رویداد: مدیریت تغییر 2019

رویداد: کنفرانس مدیریت پروژه‌ی APM لندن

رویداد: کنفرانش 2019PMO

خط دید

نرم افزار

سال دوم. شماره ۷٥
دی‌ماه سال ۱۳۹۷
قیمت: ۱۵۰۰۰ تومان

مدیریت پروژه از تئوری تا عمل

راهــکاری

پیشنهادی‌برای مواجهه‌بااختلافات اجتناب‌ناپذیر درتیم

راهنمای جدید PMBOK

- چگونه به اهداف دست یابیم؟
- فرهنگ سازی عامل ماندگاری
- فرآیند پیشبرد یک پروژه از شروع تا پایان

PMBOK

- راهنمای جدید PMBOK
- عامل موفقیت پروژه‌ها چیست؟
- بازخورد (فیدبک) یک مکلمه دوطرفه است، نه یک خیابان یک طرفه

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 ۶، واحد ۱
تلفن : ۷۰ - ۸۸۶۵۹۴۶۸   ایمیل : info@khatedid.com

خدمات دولت هوشمند

www.iran.gov.ir

هر زمان و هر مکان، پای درس بهتـــرین استاد

WWW.MODARES.ONLINE

چرا مدرس آنلاین؟

با مـدرس آنلایـن ارزش‌هـای زیـادی را یک‌جا دریافت کنید.
آموزش‌هـای دلخـواه شـما در هـر زمـان و مکانی در دسترس شماست.
با مـدرس آنلایـن، بـرای بهتریـن 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

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 ۶، واحد ۱
تلفن : ۷۰ - ۸۸۶۵۹۴۶۸ ایمیل : info@khatedid.com

یک روش مطمئن، متفاوت و شیرین برای انتخاب یک کتاب
به زودی ...

مدرس آنلاین

هر زمان و هر مکان، پای درس بهتـــرین 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

همراه من
my mci

سامانه
خدمات الکترونیکی
همراه اول

مشترکین **دائمی** و **اعتباری** همراه اول می‌توانند با استفاده از اپلیکیشن همراه من، به خدمات مختلف همراه اول به صورت **آنلاین** دسترسی داشته باشند.

ویژگی‌های برنامه

- مشاهده و پرداخت صورتحساب پایان دوره و میان دوره
- مشاهده موجودی و خرید شارژ (ویژه مشترکین اعتباری)
- دریافت گزارش کامل ریز مکالمات و خلاصه کارکرد
- فعالسازی، خرید و مشاهده گزارش مصرف بسته های نوترینو
- و...

امکانات برنامه

- صورت حساب پایان دوره
- گزارش مصرف اینترنت
- انتقال اعتبار

www.my.mci.ir

ERP

Enterprise Resource Planning

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

www.booket.ir
بوکت
BOOKET
ارزان بخوانید
راحت بدانید
امکان مطالعه در هر زمان و مکان
کتاب های آکادمیک
GET IT ON
Google Play
@Booketacademic
Booketacademic
Booket Academic

پیام گستر
PAYAM GOSTAR
فناوری در مدار مشتری

٪۳۰ فروش بیشتر
٪۶۰ پشتیبانی بهتر
٪۱۰۰ بهبود روابط
با مشتریان

نرم افزار CRM پیام گستر

- تمرکز بر روی نیاز مشتریان ، نه فقط دانش و ابزار
- عدم استرس بر امنیت اطلاعات و داده ها
- با سازمانتان یک کلیک فاصله دارید
- به وسعت نیاز های شما
- بیشتر از یک CRM، یک همکار استراتژیک
- چرخ پیام گستر آنگونه که شما می خواهید می چرخد

برگزیده ای از ماژول های پیام گستر

- کنسول تحت وب
- مدیریت سوابق
- مدیریت وظایف
- فرم ساز
- مدیریت فرصت‌ها
- شخصی سازی CRM
- و...

1st
عضو تایید شده
تجارت الکترونیک اول

شماره تماس
۰۲۱ - ۴۹۷۱۴

مشاهده دمو آنلاین
نرم‌افزار CRM پیام گستر

متمم

محـل توسـعه مهـارت‌های فـردی

متمـم، مجموعـه‌ای از مقاله‌هـای آموزشـی نیسـت. بلکـه یـک ابـزار یادگیـری اسـت کـه بـه شـما کمـک می‌کنـد بـه تدریـج، بـا مطالعـه، حـل تمریـن، یادگیـری از دوسـتان‌تان و در یـک کلام، وقـت گذاشـتن منظـم بـرای خودتـان، دانـش و مهـارت های خود را در مدیریت و زندگی توسعه دهید.

متمـم، توسـعه دانـش و مهـارت فـردی را نـه صرفا بـه عنـوان ابـزاری بـرای رشـد و پیشـرفت و موفقیـت فـردی، بلکـه بـه عنـوان تنهـا راهـکار عملـی رشـد و پیشرفت و توسعه جامعه می‌بینند.

کارگاه افزایش عزت نفس

- ✓ بررسی نشانه‌های کمبود عزت نفس
- ✓ تکنیک بهبود عزت نفس
- ✓ ارائـه فایل‌هـای صوتـی مرتبـط بـا عـزت
- ✓ نفس

چالش‌هـا، راهکارهـا و روشـهای افزایـش عـزت نفس در محیط کار و زندگی

محـل توسـعه مهـارتهای من

برگزاری دوره های

MBA

دوره استعدادیابی

کارگاه آموزش مهارتهای تیمی

بن ریل

BONRAIL

WWW.BONRAILCO.IR

مسابقه آنلاین کیو
www.APPQ.ir

مدیریت پـروژه از تـئوری تا عمل
شـمـــــاره ۷ • دی مـاه ۱۳۹۷

صاحب امتیاز: داده‌پرداز پویای شریف
مدیرعامل: مهدی استوی
مدیر مسئول و سردبیر: محمدرضا اسماعیل طهرانی
مدیر اجرایی: امیر تیمور نژاد
گرافیک و صفحه آرایی: امیر شریف
واحد جذب آگهی: بازرگانی نشریه خط دید
اعضـای تحریریـه: محمدرضـا اسـماعیل طهرانـی، محمدبهنیا، علیرضا فیروزی

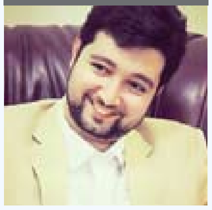
آدرس: تهـران، میـدان ونک، خیابـان برزیل، تقاطع توانیر، بن‌بست امید، پلاک ۶، واحد ۱
تلفن دفتر نشریه: ۷-۰۲۱-۸۸۶۵۹۴۶۸
نمابر: ۰۲۱-۸۸۶۵۹۴۶۸

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

لیتوگرافـی، چـاپ و صحافـی: چـاپ ایـران کهـن

» تحریریـه در رد، تلخیـص و ویرایـش مطالب ارسـالی مجاز اسـت.
» مسوولیت مطالب و آگهی‌ها به عهده نویسنده و سفارش‌دهنده است.
» چـاپ مطالـب و تصاویـر مجله بدون اجازه کتبی مجاز نمی‌باشـد.
» مطالب دریافت شده بازپس داده نمی‌شوند.

