مشارکت تیم و واکنش عاطفی‌شان به رهبری اثر می گذارد. این واکنش‌ها شامل تعهد و رضایت آن‌هاست. هرچه اعتبار بیشتری داشته باشید، مشارکت تیم شما بیشتر است. این اثربخشی متقابل بین مشارکت و عملکرد افراد نیز صادق است.

عملکرد افراد موجب تفکر خلاق و رفتار ابتکاری آن‌هاست (اینکه چقدر پذیرای تغییرات هستند و ایده‌های جدید را می‌پذیرند)، و چقدر پیش‌اندیشی می کنند و تلاش و وقت بیشتری صرف کارشان می کنند؛ منظور وظایف و انتظاراتی نیست که از آن‌ها توقع می‌رود، بلکه آن ایمان و عِرقی است که کوه‌ها را جابه‌جا می کند.

درآخر، عملکرد است که نتایج جمعی کسب‌وکار را تعیین می کند. به این دلیل، برای یک رهبر اعتبار حرف اول را می‌زند.

محمد بهنیا

کارشناس سئو و دیجیتال مارکتینگ

■ خودتان را معرفی بفرمایید.

سلام. محمد بهنیا هستم متولد سال ۱۳۶۵ و فارغ‌التحصیل رشتهٔ مهندسی فناوری اطلاعات؛ از سال ۸۳ در زمینهٔ فناوری اطلاعات سابقهٔ فعالیت دارم.

■ اصلی‌ترین زمینهٔ فعالیت شما چیست؟

بنده از سال ۱۳۹۰ به‌صورت تخصصی در حوزهٔ مدیریت و بهینه‌سازی سایت فعالیت می‌نمایم و در این مدت با هدف بررسی و کار بر روی کلمات کلیدی منتخب زمینه‌های شغلی مختلف، به بهینه‌سازی و مدیریت سایت مشغول بوده‌ام و نتایج بسیار خوبی کسب کرده‌ام.

■ شغل فعلی شما چیست؟

در حال حاضر فعالیت‌های بنده به دو قسمت تقسیم می گردد؛ ابتدا مدیریت درگاه‌های (پورتال) ملی، مانند همراه اول، سازمان فناوری اطلاعات ایران، سایت‌های مطرح خبری و فروشگاهی ...، و همچنین مدیریت دپارتمان بهینه‌سازی شرکت داده‌پرداز که در این دپارتمان خدمات سئو، دیجیتال مارکتینگ و کلیهٔ سرویس‌های مربوط به متقاضیان ارائه می‌شود.

چقدر برای اینکه در حوزهٔ فعالیت اصلی خود تأثیرگذار باشید تلاش کرده‌اید؟

تا جایی که به خاطر دارم، برای رسیدن به اهدافم همیشه تلاش زیادی کرده‌ام و بهای زیادی نیز برای آنها پرداخته‌ام؛ البته این بها به‌معنای صرفاً پرداخت هزینه نیست؛ بلکه گاهی برای استفاده از تجارب اساتید فن ساعت‌ها انتظار کشیده‌ام تا به تنها یک نکتهٔ کلیدی دست پیدا کنم؛ یا به خاطر دارم برای شرکت در سمینار بسیار مهمی که در خارج از کشور برگزار می‌شد، از موانع زیادی

عبور کردم تا بتوانم ۴ ساعت محتوای آموزشی باارزش دریافت کنم.

درهرصورت آنچه به خاطرم است، تلاش و تلاش و تلاش بوده و البته این نکته را هم فراموش نکنید؛ به دلیل اینکه برای خدمات سئو مستند معتبری وجود ندارد، تمامی تلاش‌ها به سعی‌وخطا برمی‌گردد و یک بهینه‌ساز خبره و متخصص تنها با کسب تجربه و تلاش بلندمدت در این عرصه قادر به فعالیت خواهد بود.

آیا کارها و پروژه‌هایی را که به‌انجام رسانده‌اید، موجب خوشحالی‌تان شده است؟

یکی از افتخارات و خوشحالی‌های من که در سال ۲۰۱۷ اتفاق افتاد، معرفی بنده به‌عنوان ۳ نفر از برترین متخصص‌های سئو در ایران ازنگاه مرجع تخصصی بهینه‌سازی بین‌المللی Woorank بوده است و امیدوارم بتوانم روزی در پروژه‌های بین‌المللی نام و پرچم کشورم را در زمینهٔ سئو و دیجیتال مارکتینگ به‌اهتزاز درآورم.

دقت داشته باشید زمانی که مدیریت و بهینه‌سازی یک وب‌سایت یا بهینه‌سازی شبکه‌های اجتماعی یک مجموعه به بنده یا تیم ما سپرده می‌شود، عموماً در شرایط نامناسبی قرار دارند و در این مرحله ما با صرف زمان، انجام فعالیت‌های موردعلاقهٔ موتورهای جست‌وجو و انجام هر آنچه جزء تجربه‌های کسب‌شده در سطح جهانی می‌باشد، یک فعالیت آنلاین را به موفقیت و حضور چشم‌گیری در موتورهای جست‌وجو و نمایش سرویس‌های کاربردی به کاربران می‌رسانیم و این برای کل تیم ما بهانهٔ خوبی برای جشن گرفتن خواهد بود.

در یک کلام شعار هلدینگ داده‌پرداز «موفقیت ما در گرو موفقیت مشتریان ما است»، باور قلبی تک‌تک اعضاء این تیم است.

■ برای اینکه علاقه‌مندان در SEO شروع به فعالیت کنند، چه کارهایی باید انجام دهند؛ آیا مدارک خاصی را دارید و به علاقه‌مندان پیشنهاد می‌کنید؟

همان‌طور که قبلاً هم گفتم، برای آموزش‌های سئو مستند دقیقی وجود ندارد؛ اشخاصی که آموزش می‌دهند با توجه به تجارب خود و مطالبی که از دیگران دریافت کرده‌اند آموزش می‌دهند؛ اما این آموزش‌ها بسیار خطرناک هستند؛ زیرا اگر از منبع معتبری اطلاعات دریافت نشود می‌تواند با انجام آنچه که در سایت‌ها اعمال می‌شود آسیب‌های جدی به SEO سایت وارد شود و در برخی فعالیت‌ها می‌تواند منجر به ازبین بردن تمامی جایگاه‌های فعلی و آتی سایت شود. برای همین آموزش‌هایی را که دریافت می‌کنید با حساسیت انتخاب کنید.

من به‌شخصه دارای مدارکی از جمله Google Digital garage – Google AdWords – Google Analytics – Woorank SEO Expert می‌باشم اما این مدارک ملاک انجام بهینه‌سازی تخصصی نیست و این تخصص در گذر زمان به‌وجود می‌آید. پس اگر علاقه‌مندان قصد دارند که به این عرصه پا بگذارند، از همین امروز شروع کنند.

■ چقدر با مدیریت پروژه آشنایی دارید؟

با توجه به ماهیت شغل بنده که مدیریت و بهینه‌سازی سایت می‌باشد، به‌طور موازی با مدیران پروژه فعالیت داشته ام و به جرئت اعلام میدارم زمانی که یک پروژه به صورت همزمان با مدیریت های چند جانبه پیش میرود موفق و قابل سنجش خواهد بود .

■ به‌نظر شما مدیریت پروژه چقدر می‌تواند در موفقیت یا شکست اهداف تأثیرگذار باشد؟

معتقد هستم در مجموعه‌های پروژه‌محور، مدیر پروژه پرچم‌دار اهداف یک سازمان یا یک پروژه است و سکان آن به دست این مدیر خواهد بود؛ پس به‌قول معروف کار را به کاردان بسپاریم.

به‌غیر از موضوع اصلی کسب‌وکارتان، در چه حوزه‌های دیگری فعالیت دارید؟ حتی اگر کار نباشد و جزء علاقه‌مندی‌های شخصی باشد. من عاشق خلبانی با RC و جزء انجمن مدلرهای تهرانی هستم. چندین هلیکوپتر ۴ و ۶ کانالـه دارم و تازگی شروع به برنامه‌ریزی برای خرید هواپیمای کنترلی کرده‌ام. همکاران من از شوخ‌طبعی من زیاد صحبت می‌کنند شاید دلیل آن علاقهٔ زیاد من به تئاترهای کمدی باشد؛ خیلی خوشحالم که این تفریحات و علاقه‌مندی‌های من با همسرم مشترک است و هردو از این برنامه‌ها لذت می‌بریم.