فاصلهٔ بین برنامه‌ریزی تا تحقق برنامه ۱۱
راهکاری پیشنهادی برای مواجهه با اختلافات اجتناب‌ناپذیر در تیم ۱۲
چهار روش برای تقدیر و تشکر از تیم‌تان بدون اینکه هزینه‌ی زیادی بپردازید ۱۶
مدیران پروژه باید چه چیزهایی در مورد GDPR بدانند ۱۸
رویـداد: مدیریت تغییر ۲۰۱۹ ۲۱
۱۰ راهکار برای مالکان محصول در چارچوب اسکرام ۲۲
رویـداد: کنفرانس مدیریت پروژه‌ی APM لندن ۲۰۱۹ ۲۵
بازخورد (فیدبک) یک مکالمهٔ دوطرفه است، نه یک خیابان یک‌طرفه ۲۶
چگونه به اهداف دست یابیم؟ ۲۸
فرهنگ‌سازی، عامل ماندگاری ۳۰
عامـل مـوفقیت پروژه‌ها چیست؟ ۳۲
آگاهی، ادراک و مسئولیت‌پذیری ۳۴
رویـداد: کنفرانس PMO ۲۰۱۹ ۴۰
انعطاف‌پذیری بیشتر مدیریت ۴۲
راهنمای جدید PMBOK ۴۴
رویـداد: کارگاه تحلیل شخصیت و روش‌های انتخاب نیروی انسانی متناسب ۴۶
سوزهینتون- توسعه‌دهندهٔ مایکروسافت، هکر سخت‌افزار ۴۷
چگونه با خلأهای آموزشی مربوط به مدیریت خانم‌ها در محل کار مواجه شویم ۴۸
فرآیند پیشبرد یک پروژه از شروع تا پایان ۵۲



نویسنده: محمدرضا اسماعیل طهرانی

فاصلۀ بین برنامه‌ریزی تا تحقق برنامه

یکی از اصلی‌ترین مسائل و چالش‌های مهم واحدها و دفاتر مدیریت پروژه فاصلۀ بین برنامه‌ریزی یک برنامه و اجرای آن برنامه است. اما آیا واقعاً همیشه تمام برنامه‌ریزی‌های صورت‌گرفته، ذره‌به‌ذره، مطابق انتظار برنامه‌ریز پیش می‌روند؟ به‌نظر می‌رسد اگر این انتظار درست و صحیح باشد، دانش مدیریت ریسک، دانش و مهارت حل و تحلیل مسائل پیش‌آمده، و مهارت کار گروهی، به‌واقع، بی‌معنا باشند. همواره باید احتمال محقق نشدن هر برنامه‌ای را در نظر داشته باشیم. بهترین راه‌حل برای مواجهه با این احتمال صرفاً پیش‌بینی تاحد امکان است. باید بتوانیم به مهارتی دست یابیم که ما را در پیش‌بینی مسائل احتمالی برای هر نقطه از برنامه‌ریزی صورت‌گرفته توانمند سازد؛ دراصل باید بتوانیم آمادگی لازم را برای حل آن‌ها کسب کنیم. عموماً باید در برنامه‌ریزی‌ها، ریسک یا زمان عقب یا حتی جلو افتادن از برنامه را نیز در نظر بگیریم. به‌یاد داشته باشیم که همان‌قدر که عقب ماندن از برنامه‌ریزی مضر است، جلوافتادن از آن هم بی‌ضرر نخواهدبود؛ زیرا منابع تخصیص داده‌شده را به‌سادگی هدر خواهیم داد. بی شک، یک مدیر باتجربۀ پروژه فاصله‌ای را که مابین تنظیم یک برنامه تا تحقق آن وجود دارد به بطالت یا پیش‌روی براساس ذهنیتی که از برنامۀ تنظیم‌شده دارد نمی‌گذارند، بلکه به ارزیابی تمام مخاطرات احتمالی و وضعیت عینی پروژه ادامه می‌دهد. پیش‌بینی و ارزیابی ریسک، همچون پلی امن و محکم، ما را از برنامه به تحقق برنامه می‌رساند. در این شماره از فصلنامۀ خط دید، علاوه‌بر موضوعات مختلف و به‌روز مدیریت پروژه، سعی کرده‌ایم به این موضوع توجه ویژه‌ای داشته باشیم و از زوایای مختلفی به آن بپردازیم. در شماره‌های آتی مجله دربارۀ ترکیب انواع مهارت‌های موردنیاز برای حل مسائل در پروژه‌ها بیشتر خواهید خواند.

راهکاری پیشنهادی برای مواجهه با اختلافات اجتناب‌ناپذیر در تیم

(برای حل اختلافاتِ اجتناب‌ناپذیر در تیم چه راهکارهایی پیشنهاد می‌شود؟)

باب راث: عملکرد یک تیم به‌عنوان یک کل، موفقیت آن را تعیین می‌کند. حتی اگر بهترین و برجسته‌ترین افراد دنیا را در تیم خود داشته باشید، اما درصورتی که این افراد نتوانند باهم کار کنند، آن مجموعه ارزش لحظه‌ای زمان گذاشتن را ندارد. در میان عوامل متعدد موفقیت در پروژه‌های حساس، از جمله برنامه‌ریزی دقیق و با جزئیات، محاسبات صحیح و ارتباطات شفاف، قدرتمند و یکپارچه بودن تیم، یکی از مهم‌ترین معیارها به شمار می‌رود. بدون این عامل، درگیری‌ها و اختلافات موجود در تیم می‌تواند پروژه‌های شما را از مسیر خود منحرف کرده یا حتی متوقف سازد. رهبران پروژه‌ها امروزه با درکِ اهمیتِ داشتن یک تیم متحد و یکپارچه، به‌طور قابل توجهی بر ساخت تیم‌های مناسب تمرکز دارند، تیم‌هایی که از اعضایی تشکیل‌شده‌اند که نه‌تنها مهارت‌های مکمل یکدیگر را دارند، بلکه با یکدیگر به‌خوبی کار می‌کنند. اما حتی در میان قدرتمندترین تیم‌ها نیز ایجاد درگیری و اختلاف، امری طبیعی و اجتناب‌ناپذیر است.

اجتناب‌ناپذیری اختلاف

اختلاف، بخشی از بشریت محسوب می‌شود. شخصیت‌های قدرتمند اغلب با سایرین برخورد می‌کنند؛ زیرا هر یک از افراد، پیشینه‌ی متفاوت و ارزش‌های گوناگونی دارند. این تفاوت‌های ظریف شخصیتی را با تفاوت‌های دیگر تلفیق کنید تا به دستورالعملی بی‌نقص برای دستیابی به موفقیت و البته، مدیریتِ اختلافات برسید.

اما اختلاف چه زمانی رخ می‌دهد و چگونه با آن مواجه شوید؟

مواجهه با اختلاف: آیا به یک طرح و برنامه نیاز دارید؟

در دنیای ایده‌آل، تمام اختلافات مانند هم بوده و به‌راحتی برطرف می‌شوند. افراد به هر موقعیتی واکنشی یکسان نشان می‌دهند و حتی احساس آنان نیز یکسان است. شما تمام متغیرها را می‌شناسید، یک برنامه و طرح دارید و همیشه دقیقاً می‌دانید که چه باید بکنید. اما احساس و عاطفه‌ی بشر، متغیری است که به‌سختی کنترل می‌شود. نمی‌توان با اطمینان پیش‌بینی کرد که مردم در قبال شرایط به‌وجودآمده چه احساسی خواهند داشت و چه واکنشی از خود نشان می‌دهند؛ حال نظر دادن در مورد اینکه آنان چگونه با یکدیگر کار می‌کنند نیز کار را مشکل‌تر می‌سازد. با توجه به متنوع و منحصربه‌فرد بودنِ اختلافات، آیا داشتنِ برنامه‌ای برای کنترل این اختلافات، ارزشمند است؟

مسلماً شما می‌توانید X گام، طراحی کنید که برای مواجهه با اختلاف به وجود آمده، از آن‌ها پیروی نمایید و آن گام‌ها را با هر موقعیتی تطبیق دهید، که با واضح‌ترین روش آغاز می‌کنیم؛ یعنی شناسایی دلیل اختلافِ به وجود آمده.

از آن مرحله به بعد شما می‌توانید جلسه‌ای با حضور افراد مرتبط با اختلاف به‌وجودآمده برگزار کنید و آنان را تشویق نمایید تا به مشکلات یکدیگر گوش دهند؛ به این امید که بین آنان صلح برقرار شود. اما حتی در این صورت نیز هیچ تضمینی برای حل اختلاف آنان وجود ندارد و ممکن است مجبور شوید یک نفر را به‌عنوان واسط میان آنان در نظر بگیرید.

راهکار فوق، روشی است که بسیاری از رهبران پروژه‌ها اتخاذ می‌کنند، اما من می‌خواهم روشی متفاوت را به شما پیشنهاد کنم. یک راهکار دیگر برای مدیریت و کنترل اختلاف آن است که اختلافِ به وجود آمده را اصلاً مدیریت نکنید. دو سال پیش، زمانی که در حال ساخت کسب‌وکار خود بودم، به همراه سه نفر دیگر، اعضای یک گروهِ کاری بودیم. ما هفته‌ای یک‌بار با یکدیگر ملاقات می‌کردیم تا نه‌تنها مسئولیت‌های خود را در قبالِ به اهدافمان حفظ کنیم بلکه ایده‌هایمان را نیز با یکدیگر به اشتراک بگذاریم. ازآنجاکه زمان زیادی را با یکدیگر می‌گذراندیم و این احتمال وجود داشت که در پروژه‌های آینده نیز بخواهیم با یکدیگر همکاری کنیم، یکی از دوستانم در تیم پیشنهاد کرد که تست‌های شخصیتی بدهیم تا رفتارهای یکدیگر را بشناسیم و پویایی تیم را آنالیز کنیم.

او به ما Shadowmatch را معرفی کرد، سیستم آنلاین نقشه‌برداری رفتاری که کمپانی‌ها، مربیان، روان-سنج‌ها و روانشناسان از آن استفاده می‌کنند تا با دقت نیروهای خود را استخدام نمایند و رفتارهای منحصربه‌فرد هر شخصی را دریافته و به آنالیز، آموزش و ساخت تیم بپردازند. در حقیقت، ویژگی‌های آموزش، آنالیز و ساخت تیم بسیار مفید و کاربردی بودند. پس از تکمیل تست‌ها، Shadowmatch نتایج ما را

روی نمودار به تصویر کشید تا آن‌ها را مقایسه کند. پلتفرم شکل گرفته موارد زیر را مشخص کرد:

- نقاط قوت و ضعف هر یک از ما
- وجه تشابهات ما در این نقاط قوت و ضعف
- رفتارهای به‌خصوص همه‌ی ما که با دقت مشاهده شد
- اعضایی از گروه ما که رفتارهای بسیار متفاوتی با سایرین داشتند

هر یک از ما می‌توانست به نتایج، دسترسی داشته باشد تا پویایی تیم را مشاهده نماید. با آنالیز و بررسی مقایسه‌ها، توانستیم نقاط اختلاف احتمالی را از همان آغاز شناسایی کنیم. این شناسایی و تشخیص، به آن معنا بود که ما آماده‌شده بودیم تا با هر گونه اختلافی روبرو شویم؛ زیرا دلایل آن اختلافات را می‌دانستیم.

مزایای متعدد شناخت و آگاهی اعضا نسبت به پویایی تیم

مسلماً شما مجبور نیستید از ابزاری مثل Shadowmatch استفاده کنید؛ اما اگر اعضای تیم شما رفتارهای خود و سایر اعضا را بهتر بشناسند، در مواجهه با اختلافاتشان قدرتمندتر خواهند شد، بدون اینکه درگیر اختلاف به وجود آمده شوند.

شما و تیمتان به روش‌های گوناگونی می‌توانید از روش فوق‌الذکر بهره‌مند شوید:

- نقشه‌برداری رفتاری می‌تواند به شما در ساخت بهتر تیمتان از آغاز کمک کند. به‌عنوان‌مثال، می‌توانید اعضایی را که با سایرین تطابق ندارند و برای تیم مناسب نیستند شناسایی کرده و به آنان مسئولیت دیگری بدهید.
- زمان کمتری برای مدیریت تیمتان صرف خواهید کرد، درواقع زمان بیشتری برای سایر وظایف ضروری پروژه دارید.
- اعضا یکدیگر را بهتر خواهند شناخت و می‌فهمند که چه کاری را به چه علت انجام دهند.
- اعضا می‌آموزند تا به روشی سازنده با اختلافات مواجه شوند. شما احتمالاً متوجه می‌شوید اعضایی که دائماً با مسائل برخورد می‌کنند به تدریج به افرادی تبدیل می‌شوند که به آن موضوعات بخندند، چراکه می‌آموزند شاد باشند و به تفاوت‌های یکدیگر احترام بگذارند.

نتیجه‌گیری

اختلافات تیم همیشه اجتناب‌ناپذیر باقی می‌ماند، اما مدیریت آن اجتناب‌ناپذیر نیست. با دادن ابزارهایی به تیم خود، برای اینکه خودشان اختلافات را مدیریت کنند، به آنان قدرت ببخشید و همگی از این وضعیت و مزایای آن بهره‌مند شوید. زمان کمتری برای مدیریت و زمان بیشتری برای رهبری تیم خواهید داشت، همچنین تیم شما خواهد آموخت که تفاوت‌های هر یک از افراد در حقیقت بزرگ‌ترین نقطه قوتِ همه‌ی افراد است.

یک روش مطمئن، متفاوت و شیرین برای انتخاب یک کتاب

به زودی ...