چقدر وقت شما را می‌گیرد؟

فقط آخر هفته‌ها یا تعطیلات رسمی حداکثر ۳ ساعت.

■ چند ساعت در طول روز مطالعه می‌کنید؟

مطالعات من عموماً دربارهٔ SEO و Digital Marketing است و در زمان‌های اضافه کتاب‌های موفقیت و مدیریت را مطالعه می‌کنم.

تصور می‌کنم که اگر مجموعهٔ زمان‌های مطالعه‌ام را محاسبه کنم، روزانه یک‌ساعت‌ونیم باشد.

■ این روزها در سطح جامعه با چالشی روبه‌رو هستیم که انتخاب شغل براساس درآمد نام دارد و نه علاقه. نظر شما در این‌باره چیست؟

قطعاً درآمد نقش مؤثری در پیشبرد علایق دارد؛ یعنی این دو المان، در کنار هم، علت‌ومعلول دوگانه‌اند؛ اگر یکی از المان‌ها فعالیت مؤثر نداشته باشد، نتیجهٔ نهائی و مؤثر دریافت نخواهد شد.

معتقدم قوی‌ترین علاقه‌مندی‌ها بدون کسب درآمدِ موردانتظار، در بلندمدت قدرت خود را از دست می‌دهند.

■ یک درددل کلی و عمومی با مردم یا هم‌صنف‌های خود؟

من یک نکته دارم که به‌عنوان درددل بیان می‌کنم؛ باور کنید بهینه‌سازان سایت جادوگر نیستند و معجزه نمی‌کنند! اکثر اشخاصی که قصد دارند از سرویس‌های بهینه‌سازی استفاده کنند، به‌دنبال معجزه هستند! برای مثال یک وب‌سایت تازه‌تأسیس، که از تاریخ ثبت دامنهٔ آن ۲ ماه نیز نگذشته، توقع دارند که بهینه‌ساز در یک بازهٔ یک‌ماهه، وب‌سایتشان در تک‌کلمهٔ ویژه‌ای به جایگاه نخست نتایج موتورهای جستجو دست پیدا کنند. این فعالیت نیاز به زمان دارد و باید با صبوری و برنامه‌ریزی بلندمدت نتایج مطلوب را کسب کرد. پس صبر پیشه کنید و اجازه دهید متخصصان در بلندمدت برای شما معجزه کنند.

■ سخن پایانی

در پایان، تشکر ویژه‌ای از مجلهٔ خط دید دارم که زحمت زیادی را برای پیشرفت کنترل پروژه کشور می‌کشد؛ من مطالب وب‌سایت خط دید، شبکه‌های اجتماعی و مجلهٔ چاپ‌شده این مجموعه را دنبال می‌کنم؛ به‌نظرم اطلاعات بسیار مفیدی را به مخاطبان منتقل می‌کنند. در بعضی پروژه‌های مشترک میزان فعالیت‌های این تیم را از نزدیک دیده‌ام و دقت و حساسیت این مجموعه برای ارائهٔ محتوای کاربردی و خاص لمس کرده‌ام؛ همین جا از مدیر محترم این مجموعه سپاس گزاری می‌کنم و برای این مجموعه موفقیت و توفیق روزافزون آرزو می‌کنم.

میلاد نکوفر

توسعه دهنده نرم افزاروب وموبایل (منبع: https://usesthis.ir)

■ کی هستی و داری چی کار می کنی؟

من میلاد هستم، میلاد نکوفر، فروردین شصت و پنج در اصفهان به دنیا اومدم و از اون موقع ساکن اصفهان هستم.

تا جایی که یادم میاد، درس و کلاس و مدرسه همیشه برای من کسل کننده بود، تنها درسی هم که برام جذابیت داشت، ریاضی بود، اون هم به خاطر چالشی که برام ایجاد می کرد. کلاً دوست داشتم هر روز چیزهای جدیدتری را تجربه کنم.

مدرسه راهنمایی می‌رفتم که با کامپیوتر آشنا شدم، از اون به بعد همه فکر و ذکرم یاد گرفتن کامپیوتر و بعد از اون اینترنت و کم کم برنامه نویسی شد. چی می‌تونست بیشتر از کامپیوتر هر روز برای من یک تجربه جدید رقم بزنه و بتونم چالش جدیدی توی اون پیدا کنم؟ کلاً قید درس و مدرسه را زده بودم و دیگه به ضرب و زور خانواده دبیرستان را توی رشته ریاضی و فیزیک تا دیپلم ادامه دادم.

برنامه نویسی را از همون زمان دبیرستان شروع کردم، اولین زبانی که با اون کد زدم، Perl بود، برای مووبل تایپ، اون زمان یک سیستم مدیریت محتوای ترند بین وبلاگ‌های فارسی زبان بود، چندتا پلاگین براش نوشتم، کمی هم توی بومی سازی اون کمک کردم. دوستان قدیمی تر احتمالاً کامیونیتی مووبل تایپ فارسی را به خاطر دارند. با کمک عزیز آشفته و شاهد رضامند اون کامیونیتی را راه اندازی کردیم و چند سالی هم بر پا بود و من از اواسط راه به خاطر مشغله از اون کامیونیتی جدا شدم.

مووبل تایپ در زمان خودش به علت اینکه صفحات وبلاگ را به صورت استاتیک ایجاد می کرد و به غیر از در زمان بازسازی وبلاگ منابعی مصرف نمی کرد ترند بود. اما همین موضوع باعث می شد که صفحات استاتیک ایجاد شده هیچ قابلیت خاصی نداشته باشه، این بود که ما در اون زمان صفحات را با پسوند PHP جنریت می کردیم و قسمت‌هایی که می‌خواستیم داینامیک باشه با کد PHP می‌نوشتیم، در اونجا بود که برای اولین بار کد نویسی PHP را شروع کردم.

همون وقت‌ها بود که ای جکس هم که اون روزا ما بهش می گفتیم آژاکس ترند شده بود و همه سایت آژاکسی می‌خواستن. توی وبلاگ یکی از توسعه دهنده‌های خارجی مووبل تایپ با prototype.js آشنا شدم که یه جورایی پدر jQuery به حساب میاد. اون موقع‌ها با ترکیب مووبل تایپ و PHP و این prototype.js کارای عجیب غریبی می کردیم و یه جورایی استارت ورود من بود به اون چیزی که الان بهش می گن فرانت‌اند دولوپمنت.

اون موقع‌ها بزرگ‌ترین مشکل، زیر ساخت لازم برای ران کردن نرم افزارها روی سرور بود، که برای زبانی مثل Ruby عملاً زیر ساختی در دسترس نبود، یه مدت هم سعی کردم با Ruby کار کنم، با Ruby از طریق عزیز آشفته آشنا شدم، اون موقع ها روی بالاترین کار می کرد، اما در نهایت با توجه به شرایط موجود، PHP را به عنوان زبان برنامه‌نویسی اصلی برای خودم انتخاب کردم و توسعه نرم افزار در سمت وب را ادامه دادم، تا این اواخر و یکی دو سال گذشته که به سمت برنامه‌نویسی موبایل برای Android و کمی هم بفهمی، نفهمی برای iOS روی آوردم.

بعد از جدا شدن از کامیونیتی مووبل تایپ، تمرکزم روی PHP بیشتر شد، از اونجایی که به دنبال چالش‌های جدید بودم به سمت وردپرس که با PHP نوشته شده بود جذب شدم، یه مدت کوتاهی با اون کار کردم و چند تا پلاگین هم برای اون نوشتم که تا چند سال پیش روی سایت پلاگین های وردپرس که یک سایت جداگانه بود قابل دسترس بود. اون زمان وردپرس کامیونیتی به این منسجمی نداشت و کامیونیتی فارسی زبان اون هم تازه شکل گرفته بود. اما وردپرس هم پس از مدتی دیگه جداب نبود.

اون موقع زمانی بود که بحث فریم‌ورک‌های PHP تازه‌ترند شده بود و Zend و CakePHP جزو فریم ورک های محبوب بودن، من اون زمان CakePHP را انتخاب کردم و این اولین فریم ورکی بود که با اون کار می کردم. یکی دو سالی با CakePHP کار می کردم و با صالح سوزنچی عزیز و یک نفر دیگه از دوستان که اسمش خاطرم نیست و اون زمان دامین دات آی آر CakePHP را ثبت کرده بود، یک کامیونیتی برای CakePHP راه‌اندازی کردیم و باز بعد از مدتی به طبع تنوع طلبی، از CakePHP هم خسته شدم و از اون کامیونیتی جدا شدم.

از همون اوایل، هر از چندی حسب فرصتی که پیش می‌اومد، به زبان‌ها و تکنولوژی‌ها و فریم‌ورک‌های

مختلف ناخنکی می‌زنم، بگی، نگی، با بیشتر زبان‌ها کدزدم یا حداقل با سینتکس اون‌ها آشنایی دارم، کلاً خودم را یه گیک توی حوضه توسعه نرم افزار می‌دونم، اخیراً هم دنبال این هستم سری به زبان Go بزنم.

اما چطور شد که وارد بحث توسعه نرم افزار موبایل شدم، راستش از سه، چهار سال پیش، پیگیر اخبار حوزه توسعه موبایل بودم، حتی کم و بیش ناخنکی هم به Android و iOS زده بودم، اما حدوداً دو سال پیش به یک تیم بازی سازی موبایل جوین شدم، از اونجا بود که تصمیم گرفتم به صورت جدی برنامه‌نویسی موبایل را هم شروع کنم، به خاطر تجربه‌ای که در برنامه نویسی داشتم و اطلاعات جسته گریخته قبلی ظرف دو، سه هفته تونستم سر از کار در بیارم تا جایی که روی پروژه اصلی تیم مشغول به کار شدم و در کنار قسمت بک‌اند و API اپ روی خود اپ هم شروع به کار کردم. کار با اون تیم باعث شد به بازی سازی هم علاقه‌مند بشم و ناخنکی هم به گیم‌انجین Unity3D بزنم، اما خوب همکاری من با اون تیم هم زیاد به طولی نینجامید. در همون دوره دبیرستان بود که با لینوکس آشنا شدم و به طبع خصیصه چالش طلبی خودم و علاقه‌ای که به یادگیری و تجربه‌های جدید داشتم بعد از مدتی به اون مهاجرت کردم. اون موقع ها از اعضای فعال اوبونتوی فارسی بودم و اون زمان در اصفهان سی‌دی‌های اوبونتو را بین متقاضیان توزیع می‌کردم. اما بعد از مدتی از اون کامیونیتی هم جدا شدم.

بعد از دیپلم، یه مدت، باز هم به ضرب و زور خانواده در بازار و در فروشگاه پدرم به عنوان فروشنده مشغول به کار شدم، اما مدت زیادی دووم نیاوردم. از همون زمان دبیرستان، کار به عنوان یک برنامه‌نویس فریلنس را شروع کرده بودم، الان سال‌ها از اون زمان می‌گذره، تجربیاتی در کار با شرکت‌ها و تیم‌های مختلف در داخل یا به صورت ریموت در خارج کشور به دست آوردم، با پلتفرم‌های مختلف کار کردم و کار با تکنولوژی‌های مختلف را یاد گرفتم، آینه که اینروزها خودم را یک توسعه دهنده نرم‌افزار یا همون دولوپر، یا حتی با کمی احتیاط بیشتر فول استک دولوپر می‌دونم.

در همون زمانی که با موبل تایپ کار می‌کردم، یک سرویس میزبانی به اسم موبل هاست راه‌اندازی کردم که بعضی‌ها اینطرف و اونطرف هنوز به خاطر دارن، امکان راه اندازی وبلاگ اختصاصی برای افراد با موبل تایپ را فراهم می‌کردم. بعد از چند سال این خدمات تا ارائه سرورهای مجازی و اختصاصی گسترش پیدا کرد، اما با بالا رفتن قیمت دلار حاشیه سود به شدت کاهش پیدا کرد و از اونجایی که درگیر پروژه‌های نرم افزاری هم بودم و زمان کافی برای پشتیبانی اون سرویس را هم نداشتم، به مرور زمان کمرنگ شد و یکی دو سال پیش اون سرویس را هم دان کردم. تجربه خوبی بود و حدود شش الی هفت سال ادامه داشت و فرصت خوبی بود که اطلاعاتی در زمینه مدیریت و امنیت سرورهای لینوکس به دست بیارم.

هنوز ترجیح میدم یک توسعه دهنده و فریلنسر باشم تا هر چیز دیگه، اما خوب کم‌کم با بالا رفتن سن می‌فهمید که دیگه زمان شما داره می‌گذره و نسل جدید بنا به به اصل تکامل با سرعت زیادی داره از مرزهایی که شما پشت سر گذاشتید عبور می‌کنه، آینه که این روزها برنامه آینه که دانش خودم را توی حوزه مدیریت و بازاریابی در زمینه پروژه‌های نرم افزاری، استارتاپ‌ها و کارآفرینی گسترش بدم و از تجربه‌هایی که به دست آوردم در این زمینه استفاده کنم. همچنین در راستای این مهم، در دو سال گذشته سعی کردم دایره ارتباطی خودم را با شرکت در رویدادها و گردهم‌آیی‌های مختلف در شهرهای مختلف گسترش بدم.

لازم به ذکر هست که، موارد بالا مربوط به یک بازه زمانی حدود سیزده ساله هست، سعی کردم زیاد وارد جزییات نشم که مطلب طولانی نشه اما باز هم فکر می‌کنم طولانی شد. در بالا جاهایی به لفظ ناخنک زدن اشاره کردم، وقتی از این لفظ استفاده می‌کنم مواردی مد نظرم هس که در حد آشنایی به سینتکس یا کمی بیشتر با اون‌ها کار کردم ولی هنوز فرصتی پیش نیومده که پروژه‌ای با اون ها پیاده سازی کنم.

خیلی وقته ننوشتم، اما اخیراً باز شروع کردم و سعی می‌کنم در وبلاگم، مجله هایو و یکی دو جای دیگه تجربیات خودم را به اشتراک بزارم.

اگر اصفهان باشید، من را توی یکی از جلسات همفکر یا ایمدا و بعضی وقت‌ها هم لاگ می‌تونید پیدا کنید. توی شبکه‌های مجازی مثل لینکدین، گیت‌هاب، توئیتر، فیس‌بوک، گوگل‌پلاس و یا

مصاحبه با ویکتور تامپسون

برنامه‌نویس (آکادمی Caper)

توسعه‌دهندهٔ بازی، لینوکس، مک، ویندوز

■ خودتان را معرفی کنید؛ چه می‌کنید؟

من ویکتور تامپسون هستم، یک برنامه‌نویسِ مستقل بازی. اخیراً محفظه و دستگاه‌های داخلی بازی‌های رایانه‌ای می‌سازم که در مکان‌های عمومی نصب و نمایش داده می‌شوند.

درحال‌حاضر برنامه‌نویسی موتورِ West Of Loathing را- که تازگی‌ها عرضه شده- انجام می‌دهم؛ West of Loathing یک بازی کمدی نقش‌آفرینی (RPG) است. کمی وحشت کرده‌ام؛ چون در حرفهٔ قبلی‌ام در تیم‌های برنامه‌نویس‌ها کار می‌کردم، اما وقتی به این کار آمده‌ام فقط خودم باید کد موتور را برای بازی‌هایی که می‌سازم بنویسم یا

اشکال‌زدایی (دیباگینگ) کنم.

من و Zach Johnson یک کابینِ عمومی و رایگان چندبازیکنی (مولتی‌گیم) ساختیم. هدف از ساخت آن را بیشتر نمایش بازی‌های محلی و پروژه‌هایی است که ناتمام مانده‌اند یا برای عرضه‌های تجاری نیستند. در اینجا فرصت گفتن تمام انگیزه‌هایمان برای ساخت این کابین نیست.

من تقریباً نصف تیمِ سازندهٔ Please Stand by هستم و نصف دیگر جری بلیک، یک بازی‌ساز، است! این بازی مجموعه‌ای از تعامل‌هایی است که با استفاده از یک کنترل تلوزیون قدیمیِ Vintage -که بازی را نشان می‌دهد- انجام می‌شود.

■ از چه سخت‌افزاری استفاده می‌کنید؟

اخیراً بر یکی از رایانه‌های اصلی‌ام برای کار توسعه، که یک MacBook Pro ۱۳-اینچی تولید سال ۲۰۱۳ است، کدنویسی می‌کنم. برای برخی از نرم‌افزارها توانش کم می‌آید و برای همین نمی‌گذارم نیازهای پردازشی و گرافیکیِ West of Loathing بالاتر از حداقلِ نیازمندی‌های سیستممان شود. ضمناً خوشحالم که می‌توانم در یک

کول‌پشتی حملش کنم یا در یک میزکار مشترک، که معمولاً پر از رایانه است، بگذارمش.