چهار روش برای تقدیر و تشکر از تیم‌تان بدون اینکه هزینه‌ی زیادی بپردازید

با توجه به اینکه چند روز آینده روز شکرگزاری است و به‌روزهای تعطیلات ماه دسامبر نیز نزدیک می‌شویم، می‌خواهم به اهمیت زمان گذاشتن برای شناخت تیم اشاره کنم. ما مدیران باید به اعضای تیم توجه کنیم اما پایان سال معمولاً فرصتی بزرگ در اختیار ما قرار می‌دهد. تیم‌های پروژه ساعت‌ها کار می‌کنند تا پروژه‌های بسیار طولانی را تحویل دهند و این وظیفه‌ای خسته‌کننده است که مشکلات و مسائل بسیاری را یکی پس از دیگری بر سر راه افراد قرار می‌دهد. وقتی پروژه درنهایت تمام می‌شود، توجه به تیم، امری مهم است.البته که همه به پول علاقه دارند اما افزایش حقوق تنها یک انگیزه‌ی موقتی است و معمولاً سالی یک‌بار انجام می‌شود. جشن‌های تیمی و مهمانی‌های پایان سال همیشه مفید و انگیزه‌بخش هستند؛ اما زمانی که بودجه محدود باشد، مدیران گاه باید در یافتن روش‌های تشویقی مقرون‌به‌صرفه برای تشکر از تیم خود بابت کار خوبی که انجام داده‌اند خلاق باشند. در اینجا چهار راه برای تشکر از اعضای تیم بدون نیاز به صرفه هزینه‌ی زیاد یا استفاده از بودجه‌ی شخصی ارائه‌شده است:

تشکر از تیم شماره ۱: پیتزا بگیرید

در شهر دیترویت، ما رستوران زنجیره‌ای Buddy Pizza داریم، رستورانی که از سال ۱۹۴۶ تا امروز پیتزاهای بسیار خوشمزه‌ای درست می‌کند. مطمئناً شما هم در شهر خود یک پیتزافروشی قدیمی و عالی دارید. افرادی که در ایالات‌متحده زندگی می‌کنند می‌توانند به سایت eater.com بروند تا بهترین پیتزافروشی‌های شهر خود را جستجو نمایند. پیتزا گزینه‌ی بسیار خوبی برای تشویق اعضای تیم است؛ زیرا ظاهراً هم با سلیقه‌ی گیاه‌خواران تناسب دارد و هم گوشت‌خواران؛ چون در کنار پیتزا، سالاد و تکه‌های نان نیز سرو می‌شود. با خرید پیتزاهای ۲۰ دلاری، هزینه‌ی شما برای ده نفر حدود ۲۰۰ دلار می‌شود.

در یکی از پروژه‌های اخیر، بودجه‌ی در نظر گرفته‌شده‌ی ما کم بود، بنابراین من هزینه‌ی این ناهار ۲۰۰ دلاری را با مدیر دیگری تقسیم کردم. در پروژه‌های دیگر، من هزینه‌ی ناهار را از جیب پرداخت می‌کنم. پرداخت هزینه برای جمع کردن اعضای تیم در کنار هم کاملاً ارزشش را دارد چون باعث مستحکم‌تر شدن روابط بین اعضا می‌شود.

تشکر از تیم شماره ۲: به آنان اشانتیون بدهید

اگر بودجه‌ی کافی برای پاداش دادن به تیم‌تان را ندارید، فرصت دیگری دارید که اقلام آن را فروشندگان نرم‌افزارهایتان تهیه می‌کنند. نمایندگان فروش و دپارتمان‌های بازاریابی اغلب بودجه‌های اختیاری دارند که می‌توانند با آن اعضای تیم خود را به صرف ناهار دعوت کنند و یا تعدادی دفترچه یادداشت، پدموس، تیشرت و خودکار (اشانتیون) بدهند. اگر تیم شما با استفاده از یک راهکار نرم‌افزار فروشنده کار بزرگ و مهمی انجام داده‌اند، از نمایندگان فروش خود بپرسید که آیا آنان اشانتیون‌های تبلیغاتی دارند یا خیر.

من تا به حال این کار را بارها انجام داده‌ام و متوجه شدم که فروشندگان همیشه مایل‌اند اشانتیون‌های تبلیغاتی از برندهای معروف و بزرگ را به مردم ارائه دهند. به همین دلیل است که به آنان بودجه‌ی بازاریابی تعلق می‌گیرد!

تشکر از تیم شماره ۳: به آنان کتاب بدهید

یک روش جالب برای پاداش دادن به شخص یا گروه، دادن کتاب است. من به‌شخصه از طرفداران خلاقیت افراد و کتاب‌هایی هستم که به اعضای تیم کمک می‌کنند تا به اهدافشان جامه‌ی عمل بپوشانند. هر کتابی بین ۱۰ الی ۱۵ دلار قیمت دارد و خرید آن‌ها به‌عنوان بمبی از دانش جدید، روشی عالی برای تشکر از تیم است! در زیر من پنج کتاب را پیشنهاد می‌کنم.

- GRIT: قدرت عشق و پشتکار نوشته‌ی آنجلا داک وُرث

GRIT کتابی الهام‌بخش است که انگیزه برای دستیابی به نتایج و پشتکار برای دستیابی به چالش‌های کسب‌وکار را نهادینه کرده و پرورش می‌دهد.

- صبح معجزه نوشته‌ی هال الرود
- این کتاب بهترین تمرینات و نصیحت‌ها را در مورد خلاق‌تر بودن ارائه می‌کند، با اینکه قبل از ۵ صبح بیدار شدن، کار سختی است.
- یک چیز نوشته‌ی گری کِلر
- این کتاب با ارائه‌ی مسیری روشن از وظایف اولویت‌بندی شده به شما کمک می‌کند تا احساس سردرگمی‌تان برطرف شود و در زمان کمتری برای رسیدن به اهدافتان پیشرفت کنید.
- رسیدن به بله نوشته‌ی فیشر و یوری
- من این کتاب را ۱۵ سال پیش خواندم نتایج آن همچنان در قراردادها و مذاکرات من تأثیرگذار بوده و کاربردی می‌باشد.
- ماجراهای جانی بانکو: آخرین راهنمای شغلی که باید بخوانید نوشته‌ی دنیل پینک

ماجراهای جانی بانکو یک کتاب شغلی سرگرم‌کننده و ساده است که به‌صورت یک داستان (رمان) روایت‌شده است. هدیه‌ای عالی که اصول برنامه‌ریزی شغلی کاملی را مطرح می‌کند.

کتاب، هدیه‌ای بسیار خوب بوده و دوام آن بسیار بیشتر از یک ناهار است!

تشکر از تیم شماره ۴: یک یادداشت بنویسید

در دوره‌ای که یادداشت‌های تبریک، تشکر، یادآوری و غیره بیشتر در فضاها و شبکه‌های مجازی نظیر فیس‌بوک و اینستاگرام رایج شده، یک یادداشت با دستخط خودمان بر روی یک ورق کاغذ یا کارت، همچنان روشی جذاب و خوشایند برای تشکر به شمار می‌رود. چند سال پیش من یادداشتی از مدیر IT خود دریافت کردم که با کاغذ و قلم خودش آن را نوشته بود. بسیار تحت تأثیر قرار گرفتم که باوجود ساده بودن یادداشت اما زمان گذاشته بود و چند کلمه‌ی متفکرانه نوشته بود که ارزش کار وی را به‌خوبی می‌رساند. او تنها چند دقیقه وقت گذاشته بود تا آن کلمات پرمعنا را روی کاغذ پیاده کند، اما این تشکری بود که هنوز هم در خاطرم مانده است.