در خانه لب‌تاپ را به یک صفحه‌نمایش Thunderbolt اپل ۲۷-اینچی و یک صفحه‌کلید تمام‌اندازه و موشوارهٔ (موس) مایکروسافت وصل می‌کنم. سال‌هاست که به این سیستم رومیزی عادت کرده‌ام و هنوز برایم راحت‌ترین سیستم است.

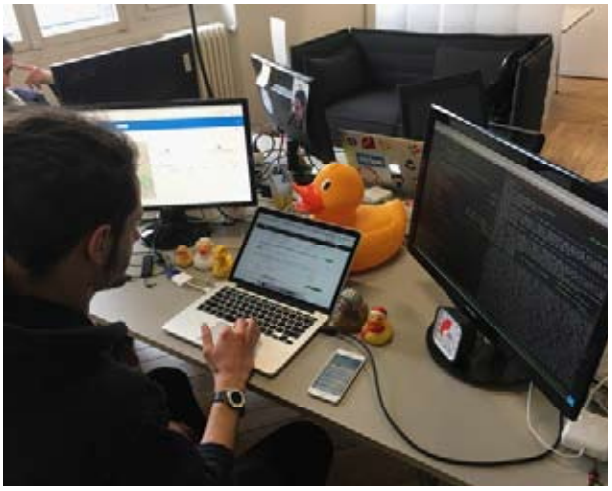
صوت کیفیت‌بالا خیلی برای کار من مهم نیست اما همیشه برای آزمون میکس‌ها و نشنیدن صداهای چاپگر یا خیابان زیر پنجره‌ام از هدفونی که روی میزکارم است استفاده می‌کنم. این هدفون آخرم آن‌قدر قیمتش پایین است که به خودشان زحمت نوشتن شمارهٔ مدل را بر آن نداده‌اند.

برای آزمون بازی‌ها بر دیگر پلتفرم‌ها یک رایانهٔ شخصی برجی (tower PC) دارم که سال ۲۰۰۸ یا ۲۰۰۹ ساختمش. خیلی باکیفیت نبود و الآن سن‌وسالش را نشان می‌دهد. ویندوز ۷ بر آن نصب است اما با چند حافظهٔ استیک یواس‌بی (USB memory Sticks) لینوکس را در آن بارگذاری (boot) کردم.

وقتی یک پروژهٔ فیزیکی می‌سازم (آخری‌اش یک دستگاه آرکید برای Joggernauts و یک قفسهٔ خارجی است) با مجموعی از ابزار دستی ساده کار می‌کنم، مثل چکش و اسکنه و یک رندهٔ صاف‌کاری Stanely و ابزار برقی مختلفی مثل دریل یا میز حمل‌پذیر ارّه، و یک سمباته‌کش. این روزها از کیفیت ابزار دستی باطری‌دارم خیلی خوشحالم؛ پرکابردترین ابزارم یک impact driver هیتاچی است. بسیار مفید است و روزی یک‌بار باطری‌اش شارژ می‌شود و پنج سال پیش نصف قیمت دریل کابلی‌ام خریدمش.

برای نوشتن برنامه‌ریزی‌های خُرد پروژه و طراحی‌های سادهٔ قالب‌ها یک تخته‌سفید نزدیک میزکارم دارم. کارهای زیادی برای حل مسئله و مدیریت وظایف بر تخته‌سفیدم انجام می‌دهم زیرا فهمیده‌ام کارها به سرعتی که در ذهنم یا در نرم‌افزارها برآورد می‌کنم پیش نمی‌روند. فکر کنم شاید به خاطر «اشکال‌زدایی اردک پلاستیکی» است و اینکه با دست نوشتن، سرعت افکارم را پایین می‌آورد تا هر مرحله را به‌تنهایی محاسبه کنم.

در مهندسی نرم‌افزار rubber duck debugging یک روش اشکال‌زدایی کد است که اسم آن از کتاب «برنامه‌نویس عملگر» گرفته شده است؛ در این کتاب یک برنامه‌نویس با خودش یک اردک پلاستیکی حمل می‌کند و هنگام دیباگ کردن کد، کد را مجبور می‌کند که خط‌به‌خط خودش را به اردک توضیح بدهد. به همین ترتیب، بسیاری از برنامه‌نویس‌ها با توضیح کد به کسی که هیچ‌چیزی از برنامه‌نویسی نمی‌داند، می‌توانند اشکال را بیابند.



■ از چه نرم‌افزاری استفاده می‌کنید؟

تقریباً همه محاسباتم را در OS X انجام می‌دهم. چند سال گذشته برای West of Loathing و دیگر پروژه‌ها از Unity استفاده کرده‌ام. در پنج سال گذشته کل سیستم همین بود، یک دستگاه ویندوز و چند موتور مختلف خودساخته. کدهای سطح‌پایین خودم را برای پروژه‌های شخصی هرجا شد می‌نویسم، اما وقتی عهده‌دار پروژه‌های دیگران هستم، وجدانی نیست که وقت اضافه برای کار خودم صرف کنم. برای ویرایش کد تقریباً فقط از VS Code مایکروسافت استفاده می‌کنم. ترکیب VS Code با موتور Unity واقعاً برای من خوب کار می‌کند، چون هیچ‌وقت با MonoDevelop، ویرایشگر همراه Unity، راحت نیستم. وقتی نمی‌توانم از VS Code استفاده کنم (مثلاً وقتی که نیاز دارم فایل‌های غول‌پیکر JSON را جست‌وجو کنم) با Sublime Text ۲ کار می‌کنم.

فقط چند دقیقه در روز در Slack در دسترس نیستم. با تیم West of Loathing، یا دیگر توسعه‌دهنده‌های محلی بازی و چند گروه دیگر ارتباط برقرار می‌کنم. زیرا هردو از Slack برای کارمان زیاد استفاده می‌کنیم، من و همکارم یک تیم ساخته‌ایم که طی روز از آن برای ارتباط استفاده می‌کنیم و فایل‌ها را جابه‌جا می‌کنیم. Slcak از زمان IRC تا امروز، بهترین است. معمولاً در نرم‌افزارم طراحی فیزیکی انجام نمی‌دهم اما از Sketchup استفاده می‌کنم، مثلاً برای کشیدن. معمولاً برای اینکه اندازهٔ نسبی پروژه و قطعات آن را بدانیم از آن استفاده می‌کنم. هنوز برداشتن یک مداد و خط‌کش برای کشیدن چیزی یا تغییری در یک طراحی بسیار حس بهتری دارد؛ بیشتر طراحی‌هایم در ذهنم است.

■ سیستم رؤیایی شما چیست؟

برای کار نرم‌افزاری دوست دارم یک دفتر کار داشته باشم که ۵۰ درصد ضدصدا باشد (نمی‌دانم تا این دسیبل ممکن است یا نه). فکر می‌کنم که سیستم میزکارم خوب است؛ خواسته‌هایم برای محیط توسعه‌ام آن‌قدر تغییر می‌کند که مطمئن نیستم سیستمی وجود داشته باشد که بتواند محیط توسعهٔ بهتری به من بدهد. بدم نمی‌آید یک لب‌تاپ جدید داشته باشم که بتواند سرعت مراحل کدنویسی را حین اشکال‌زدایی یک برنامهٔ کاربردی Unity بالا ببرد. برای سازه‌های فیزیکی دوست دارم که یک ماشین بزرگ‌تر داشته باشم، احتمالاً با یک فضای بیشتر. دستگاه‌های اصلی موردعلاقه‌ام شاید یک ارّهٔ نواری، یک دریل فشاری، و یک رندهٔ نجاری ضخیم باشد. یک سیستم CNC تولیدومصرف‌کننده (prosumer) مثل ShopBot نیز می‌خواهم. شاید خوب بود که یک محیط الکترونیکی می‌ساختم که بتوانم از قطعات ارزان‌قیمت وسایل موردنظرم را بسازم؛ بهتر بود از اینکه وسایلی بخرم که اهداف عمومی دارند و فقط برای بخشی از نیازهای من مناسب است. جدایی از ابزار و برنامه‌های کاربردی و فرایندهای که از آن‌ها استفاده می‌کنم این تنوع کار روزانه‌ام است که زندگی را دلنشین‌تر می‌کند. اخیراً حدود ۸۰ درصد از زمان کارم را در لب‌تاپ انجام دادم و ۱۰ درصد از از زمان بر تخته‌سفید یا پای تلفن و ۵ درصد برای خرید و ۵ درصد برای پرسه زدن در حین صحبت کردن با خودم صرف شده است. دوست دارم زمان کار با لب‌تاپ را به ۶۰ درصد کاهش بدهم و زمان خرید را به ۲۵ درصد برسانم.