تشکر کردن از تیم الزاماً نباید با مخارج زیاد همراه باشد. پرداخت هزینه در حدی که یک عصر را در کنار تیم خود، دورهم جمع شوید یا آنان را به ناهاری دعوت کنید که هزینه‌ی هر نفر حدود ۸ دلار یا حتی بیشتر شود، درصورتی که مشکل بودجه نداشته باشید، کاری مفید و مثبت است. اما بودجه‌ی پروژه‌ها محدود هستند و اگر نمی‌توانید با این بودجه از تیم‌تان تشکر کنید، ایده‌هایی که مطرح کردم می‌توانند کاربردی باشند.

مدیران پروژه باید چه چیزهایی در مورد GDPR بدانند

(حتی اگر در اروپا زندگی نمی کنید)

ظاهراً هر هفته خبری می‌رسد که یک کمپانی از داده‌های مجموعه‌ی دیگری سوءاستفاده کرده است. این روزها داده‌ها بسیار خبرساز شده‌اند، چه از جهت کپی کردن و استفاده از داده‌ها توسط کمپانی‌های مارکتینگ و چه سرقت داده‌ها توسط هکرها. در میان تمامی خبرهای ناخوشایندی که از پایگاه داده‌ها و مشکلات حفاظتی به گوش می‌رسد تعدادی استاندارد وجود دارد که تا حدودی کمک کننده است و مدیران پروژه‌ای که با داده‌های شخصی کار می کنند باید این موارد را در نظر داشته باشند. در مورد چه استانداردی صحبت می کنم؟ اگر در اتحادیه‌ی اروپا هستید، مطمئناً حجم زیادی از فعالیت‌ها را در فصل بهار از دست نداده‌اید؛ زیرا کمپانی‌ها برای اجرای تغییرات آماده شدند تا آن استانداردها را پیاده‌سازی کنند. منظور همان GDPR است.

اما من با اروپا کار نمی‌کنم!

حتی اگر با مشتریان یا پرسنلی در کشورهای اتحادیه اروپا کار نمی‌کنید، پروژه‌ی بعدی شما ممکن است به‌نوعی به کشورهای اروپایی مرتبط شود و آن مشتریان یا اعضای تیم اروپایی شما انتظار خواهند داشت تا داده‌های مربوط به استخدام آنان را بر اساس قوانین GDPR داشته باشید. پروژه‌ی بعدی شما ممکن است از شما بخواهد تا با تولیدکننده‌ای کار کنید که در اتحادیه اروپا باشد یا خدمات اینترنتی را راه‌اندازی کنید که برای هر کسی از هر کشوری فعال باشد. با در نظر گرفتن اینترنت، آیا کسب‌وکارتان وب‌سایتی دارد که اطلاعات تماس در آن قیدشده باشد؟ و آیا مردم اروپا می‌توانند جزئیات خود را در آن قرار دهند؟ اگر پاسخ مثبت است، GDPR شامل شما نیز می‌شود. بیایید فرض کنیم که شما یک مدیر پروژه برای کسب‌وکار آجر و ملات هستید که اصلاً وب‌سایتی ندارید و کارکنان و مشتریانتان بسیار محلی هستند. GDPR تنها برای ایمن‌سازی داده‌های شخصی به کار می‌رود. اگرچه آن مقررات حفاظت از داده‌ها شاید هنوز در قوانین محلی شما اجرایی نشده باشد، اما من فکر می‌کنم که بسیاری از کشورهای دیگر فعالانه در حال محافظت از داده‌ها هستند.

GDPR چیست؟

GDPR مخفف مقررات محافظت از داده‌های عمومی است و مجموعه‌ای از استانداردهای حریم خصوصی داده‌ها به شمار می‌رود.

GDPR گروهی از قوانین حفاظت داده‌ها را برای کمپانی‌های اجرایی در اتحادیه‌ی اروپا فراهم می‌سازد و قوانین حفاظت داده‌ها را در کشورهای عضو اتحادیه اروپا هماهنگ می‌سازد. مزیت اصلی، آن است که مردم با کمک این استانداردها کنترل بیشتری روی داده‌های شخصی خود و آنچه برای آن اتفاق می‌افتد خواهند داشت. این قانون در تاریخ ۲۵ می ۲۰۱۸ به اجرا گذاشته شد. پس آیا این قانون تنها مختص اروپاییان است؟ هم بله و هم خیر. شهروندان اتحادیه اروپا تحت پوشش GDPR هستند یا به‌بیان دیگر مقررات موجود در کشورهای آنان GDPR را نیز شامل می‌شود. به‌عنوان مثال در انگلستان قانون ۲۰۱۸ محافظت از داده نسبت به قوانین GDPR متعهد است. GDPR در اروپا از اهمیت بالایی برخوردار شده است، اما بر کمپانی‌های خارج از اتحادیه اروپا که مشتریان بین‌المللی نیز دارند نیز تأثیرگذار بوده است؛ از جمله شهروندان اروپایی. اگر داده‌های افراد در اتحادیه اروپا را ذخیره یا پردازش می‌کنید، GDPR برای شما نیز کارآمد خواهد بود.

سرفصل‌های GDPR

سرفصل اصلی GDPR شفافیت است. کمپانی‌ها باید در مورد دلیل ذخیره‌سازی و پردازش داده‌های شما و دلیل استفاده از آن‌ها صریح باشند. اگر آنان حق نداشته باشند با شما در این زمینه به دلیل حفظ امنیت داده‌هایتان تماس بگیرند، پس ارتباط آنان با شما ممنوع خواهد بود. این ممنوعیت عبارت است از نفرستادن ایمیل‌های بازاریابی به آدرس ایمیل شخصی شما.

کمپانی‌ها نباید برای رسیدن به اهدافشان اطلاعات بیشتری نسبت به نیاز شما جمع‌آوری کنند. پروژه‌ی شما باید تنها داده‌های لازم برای وظایف محول شده را جمع‌آوری کند، چه آن وظیفه پیروی از یک دستور باشد یا مواجهه با یک شکایت. اگر در حال تهیه‌ی یک فرم آنلاین جدید برای رزرو نوبت در یک سالن خدمات ناخن هستید، نباید از مشتریان بخواهید در مورد غذاهای مورد علاقه‌شان بگویند مگر اینکه بخواهید در حین کار به آنان غذا بدهید.

GDPR برای افراد این امکان را فراهم ساخته تا داده‌های خود را بین کمپانی‌ها انتقال دهند. به طور مثال، اگر بیمه‌ی اتومبیلتان شامل تخفیفی نشده، می‌توانید سابقه‌ی کامل بیمه‌ی اتومبیل خود را از یک ارائه‌دهنده به دیگری انتقال دهید، به امید آنکه شرکت جدید، تجربه‌ی رانندگی پیشین شما را در نظر بگیرد.

GDPR مقرراتی است که بر عوامل بسیاری اثر می‌گذارد از جمله اینکه چگونه داده‌ها نگهداری و استفاده می‌شوند و چگونه کمپانی‌ها باید اطلاعات در مورد آن را با شما به اشتراک بگذارند. بررسی کنید تا ببینید کشور شما در مورد اجرای این استانداردها چه نکاتی اضافه کرده است.