مت مارکوئیز طراح وب و توسعه دهنده

■ خودتان را معرفی کنید؛ که هستید و چه می‌کنید؟

من مت مارکوئیز هستم؛ خیلی مرموز است! از دبیرستان من را ویلتو می‌نامند. در Bocoup وب‌گاه‌های پاسخگو، دسترس‌پذیر و سریع می‌سازم. گاهی سخنرانی می‌کنم، کتاب «JavaScript برای طراحان وب» و چند ماه بعد از آن کتاب A Book Apart را تألیف کرده‌ام.

غیر از کار سرگرمی‌های زیادی دارم. من یک چوب‌کار نیوانگلندی به دنیا آمدم و پرورش یافتم. از کسانی که بوکس بلد نیستند، بهتر بوکس بازی می‌کنم، اما نه خیلی زیاد! آشپزی می‌کنم، بیشتر فرانسوی، و دائماً در حال تعمیر موتورسیکلتم، Triumph Bonneville 78 هستم.

■ از چه سخت‌افزاری استفاده می‌کنید؟

ابزار خیلی زیادی دارم.

خط‌مشی خانواده ما این بوده که «ارزان‌ترین نوع یک ابزار را بخر و فقط اگر شکست، یک جدید بخر». خیلی راحت پول صرف چاقوهایی که تیغشان خوب نیست یا رنده‌های چوب که در انبار خانه مانده می‌کنم.

مطمئنم بیش از هر کسی در ناحیه بوستون پیچ‌گوشتی دارم و حتماً جزء ۵ نفری هستم که بیشترین انبردست زنگ‌زده را دارند. یک اره‌ی دستی Disston D-8 دارم که بهتر از توانایی‌اش برایم کار می‌کند، بنده‌ی خدا حدود ۱۵۰ سالی سن دارد. تعداد زیادی رنده‌های نجاری قدیمی Stanley و یک کیف چرمی ابزار تخریب‌ناپذیر دارم که الآن استفاده چندانی از آن‌ها نمی‌شود. میز اره من یک میز کوچک و مناسب Bosch است؛ وقت زیادی برای کار کردن با آن ندارم. اره نواری‌ام را که با تخفیف از آمازون دات کام خریده‌ام، هنوز راه درازی دارد تا دست از سرش بردارم اما خیلی هم‌زمان ندارد. چند سال پیش یک جت داشتم، اما الآن دیگر یک چوب‌کار آپارتمانی هستم، در یک گاراژ کوچک. دلم برای اره متری DeWalt قدیمی‌ام تنگ شده. الآن یک Rigid دارم، یک کم لگد میزند؛ این اره‌های لیزی هم که احمقانه‌اند، اما خوب می‌برند. Diablo هر چیزی را می‌برد!

با دستکش‌های Cleto Reyes بوکس بازی می‌کنم (شک نکنید که قرمز است!) آن‌قدر راحت‌اند که هر جفت دستکش دیگری را که استفاده کنم انگار دست‌هایم فومی و احمقانه می‌شوند؛ یک محافظ دهن سفارشی هم دارم که آواره‌های من را که مطمئناً زیاد مشت می‌خورد حفظ می‌کند. این شعارم که «از چیزهای ارزان استفاده کن تا زمانی که خراب بشوند» از کفش‌های بوکسم معلوم است، آدیداس یا یک چیزی مثل آن، بدک نیستند.

در آشپزخانه یک همزن ایستاده قدیمی زنگ‌زده‌ی KitchenAid دارم (البته قرمز رنگ!) که تا وقتی که زیادنان نمی‌پختم، از آن مطمئن بودم؛ قصد دارم با خودم آن را به گور ببرم. ست چاقو ندارم اما یک چاقوی سرآشپزی Shun Classic ۸" اعصای دستم است و یک چاقوی فولادی کربنی دارم که از آن برای کارهای سنگین‌تر استفاده می‌کنم؛ تیغش از Hock Tools است که تیغ‌های رنده چوب‌سازی فوق‌العاده‌ای می‌سازد. یک Shun Santoku "۲/۱ ۵ دارم که با آن بدرفتاری می‌شده و با ۳۰ دلار آن را به‌عنوان یک چاقوی مفید گرفتم و یک چاقوی جداکننده Shun Sora که به‌ندرت از آن استفاده می‌کنم. موفق شدم که یک مجموعه تحسین‌برانگیزی از قابلمه‌ها و ماهی‌تابه‌ها را جمع کنم که اکثراً روکش شده‌اند و روکش آهنی و سایزها و شکل‌های مختلفی دارند که هر کس ممکن است نیاز داشته باشد (اگر بخواهم به آن امتیازی بدهم ۵ است). همین تازگی‌ها فهمیدم که چگونه با یکی از ماهیتابه‌های سبک‌وزن روکش‌دار آهنی‌ام کار کنم که ممکن است از این به بعد اعصای دستم شود. گرما را کامل، حفظ نمی‌کنند، اما باور کنید که تکان‌دادنش خیلی آسان‌تر است. اخیراً یک ابزار کیسه‌پزی خریدم. واقعاً خریدم! خیلی خیلی به آن بدبینم، بیشتر به دلیل اینکه از خطر نفوذ کامل فنّاوری در غذاهایم می‌ترسم. بیشتر یک آدم اهل آهن و آتش و دست‌به‌کارشدنم تا آلومینیوم و آب و صبر کردن. یک ساعت در حمام و سرخ کردن چیزی روی شعله‌های آتش خوب من را سرحال می‌آورد! در کار، یک MacBook Pro 15 اینچی دارم که بیش‌ازحد قدیمی شده؛ خوب کار می‌کند؛ مدل‌های جدیدش هم خیلی چیز جالبی ندارند. راز پنهانم این است که فقط از صفحه کلیدهای توکار و لمسی استفاده می‌کنم؛ هیچ موس خارجی یا صفحه کلیدی ندارم، حتی وقتی که طراحی می‌کنم. نمی‌دانم واقعاً چه شد که این‌طوری شدم، اما حالا که همین است! اخیراً یک میز ایستا از برخی از پالت‌هایی که اطراف اداره پیدا کردیم ساختم. خب شبیه Pinterest نیست. طبق سنتم تا زمانی که نشکند از آن استفاده می‌کنم. مونیتور دومم نیز خودِ آشغال است! اما همیشه Term را تمام‌صفحه نشان می‌دهد؛ تقریباً برای بررسی مشکلات مربوط به تضادهای رنگی خوب است.

■ از چه نرم‌افزاری استفاده می‌کنید؟

فوت کوزه‌گری که قابل گفتن نیست، برای کد از Sublime Text استفاده می‌کنم. Canvas ابزار بسیار خوبی است؛ اما هنوز دلم برای Editorially تنگ می‌شود. در کار طراحی، کمی از دوران عقبم اما چند ماه پیش به Sketch پیوستم. تصمیم داشتم در Adobe suite گشتی بزنم، البته به غیر از بخش مزخرف "save for the web" در فتوشاپ.

احتمالاً در 12Rounds بیشتر از هر برنامه کاربردی دیگری در گوشی‌ام کار کردم. یکی از همین روزها کمی از زمان کدنویسی متن‌بازم را برای ساختن یک تایمر بوکس برای خودم صرف می‌کنم، چرا که نه؟

■ ستاپ رؤیاهای شما چیست؟

میز آشپزخانه. میز آشپزخانه. میز آشپزخانه! وقتی چشم‌هایم را می‌بندم و جرئت دیدن رؤیای خانه بی‌نقصم را می‌کنم، فقط میزهای آشپزخانه را می‌بینم. میزهای بریدن گوشت، سنگ‌های soapstone و مرمر... خیلی گیر نمی‌دهم. میزهای آشپزی در هر جای اتاق کافی است. LMM و من حدود سه فوت فضای کانتر را گرفتیم و یک میز IKEA در آشپزخانه آپارتمانمان واقعاً دارد ما را می‌کشد! تمام چوب‌کاری‌ها و دست‌کاری موتورسیکلت را در یک گاراژ دو اتومبیلی بدون حرارت انجام می‌دهم و این وضع خوبی برای شرایط یخبندان اینجا در شمال نیست (نیوانگلند). فقط گرما! گرما را دوست دارم. از ۷۲ درجه سانتی گراد در سال حرف نمی‌زنم، فقط آن‌قدر گرم که به غیر از نصف تابستان و یک هفته بعد از آن، چسب چوبم مثل گچ یخ نزند. منظورم این است که ابزار جدید نباشد قاتی نمی‌کنم اما فقط گرما می‌خواهم!