پیش‌بینی می‌کنیم که حریم خصوصی داده‌های مربوط به GDPR با قانون‌گذاران در سراسر جهان در میان گذاشته شود. این قوانین و مقررات می‌توانند به‌زودی به کشور شما بیایند.

نکته: الیزابت هارین وکیل نیست و این مقاله مشاوره‌ی قانونی به شمار نمی‌رود.

رویـداد

مدیریت تغییر ۲۰۱۹

تاریخ: ۲۸ آوریل تا ۱ می ۲۰۱۹

مکان: اورلاندو، فلوریدا، ایالات‌متحده

مدیریت تغییر ۲۰۱۹ یکی از کنفرانس‌های سالانه‌ی مدیریت پروژه است که توسط انجمن مدیریت حرفه‌ای تغییر، سازمان‌دهی می‌شود. این کنفرانس به‌منظور پشتیبانی از رشد حرفه و شغل و اهداف، طراحی‌شده و بیش از ۱۰۰۰ شرکت‌کننده از جای‌جای جهان دارد. شرکت‌کنندگان این رویداد فرصت آن را دارند تا در بخش‌های مختلف کنفرانس مدیریت پروژه شرکت کنند، با سایر شرکت‌کنندگان ارتباط برقرار کنند و مهم‌تر از همه تخصص خود را ارتقا بخشند. علاوه بر این، آنان این فرصت را دارند تا از متخصصان در این زمینه مواردی را بیاموزند. بهترین کتاب‌های مربوط به مدیران پروژه، تمرینات رایج و آخرین فناوری‌های مدیریت پروژه تعداد کمی از موضوعات رایجی هستند که در خلال این کنفرانس به بحث گذاشته می‌شوند.

۱۰ راهکار برای مالکان محصول در چارچوب 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. با این حال 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 اسپرینت 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 اسپرینت 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 کردنِ ارزش محصول یاری می‌کند. اگر شما به تیم اسکرام کمک کنید آن‌ها نیز درعوض 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 مربوطه بک‌لاگ محصول، می‌توانید تمرکز بیشتری بر اسپرینت 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 مربوطه‌به 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. بااین حال 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 تیم توسعه خودسامان‌دهیِ 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 اقلام 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، هدفتان را 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، متحمل 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 تاحد 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 سریع‌تر وامی‌دارند. 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 منجر به 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 پرداختِ 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.

دِین 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. بااین‌حال 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 دادن افزایش محصول (Product Increment) بخشی از بررسی اسپرینت است؛ اما تنها هدف آن نیست! هدفِ «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 مربوط به 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. بااین‌حال مالکان محصول بسیاری را دیده‌ایم که به‌ندرت با اسکرام‌مستر خود همکاری می‌کنند. تأسف‌آور است! اسکرام‌مستر می‌تواند در هدایت ذی‌نفعان، تسهیل رویدادهای مربوط به 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 اسپرینت حضور داشته است. این سوء تفاهم 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، بایَد 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.

رویداد

کنفرانس مدیریت پروژه‌ی APM لندن ۲۰۱۹

تاریخ: ۲ می ۲۰۱۹

مکان: Park Plaza Westminster Bridge لندن

APM منشوری است برای تخصص در پروژه که به کنفرانس‌های مدیریت پروژه‌ی برنده جایزه معروف است. این کنفرانس‌ها باتجربه‌ترین و متخصص‌ترین سخنوران جهان را دورهم جمع می کنند تا افکار آنان را به اشتراک بگذارند. آنان روش‌های بی‌شماری را کشف و بررسی نموده و بهترین روش‌های مدیریتی را مشخص می کنند. اگر تصمیم دارید در کنفرانس مدیریت پروژه APM ۲۰۱۹ لندن شرکت کنید، باید با محیط کار پویا و همواره در حال تغییر، رهبری اثربخش و مهارت‌هایی که برای موفقیت در پروژه مطلوب است، آشنا شوید.

بازخورد (فیدبک) یک مکالمهٔ دوطرفه است، نه یک خیابان یک‌طرفه

برخی از ما همیشه حس خوبی به تغییر داریم. تغییر برایمان دلچسب است. تغییر را نشانی از پیشرفت، و بخش لازمی از هر جنبهٔ زندگی می‌دانیم. اولین کسانی که به این نگرش دست یافت هراکلیتوس، فیلسوف یونانی پیشاسقراطی است (۵۸۰-۴۸۰ ق.م) که می‌گوید:

"هیچ کس هرگز در یک رودخانه دوبار قدم نگذاشته است، چرا که نه آن رودخانه همان رودخانه است و نه او همان کَس".

مدیریت تغییر از آن جهت دشوار است که تغییرآور با مقاومت روبه‌رو می‌شود. این است که مدیر پروژه باید مطمئن شود که اعضای تیم پروژه انتظار تغییر را دارند، می‌پذیرندش و با آغوش باز به آن خوشامد می‌گویند. اما چگونه؟

یک رابطهٔ پُراعتماد را بناگذاری کنید! درواقع نقطهٔ شروعْ روابط است، اینکه چطور حسِ اعتماد را در تیمتان القا کنید. شاید می‌پرسید دقیقاً چطور؟ پاسخْ ارتباط سازنده و پیوسته است ... به عبارت دیگر، بازخورد!

چگونه بازخورد (فیدبک) همچون یک ابزار ضروری برای پیشبرد مؤثرانهٔ تغییر عمل می‌کند؟ درادامه نکاتی را دربارهٔ نحوهٔ بازخورددهی ارائه می‌کنیم.

مزایا

- با بازخورد پیوسته امکان مکالمه می‌یابید و روابط بهتری را بنا می‌نهید؛
- شما که مدیر پروژه هستید، با بازخورد می‌توانید تیمتان را هدایت کنید و درعوض شاید بتوانید پیشنهادهای بسیار جالبی را دربارهٔ فرایندها و مانند آن از تیم پروژه بگیرید. بازخورد یک سناریوِ بردبرد فراهم می‌کند؛
- بازخوردِ مداوم همواره به‌معنای آموختن مداوم است؛ زیرا همه می‌توانند نقاط ضعف و قدرتشان را دوباره ارزیابی کنند؛
- بازخورد دائماً شما را به یک فرهنگ مبتنی بر عمل هدایت می‌کند؛ چرا که از اشتباه‌های خود می‌آموزید و خودآگاه می‌شوید. نتیجهٔ آن، نیروی کاری انطباق‌پذیر و با کیفیت بودن عملکرد آنهاست؛
- موجب رضایت کارفرما می‌شود، درنتیجه مشارکتش بیشتر می‌شود و عملکردش بهبود می‌یابد.

حال که مزایای بازخورد را شناختیم، سؤال این است که چگونه می‌توان بازخورد مفید و صادقانه‌ای از تیم پروژه گرفت؟ بی‌شک کار آسانی نیست، اما بایدها و نبایدهایی که در ادامه ارائه می‌کنیم شایستهٔ توجه‌اند.