در کار، همه چیز خوب است. لپ‌تاپ داشته باش و سفر کن! حداقل در کار کم‌توقع هستم.

جِف دین توسعه‌دهنده، محقق (گوگل)

فوریه ۲۸، ۲۰۱۷

■ خودتان را معرفی کنید؛ که هستید و چه می‌کنید؟

من جِف دین، یکی از دو عضو ارشد گوگل در شرکت گوگل هستم. به تحقیق در مورد هوش مصنوعی و سیستم‌های رایانه‌ای می‌پردازم؛ نرم‌افزار می‌نویسم و بر سیستم‌های مختلفی مثل سیستم‌های تبلیغات گوگل، جستجوی گوگل، ترجمه گوگل، Bigtable، MapReduce، Protocol Buffers، Spanner و Snappy، LevelDB (a.k.a. Zippy) و TensorFlow و غیره کار می‌کنم.

در حال حاضر Google Brain project و گروه تحقیقاتی آموزش عمیق گوگل را هدایت می‌کنم که در آن بر دستگاه‌های مربوط به یادگیری ماشین، خصوصاً آموزش عمیق کار می‌کنیم و من به هدایت تحقیقات در مورد آموزش ماشین و دیگر حوزه‌های تحقیقات کاربردیِ در حال ظهور که یادگیری ماشین در آن‌ها ضروری است می‌پردازم؛ از جمله‌ی این حوزه‌ها عبارت‌اند از: سلامت و بهداشت، رباتیک، ادراک و فهم زبان طبیعی و توسعه‌ی پیوسته TensorFlow که یک پلتفرم کلیدی برای تحقیق درباره‌ی یادگیری ماشین و توسعه‌ی سیستم‌های یادگیری ماشین و ویژگی‌هاست. می‌توانید با مشاهده فهرست اخیر انتشارات گروه Brain، در مورد تحقیقات ما بیشتر بدانید.

■ از چه سخت‌افزاری استفاده می‌کند؟

در محل کار معمولاً از HP Z620 Workstation با کارت NVidia GPU و لینکوس که فقط در یک مانیتور ۳۰اینچی اجرا می‌شود استفاده می‌کنم. برای کاربردهای حمل‌پذیر گفتاری، تکیه‌ام اول بر یک MacBook Air و گاهی بر Chromebook Pixel است. یک Nexus 5X Android phone دارم و همچنین از یک Nexus 9 tablet برای جستجوی وب و خواندن کتاب‌های الکترونیک و برگه‌های فنی استفاده می‌کنم. از یک میز قابل‌تنظیم برای نشستن و ایستادن استفاده می‌کنم و به قهوه‌ساز اسپرسوی میکروکیچن که نزدیکم قرار دارد وابسته‌ام.

■ و از چه نرم‌افزاری استفاده می‌کنید؟

برای نوشتن کد از emac و سیستم ساخت توزیع‌شده‌ی داخلی گوگل (که یک نسخه آن مثل Bazel متن‌باز بود) و سیستم کنترل نسخه‌ی خودمان، به‌علاوه ابزار جستجوی داخلی کد گوگل استفاده می‌کنم که با آن می‌توانم به‌سرعت در کل منبع کد گوگل جستجو کنم (بسیار شبیه محصول Google Code Search است).

بیشتر از همه در C++ کد می‌نویسم؛ هرچند که سال‌ها است مقدار زیادی کد در Java، Python، assembly Perl, Self, Cecil, x۸۶ و Pascal نوشته‌ام. برای توضیح‌گذاری و همکاری برای اسناد فنی داخلی، از Google Docs استفاده می‌کنم. برای مقالات فنی، من و مؤلفان همکارم معمولاً از LaTeX استفاده می‌کنیم. برای ارائه دادن، تقریباً فقط از Google Slides استفاده می‌کنیم و برای کنترل ایمیل، از Gmail با مقدار زیادی فیلتر استفاده می‌کنم که به‌طور خودکار فایل و پیام برچسب را فیلتر می‌کند (هر روز ۱۴۰۰ پیام ایمیلی می‌گیرم، پس باید زمانم را مدیریت کنم). همچنین از مرورگر Chrome استفاده می‌کنم و جستجوهای وب را با گوگل انجام می‌دهم.

■ ستاپ رؤیاهای شما چگونه است؟

تقریباً آن را دارم. بیشتر از تحقیق کردن و نوشتن نرم‌افزار با همکاران فوق‌العاده لذت می‌برم؛ مخصوصاً وقتی افراد زیادی از نتایج کار ما استفاده کنند؛ البته یک لپ‌تاپ سبک‌تر با صفحه‌نمایش بزرگ‌تر فوق‌العاده بود! چشم‌انداز اقیانوس هم زیبا است!

شِلی آلُن

توسعه‌دهندهٔ بازی، سازندهٔ دوبازیِ Partyrs و Sputnik Eyes

■ خودتان را معرفی کنید، چه می‌کنید؟

شِلی آلُن (Shelly Alon) هستم؛ ۲۵ سال دارم و اهـل هامبـورگ هسـتم. طراح و توسـعه‌دهندهٔ مسـتقلم. در دانشـگاه علوم کاربردی هامبورگ (HAW Hamburg) طراحـی خوانـده‌ام و درحال‌حاضر کارشناسـی ارشـد تصویرسـازی می‌خوانـم. به‌طور تفریحـی تایپوگرافی، طراحی ویرایشـی، عکاسی و پویانمایی می‌کنم اما در سه یا چهار سال گذشته فقط بازی ساخته‌ام.

بـا چنـد دوسـت این کار را شـروع کـردم و یک چالـش کلیـک و نقطه رمزی با نام Olav & the Lute را سـاختیم. سـپس توسـعه را خودم شروع کردم، بیشـتر، بازی‌های پازلی برای تلفن همراه. مثلاً Partyrs یک بازی پازلی اسـت که باید در آن یـک مهمانـی ترتیب دهید و تمام مهمان‌ها را شـاد نگـه داریـد. Sputnik Eyes نیـز دربارهٔ کاوش دیگر سـیاره‌ها با ربات‌های تک‌چشـمی بامزه است. به‌تنهایی بازی‌ها را توسعه می‌دهم و تمام کد را خودم می‌نویسم؛ کار هنری‌اش را هم خودم می‌کنم. برای موسـیقی و صدا دوست‌های واقعاً بااستعدادی دارم که کمکم می‌کنند.

آخریـن بـازی‌ام با نـام Glitchskier یک بازی هیجانی آرکید اسـت که از یک بازی دزد دریایی در ویندوز ۹۵ ساخته شده است. البته خیلی از آن راضـی نیسـتم، اما تابه‌حـال موفق‌ترین بازی‌ام بـوده اسـت. این بـازی برندهٔ جایـزهٔ بازی‌های ویدئویی آلمان در بخش بهترین بازی تلفن‌همراه شد. همچنین کار طراحی نیز انجام می‌دهم که تا امروز کاری که ارزش ذکرشـدن داشـته باشـد نکرده‌ام.

■ از چه سخت‌افزاری استفاده می‌کنید؟

بیشتر کارم را با یک رایانهٔ سفارشی ویندوز انجام می‌دهم. آن‌قدر که توقع دارم شـاهکار نیست، اما توانایـی‌اش بیـش از نیـاز من اسـت. به‌علاوه از سال ۲۰۱۱، یک MacBook Pro دست‌دوم نیز دارم. بـرای اینکـه بتوانیـد برنامه‌های کاربردی (اَپ‌هـا) را بـه Apple App Store ارسـال کنید باید یک دسـتگاه OS X داشـته باشـید؛ خوب اسـت که یک دسـتگاه قابل حمل باشـد. فعلاً در مسافرت هستم و از یک iPhone 5s برای تلفن و تست استفاده می‌کنم.

از یک (Wacom Intuos)aprox. A4 نیز اسـتفاده می‌کنم؛ البته به‌زودی با یک Cintiq جایگزینش می‌کنم، البته امیدوارم.

■ از چه نرم‌افزاری استفاده می‌کنید؟

بـرای توسـعهٔ بـازی از Unity 3D و MonoDevelop و بـرای کار هنری از Adobe Photoshop و InDesign و Illustrator اسـتفاده می‌کنـم. بـرای پویانمایـی و ویدئو از Adobe After Effects اسـتفاده می‌کنـم. هیچ‌وقـت Premiere را یـاد نگرفتـم؛ در AE ویرایش می‌کنم. اگر لازم باشـد کار صوتی کنم، از Audacity اسـتفاده می‌کنم. سعی می‌کنم که وارد کار سـه‌بعدی شـوم و بـرای آن از C4D اسـتفاده می‌کنم (نسـخهٔ دانش‌آموزی). گاهی با Sublime Text وبگاه (وب‌سایت) می‌سازم.

■ سیستم رؤیایی شما چیست؟

راسـتش را بخواهید از سیسـتم فعلی‌ام راضی‌ام. قبـلاً گفتـم کـه به‌فکـر خریدن یـک Cintiq و جایگزینـی‌اش بـا تبلت گرافیکـی Intous-ام هسـتم. به یک Macbook Pro جدیدتر و یک پردازنـده و کارت گرافیکـی بهتـر بـرای رایانهٔ شخصی‌ام نه نمی‌گویم! خیلی چیزها باید بیاموزم و بایـد مهارت‌هایـم را تقویـت کنـم و ایـن بـه سـخت‌افزاری که از آن اسـتفاده می‌کنم بستگی دارد.

آیا می‌توانید درست تصمیم بگیرید؟

تصمیم‌گیری به پایین‌ترین کیفیت رسیده است؛ حداقل این‌گونه به نظر می‌رسد. چند بار از کسی درخواست انجام کاری را کرده‌اید و در جواب پاسخ گرفته‌اید: «لازم است که آن را با رئیسم بررسی کنم» یا «ممکن است که ایمیلی برای تأکید آنچه توافق کرده‌ایم بفرستید»؟ آیا ما به دلیل پیامدهای نامساعد احتمالی، از تصمیم‌گیری بدون ارجاع و حسابرسی، بیش‌ازحد وحشت‌زده نیستیم؟

ما مدیران پروژه هستیم و باید به تصمیم‌گیری خود اعتماد داشته باشیم. آیا علت ضعف در تصمیم‌گیری، ناتوانایی در تسلط‌داشتن یا بی‌تمایلی به آن است؟ یا گاهی مشکل، کمبود اطلاعات برای تصمیم‌گیری است؟ برخی از این علل، نقص‌های انسانی و برخی دیگر نقص‌های طبیعی است. همان‌طور که در ادامه خواهید خواند، شاید بهترین شرایط طبیعی و مادی، به علت جنبه‌های انسانی به فقدان تصمیم‌گیری منجر شود. فرض کنید که در مسیر یک روستای کوچک رانندگی می‌کنید، در انتهای روستا به یک تقاطع با تابلوی عجیبی می‌رسید؛ بر روی یک جهت آن نوشته شده: «این راه»، بر روی جهت دیگر «آن راه» و به سمت جاده‌ای که در آن بوده‌اید، نوشته شده «یک جای دیگر». هیچ راه چهارمی نشان داده نشده است. این علامت سال‌ها است که در آنجا است و تقریباً هیچ اطلاعاتی را برای موتورسواران فراهم نمی‌کند. بدون هیچ اطلاعات یا با اطلاعات ناچیز، مجبور هستیم که یا به غریزه و یا به دانش قبلی پناه ببریم؛ شکی نیست که هر سه (اطلاعات، غریزه و دانش قبلی) برای رساندن ما به یک تصمیم مطمئن وارد بازی می‌شوند. حتی با این سه، این روزها اولین بند این مقاله زیاد تجربه می‌شود؛ که حاکی از آن است که نیروهای دیگری در کار هستند. این نیروها معمولاً با خطر یا پیامد و حتی پیگیری قانونی همراه هستند. امروزه شاهد پیگیری‌های قانونی مستند بسیاری هستیم که در آن‌ها یک تصمیم نادرست به یکی از این نیروها ختم شده است؛ پس نیاز به تأیید گیری از حسابرسی و یا موافقت رسمی یک مافوق، بی‌دلیل نیست.

ممکن است که آن تابلوی سه‌جهته هیچ اطلاعاتی ندهد اما همیشه در هر موقعیتی ذاتاً اطلاعات معینی وجود دارد. موتورسوار مسیری را که از آن می‌آید، به‌خوبی می‌شناسد زیرا همین الآن از آن آماده است؛ بنابراین حداقل یک راه وجود دارد که جهت قطعی را نشان دهد. چند بار وقتی معلوم شد که گم شدیم، مجبور شدیم قدم‌های خود را دوباره پیگیری کنیم؟ این سه مسیر دیگر هستند که مشکل‌سازند، اما حتی در اینجا کلیدهایی وجود دارد. فقط لازم است که بر شواهد موجود عمیق‌تر نگاه کنیم. با توجه به اینکه موتورسوار جاده را از جایی که می‌آید می‌داند و می‌تواند آن را بر نقشه تصور کند، می‌تواند جهت‌های قطب‌نما را در نظر بگیرد و یک منبع اطلاعاتی برای جهت‌یابی ایجاد می‌شود. اکنون دیگر انتخاب کردن مثل زدن تیری در تاریکی نیست. شاید باید در تصمیم‌گیری، شخصیتی مثل شرلوک هولمز داشته باشیم که می‌گوید «واتسون می‌دانی، اما نمی‌بینی!».

کار دیگری که تصمیم‌گیری مناسب را تضمین می‌کند، مدیریت ریسک است که می‌توان آن را به‌طور رسمی یا غیررسمی به کار گرفت. عبور از یک راه دوبانده شلوغ، به ارزیابی مستند و رسمی ریسک نیاز ندارد، اما ملاحظات در مورد ریسک وجود دارد که ملاحظاتی در مورد ریسک هست که طبیعتاً در نظر می‌گیریم. این ملاحظات ذاتاً شامل سه رکن مذکور هستند. اینکه با چه سرعتی تصمیم بگیریم به ما بستگی دارد. جنبه زمان از اهمیت برخوردار است زیرا ما، مدیران پروژه، غالباً برای تصمیم‌گیری صحیح تحت فشار هستیم. تصمیم سریع معمولاً مستلزم صرف وقت برای این سه رکن (اطلاعات، غریزه، دانش قبلی) است که شاید وقت کمتری بخواهند.

در کتاب «مدیریت ریسک پروژه‌ها و برنامه‌ها»، مثالی از یک شرکت مخابراتی را نشان داده می‌شود که برای عرضه‌ی هرچه سریع‌تر محصول تلفن همراه جدید خود تحت فشار است. رقابت در این بازار قدرتمند است، بنابراین بر تیم پروژه برای تکمیل به‌موقع و حتی زودتر پروژه فشار وجود دارد. نمونه‌های بسیاری از این شرکت‌ها وجود دارند که این نوع محصول را بسیار زود عرضه می‌کنند و متحمل هزینه آن که مشتریان ناراضی است می‌شوند. تصمیمی که باید در این مثال گرفته شود، انتخاب بین عرضه‌ی زودتر محصول و زمان تست کم، یا تکمیل تست و عرضه طبق زمان‌بندی است. خوشبختانه در این مورد یک تصمیم فوری لازم نیست، بنابراین بررسی خطرات، زمان وجود دارد اما زمان تصمیم محدود است. چقدر زمان برای جمع‌آوری و تحلیل داده‌ها لازم است که شما تصمیم مطمئنی بگیرید؟ یک تصمیم سریع شاید فاجعه‌بار باشد اما به تعویق انداختن تصمیم نیز می‌تواند مضر باشد. صرف‌نظر از خود تصمیم، نیروهای انسانی نیز که در زمان تصمیم‌گیری وارد بازی می‌شوند می‌توانند مشکل‌ساز باشند. برداشت، یکی از این نیروها است؛ اما در مورد برداشتِ چه کسی حرف می‌زنیم؟ آیا برداشت یک تصمیم‌گیرنده مطرح است یا برداشت‌های ذی‌نفعان دیگر همین‌قدر اهمیت دارند؟ مشکل این است که می‌توان از داده‌های مشابه برداشت‌های مختلفی کرد؛ بنابراین حتی وقتی که داده‌های مناسبی را جمع می‌کنید و اثبات می‌کنید که داده‌ها قابل‌اتکا هستند، ممکن است افراد آن را به شیوه‌های مختلفی تفسیر کنند.

چقدر در دنیای سیاست که توافقات در آن بسیار نادر است، شاهد این اتفاق بوده‌ایم؟ تصمیم درست غالباً فقط از ادراک درست حاصل می‌شود و یک مدیر ریسک خوب باید شطرنج‌باز خوبی باشد تا چند حرکت بعد را ارزیابی کند. بنابراین برای تصمیم‌گیری درست چقدر شانس دارید؟ آن‌قدر هم که بد به نظر می‌رسد نیست.