چطور بازخورد بگیریم؟

بازخورد گرفتن از تیم پروژه از جمله کارهایی است که گاهی خوشایند نیست. نگرانیم که شاهد چه واکنشی خواهیم بود. اگر مسئله را انکار کنند چه؟ اگر روابط موجود را تخریب کنند چطور؟ چه کنم‌های بسیاری به ذهن ما خطور می‌کند. با این همه می‌دانیم که بازخوردگیری دائمی از تیم پروژه برای بهبود عملکرد و مشارکت و پیشروی‌شان ضروری است. از طرفی، همهٔ ما طعم نامطبوع بازخوردهای نامناسب را چشیده‌ایم و دریافته‌ایم که بازخورددهی معمولاً با زبانی پرطمطراق و پیچیده همراه است. گویا این ناشی از طبیعت انسان است؛ همه می‌خواهند خوب باشند. متأسفانه این گونه نمی‌توان پیام صحیح را رساند؛ چه این پیام مثبت باشد چه منفی. منظور آن است که چه‌بسا بازخورد انتقادی و صادقانه دردآور باشد. اما در نهایت برآوردن نیازهای سازمانی شما در گروِ آن است و از خوشخویی ناصادقانهٔ تیم پروژه ارزش بیشتری دارد. وقتی به کسی بازخورد می‌دهید، خوب است که توصیه‌های زیر را درنظر داشته باشید:

- رفتار را درنظر بگیرید، نه شخص را؛ فاصلهٔ معینی را حفظ کنید و بی‌طرف باشید؛
- مثال‌های مشخصی بزنید؛ این نکته، هرچند برای درک کامل بازخورد اهمیت دارد، اغلب فراموش می‌شود؛
- رفتاری را که انتظار دارید تشریح کنید؛ با راهنمایی یا آموزش توضیح دهید چگونه فرد می‌تواند آن رفتار را بهتر کند؛
- بازخوردها را به‌هنگام و به‌طور خصوصی مطرح کنید؛ فراموش نکنید که یک موضوع شخصی است؛ لازم نیست دیگران را مطلع کنید.

توصیهٔ آخر توصیه‌ای متداول و مستقیم است؛ اما می‌توان به شیوهٔ راهبردی‌تر و مبتکرانه‌تری بازخورد داد؛ سعی کنید محورِ بازخورد را کار لازم برای بهترسازی قرار دهید نه کار لازم برای متوقف‌سازی. این رویکرد عاقلانه‌تر است و افراد کمتر آزرده می‌شوند. خودتان را به‌جای آن‌ها بگذارید و کمک و راهنمایی‌های موردنیازشان را فراهم کنید. فراموش نکنید که کمبودِ بازخوردِ مناسب به‌معنای ناتوانی مدیر پروژه در تعریف جایگاه شخص، تیم پروژه یا پروژه است. اگر بازخورد را نتوان طوری انتقال داد که بر فردِ مربوطه اثرگذار باشد بی‌فایده است. هم بازخوردگیر و هم بازخورددهنده باید بتوانند دیدگاه‌های یکدیگر را بفهمند و به آن‌ها بها بدهند. بازخورد اساساً مکالمه‌ای صادقانه و شایسته است، نه یک مباحثه یا مجادله؛ به‌معنای بررسی مسائل و اطلاعات موجود با هدفِ توافق کردن بر کار لازم است. بازخورد جز، پرورش یک ذهنیت تعاملی و همکارانه برای رسیدن به ادارک و اهدافِ مشترک نیست.

می‌توان این مقاله را با لیستی از توصیه‌ها ادامه داد اما کفایت می‌کند که بدانیم «تنها دلیل بازخورد دادن به افراد کمک به پیشرفت و موفقیت آن‌هاست.»

بنابراین قبل از قصدِ مکالمه با کسی برای بازخورد دادن، لحظه‌ای درنگ کنید و در انگیزه‌تان تأمل کنید؛ بازخورد به افراد و علایقشان برمی‌گردد. فراموش نکنید که قصدتان بنانهادن یک رابطهٔ پراعتماد است. انسان باشیم؛ پذیرا و یادگیرنده.

کاربران مسئله و مشکل طرح می‌کنند و توسعه‌دهندگان راهکار می‌دهند

چگونه به اهداف دست یابیم؟

در دنیای نرم‌افزار، این کارِ توسعه‌دهندگان نرم‌افزار است که مشکلات کاربران را حل کنند. کاربران مشکلی را مطرح می‌کنند و توسعه‌دهندگان آن را برطرف می‌کنند. اما تعویض نقش این دو نقش موجبِ دردسر می‌شود.

اگر مایلید که یک قطعه نرم‌افزار پیچیده و بی‌مصرف و پرزرق‌وبرق داشته باشید، توسعه‌دهندگان را به پیاده‌سازی هر راهکاری که کاربر پیشنهاد می‌کند وادارید. بی تردید، این کاربران هستند که مشکل را می‌دانند؛ آن‌ها ایده‌های بکری برای راهکارها دارند. اما همیشه تصمیم‌گیرندهٔ نهایی دربارهٔ نحوهٔ حل مشکل توسعه‌دهنده‌های سیستم هستند، نه کاربران آن.

این مشکل خصوصاً وقتی شدت می‌گیرد که برای کاربران داخلی یک سازمان یک نرم‌افزار می‌سازید. کاربرانی که نرم‌افزار را برای آن‌ها می‌سازید، بر شما تسلط دارند زیرا مدیران اجرایی شما هستند یا به مدیران اجرایی نزدیک‌اند؛ اختیار کامل دارند و می‌توانند به شما بگویند چه کنید. بااین‌حال اگر راهکاری بخواهند که واقعاً کارساز باشد باید تغییرِ روش بدهند. اگر به یک تیم آن‌قدر اعتمَاد داشته‌اید که ساختِ نرم‌افزارتان را به آن‌ها محول کرده‌اید، باید به تصمیم‌گیری آن‌ها دربارهٔ نرم‌افزار نیز اعتماد کنید. اگر به آن‌ها اعتماد ندارید چرا در شرکت خود استخدامشان کرده‌اید؟ گروهی که به هم بی‌اعتماد باشند معمولاً بازدهی کمی دارند؛ شاید درواقع یک گروه نباشند، تنها مجموعه‌ای از افرادی باشند که سعی می‌کنند از خودشان دربرابر دیگران دفاع کنند. به این شیوه نه می‌توان یک سازمان را اداره کرد و نه یک زندگی شاد داشت.

اگر کاربری بخواهد بر تصمیم توسعه‌دهنده اثرگذار باشد بهترین کار ارائهٔ داده‌هاست. برای اینکه توسعه‌دهنده‌ها بتوانند تصمیم‌های

صحیحی را برای کاربران اتخاذ کنند به اطلاعات نیاز دارند و این اطلاعات معمولاً از خود کاربران حاصل می‌شود. اگر کاربر گمان می‌کند که نرم‌افزار به جهت اشتباهی می‌رود، برای حل مشکل، اطلاعاتی را مطرح کند که مایل به رسیدگی به آن‌هاست و توضیح دهد که چرا نرم‌افزار فعلی قادر به حل آن نیست؛ اطلاعاتی دربارهٔ تعداد افراد دیگری که همین مشکل را دارند ارائه کند. بهترین کار ارائهٔ اعداد و ارقام است؛ اما گاهی یک داستان خنده‌دار و کوتاه نیز می‌تواند توسعه‌دهنده را در تصمیم‌گیری یاری کند. توسعه‌دهنده‌ها باید تشخیص درستی از داده‌ها داشته باشند؛ شکی نیست که داده‌های عینیِ کاربرانْ بسیار مؤثرتر از داستان یکْ کاربر است. درکل توسعه‌دهندگان پذیرای هر اطلاعاتی که به آن‌ها داده می‌شود هستند، اما وقتی به‌صورت داده باشد، نه به‌صورت درخواستِ یک راهکارِ معین.