ترفند تصمیم‌گیری درست درک شرایط و ماهیت نیروهایی است که در حال بازی هستند. شاید هر موقعیت تصمیم‌گیری دارای ویژگی‌های متفاوتی باشد. تصمیم برای عبور از مسیر دوبانده کاملاً با تصمیم برای عرضه‌ی جلوتر از زمان‌بندی تفاوت دارد. فقط باید مطمئن شویم که با توجه به شرایط، معیار درستی از تصمیم‌گیری را به کاربرده‌ایم.

بیشتر در مورد جان بارتلت، نویسنده‌ی این مقاله بدانیم:

جان بارتلت یک مؤلف نامدار در زمینه مدیریت پروژه و عضو افتخاری APM است. تجربه مدیریت برنامه و پروژه و مقالات و کتاب‌های بسیار او گواه بر جایگاه در رهبری تفکر مدیریت پروژه است. او به دلیل ارائه‌ی پیشنهادهای عملی و معقول برای راهکارهای آزمایش‌شده و امتحان‌شده در مشکلات معمول مدیریت پروژه شناخته شده است. از دیگر کتاب‌های جان عبارتند از: «همیشه اول»، «هر زمان» و «مدیریت ریسک برنامه‌ها و پروژه‌ها».

چطور آخرِ روزِ کاری، کار کردن را تمام کنیم و از استرس بپرهیزیم؟

مدیران زیادی هستند که نمی‌توانند بخوابند. ذهنشان ناآرام است و به کوه وظایفی که در اداره پیش رو دارند فکر می‌کنند. برخی زمان شام با ایمیل‌های کاری از شام خوردن دست می‌کشند. خانوادهٔ برخی مدیران و افراد پرمشغله شکایت می‌کنند که انگار هیچ‌گاه کارشان تمام نمی‌شود. کارکنان بسیاری هستند که نمی‌توانند در آخر روز کاری، دست از کار بکشند یا ذهنشان را از کار آزاد کنند. طبق یک تحقیق هفت‌ساله دربارهٔ عملکرد کارکنان، ناتوانی در تفکیک زمان کاری از زمان شخصی جزء ۱۰ عامل استرس‌آوری است که افراد به‌موجب آن کارآیی کمتری در کنترل امور می‌یابند. البته فناوری، که با قرار دادن محل کارمان در صفحه‌های لمسی‌مان، هم دسترسی را آسان کرده و هم کار را به ما در ساعات شخصی تحمیل کرده، در بدتر کردن این مشکل بی‌تقصیر نبوده است. اما چطور می‌توانیم کار را در محل کار بگذاریم و زندگی‌مان را در منزل مطلوب‌تر و کم‌استرس‌تر کنیم؟ در برخی از مشاوره‌ها با مدیران ارشد، آن‌ها را تشویق می‌کنند که از یک روتین آخر روز استفاده کنند تا بتوانند مابین دو دنیای کار و زندگیِ شخصی یک مرز روانی ایجاد کنند. ۵ راهبرد زیر با ۲۶ مدیر امتحان شده است؛ به‌کارگیری این راهبردها تعداد مدیران توانمند در تفکیک کار از ساعات شخصی را از ۴۰ درصد به ۶۸ درصد افزایش داده است.

قبل از خروج از اداره یک وظیفهٔ کوچک دیگر انجام بدهید

مثل یک تماس تلفنی، امضای چیزی یا پاسخ به ایمیلی. به‌این‌ترتیب روزتان را با حس مثبتی که از تکمیل کار خبر می‌دهد به‌پایان می‌برید. اینکه بدانید تصمیم دارید خودتان را به انجام کاری وادارید ولی اکنون یک بخش کوچک از آن را انجام می‌دهید و صبح روز بعد ادامه‌اش می‌دهید حس خوبی القا می‌کند. در تحقیقی که ترزا امابیل و استیون کرَمر، مؤلفان کتاب اصل پیشبرد، انجام داده‌اند، متوجه شده‌اند که حس «موفقیت‌های کوچک» حال شما را بهتر می‌کند.

یک لیست بکن‌ها (To do list) تهیه کنید

وظایفی را که باید تکمیل شوند یا به‌صورت دیجیتالی یا روی کاغذ بنویسید؛ بهتر است که به‌ترتیبِ اهمیت نوشته شوند. نوشتن این لیست یکی از سه مهارت مؤثر در بهبود عملکرد و کنترل موفقیت‌آمیز استرس است.

محل کارتان را مرتب کنید

سازمان‌دهی امور قبل از دست کشیدن از کار به شما کمک می‌کند که برای صبح روز بعد سرزنده‌تر باشید. در همان تحقیق ثابت شده که اگر مدیران و غیرمدیران میز کارشان را به‌هم‌ریخته ترک کنند، استرس و تنش روز قبلی‌شان صبح روز بعد تکرار می‌شود. بنابراین شاید این ایده که داشتن یک میز تمیز نشانِ داشتن یک ذهن سرزنده است درست باشد.

کار به‌خصوصی را، به‌عنوان نماد پایان کار، تعیین کنید

کاری که مثل یک دستور صریح می‌ماند و برای شما نماد اتمام فکرِ کردن دربارهٔ امور کاری است؛ مثلاً قفل کردن در اداره، خاموش کردن نمایشگر رایانه یا تلفن زدن به منزل. استفادهٔ مداوم از این نشان‌های معین به شما کمک می‌کند که احساسات خود را کنترل کنید و وضعیت روانی خود را بهتر کنید؛ درست مثل این می‌ماند که زمان خروجتان را در یک برگهٔ زمانی ثبت می‌کنید. تحقیقات مؤکدِ قدرتِ این مراسم و روتین‌هاست.

عصر را با یک حس مثبت شروع کنید

به‌جای احوالپرسی از دوستان و اعضای خانواده با عبارات متداولی چون «خوبی؟ چه خبر؟» سعی کنید سر صحبت را دربارهٔ تجربیات منفی کار یا استرس درس و کلاس با اعضای خانواده باز کنید. دربارهٔ مشکلات صحبت کنید. بپرسید که در روز چه اتفاقات خوب یا هیجان‌انگیزی برای آن‌ها افتاده و با آن‌ها دراین‌باره مکالمه کنید. اگر کسی از شما دربارهٔ روزتان پرسید از توضیحات طولانی اجتناب کنید، مگر اینکه مطمئن باشید به رفع یک نگرانی کمکی می‌شود. این راهبرد پنج‌مرحله‌ای به کمترین زمان و تلاش نیاز دارد؛ روزانه بین ۱۰ تا ۱۵ دقیقه زمان می‌گیرد. شاید برخی از این راهنمایی‌ها و تکنیک‌ها به‌نظر جدید برسند، اما وقتی به‌ترتیب و با همدیگر به کار برده می‌شوند، جداً حس استرس را کاهش می‌دهند و بین کار و زندگی توازن ایجاد می‌کنند.

داده پرداز (پویای شریف)

ارائه دهنده راه کارهای جامع و تخصصی تحت وب

www.DadehPardaz.com

آدرس: تهران، میدان ونک، خیابان برزیل، تقاطع توانیر، بن بست امید، پلاک ۶، واحد ۱

شماره تماس: ۰۷۰-۸۸۶۵۹۴۶۸ ۰۲۱

پست الکترونیک: info@dadehpardaz.com

کتاب‌طوری، یک انتخاب فرهنگی

فروش و ثبت سفارش انواع کتب و محصولات فرهنگی

- ارائه اطلاعات جامع پیرامون کتب
- ارائه امکان تبادل اطلاعات بین کاربران پیرامون کتب
- امکان ثبت سفارش کتبی که موجود نیست
- ایجاد بستر مطمین جهت خرید یک کتاب همسان با سلیقه کاربر
- ایجاد شرایط مناسب جهت افزایش میزان مطالعه
- رویکرد متفاوت پیرامون تشویق مخاطبین به مطالعه
- محدود نبودن سفارش کتاب و محصولات فرهنگی از سایت

آدرس: تهران، میدان ونک، خیابان برزیل، تقاطع توانیر، بن بست امید، پلاک ۶،واحد ۱

شماره تماس: ۷۰-۸۸۶۵۹۴۶۸ ۰۲۱

دورنگار: ۸۸۶۵۹۴۷۰ ۰۲۱

آدرس فروشگاه: www.ketabtori.com

ایمیل: info@ketabtori.com