ازطرفی توسعه‌دهندگان نیز گاهی معکوس همین مشکل را ایجاد می‌کنند. وقتی که توسعه‌دهندگان به‌سادگی تصور می‌کنند که کاربران مشکلی دارند و سپس شروع به توسعهٔ یک راهکار برای آن مشکل می‌کنند، نرم‌افزاری خواهید داشت که کاربران از آن متنفرند. مشکلات و مسائل را کاربران مطرح می‌کنند، نه توسعه‌دهنده‌ها. گاهی توسعه‌دهندهٔ یک نرم‌افزارْ کاربر آن نیز هست و می‌تواند مشکلاتی را که خودش تجربه می‌کند به‌وضوح تشخیص بدهد. خوب است؛ اما باید آن را به‌صورت داده و از دیدگاه یک کاربر مطرح کند و مطمئن باشد که بقیه نیز این مشکل را دارند. توسعه‌دهندگان باید اعتقادات خود را به‌نحوی باارزش‌تر از یک کاربر متوسط بدانند (زیرا بازخوردها (فیدبک‌های) زیادی از کاربران دریافت می‌کنند و مدت‌های مدیدی با برنامه‌ها کار کرده‌اند) اما همچنان باید آن را فقط ایده‌ای از سمت کاربر در نظر بگیرند.

وقتی به‌جای مسائل کاربران، مسائل توسعه‌دهندگان را حل می‌کنید، تلاشتان به نتیجه‌ای که به‌بهترین نحو به دیگران کمک می‌کند ختم نمی‌شود. شاید طرح ایده‌ها لذت‌بخش باشد و مشکل تیم را حل کند، اما عرضهٔ نرم‌افزاری که کمکی به افراد نمی‌کند حس بدی دارد. همچنین معمولاً حل مسئلهٔ توسعه‌دهندگان به پیچیدگی‌های بیشتری منجر می‌شود تا حل مسائل کاربران. بنابراین یافتن مشکل و برطرف کردن آن ازطریق کاربر آسان‌تر است تا ازطریق تصور یک مشکل و فلسفه‌بافی حول آن.

منظور این نیست که توسعه‌دهندگان هیچ‌گاه نمی‌توانند واقعاً یک مسئله یا مشکل طرح کنند؛ یا هیچ کاربری هرگز نمی‌تواند راهکار مناسبی بدهد. گاهی شدنی است. اما باید برحسب اکثریت تشخیص داد. فقط کاربران (و ترجیحاً تعداد زیادی از آن‌ها یا داده‌هایی از تعداد زیادی از آن‌ها) می‌توانند مشکلِ تجربه‌شده را نشان دهند.

به‌همین‌صورت فردی ازجانب تیم توسعه (ترجیحاً فردی که وظیفه‌اش تصمیم‌گیری بعد از شناخت مشکل و احتمالاً بازخورد گرفتن از همتایان خود است) می‌تواند به‌درستی تصمیم بگیرد که چه راهکاری پیاده‌سازی شود.

محمدرضا اسماعیل‌طهرانی

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 پروژه‌به‌پروژه دستخوش تغییر خواهد شد؛ کمی دشوار است و نیاز به همراهی فرهنگ قویِ سازمانی دارد. این فرهنگ قطعاً و صرفاً توسط مدیرکل مجموعه قابل اجرا و استقرار کامل خواهد بود و انتظار اینکه به‌صورت ناگهانی همه موارد فوق به‌صورتِ کامل و صددرصدی برقرار شود، کاملاً اشتباه است. توجه شود که موارد بیان‌شده تنها بخشی از مهمترین وظایف دفتر مدیریت پروژه بوده 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.

روزانه

.000.000.000.000

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: ۰۲۱ ۸۸۶۵۹۴۶۸-۷۰
وب سایت: www.dadehpardaz.com/seo

عامل موفقیت پروژه‌ها چیست؟

داشتن یک مدیر بااستعداد پروژه اولین قدم موفقیت یک پروژه است اما عوامل مهم دیگری هستند که بر خروجی پروژه جداً تأثیرگذار است. مدیریت پروژه مستلزم برنامه‌ریزی دقیق، توجه به جزئیات و ارتباطات مؤثر است. با مدیریت هوشمندانه و تکمیل پروژه، شرکت‌ها به موفقیت‌های دائمی دست می‌یابند. اما چندی از عوامل موفقیت پروژه‌ها را بررسی کنیم.

۰۱ افراد باهوش

بدون یک تیم مناسب، هر راهبرد و برنامه‌ای ممکن است کاملاً از هم بپاشد. به این دلیل، پرسنل اصلی پروژه، منابع کارشناس، تأمین کننده‌ها و تمام ذی‌نفعان باید بخش پویایی از تیم باشند. تمام کسانی که در پروژه مشارکت دارند باید به گروه متعهد باشند، بینش یکسانی به پروژه‌ها داشته باشند و برای موفقیت همگانی تلاش کنند. اگر تیم صلاحیت کافی نداشته باشند مدیران پروژه با مشکلات جدی روبه‌رو می‌شوند. رهبری بی کفایت یا تیم ناهماهنگ می‌تواند یک پروژه را به درّهٔ سقوط هدایت کند. اهمیت تعیین اهداف مناسب برای هر جنبهٔ پروژه و ضرورت همکاری‌ها را دست کم نگیرید. به‌علاوه کل تیم باید کاملاً از روند کار مطلع بمانند و برای بهترین نتیجه مشارکت کنند؛ ازاین‌رو ارتباطات باید دوجانبه باشند و افراد برای ارتباطات فرصت برابری داشته باشند. از یک ابزار نرم‌افزاری آنلاین و یک ذخیره‌گاه مرکزی اطلاعات استفاده کنید. ابزاری را انتخاب کنید که نه‌تنها امکان اجازهٔ مدیریت لیست بکن‌ها (to do list) و برنامهٔ پروژه را دارد، بلکه برای شما امکان دسترس‌پذیری به منابعتان را فراهم می‌کند؛ و به‌این‌ترتیب زمانشان را برای پروژه‌ها و دیگر وظایف حفظ می‌کنید. با ثبت زمان منابع نه‌فقط سیستم قدرتمندی دارید که برگه‌های زمانی و صورت‌حساب‌ها را خودکار می‌کند بلکه مانع ازهم‌پاشی فرایندهای پیش رِو پروژه می‌شوید.

۰۲ برنامه‌ریزی هوشمندانه

برای موفقیت پروژه ضروری است که از ابتدای پروژه یک برنامه‌ریزی فراگیر داشته باشید. تمام ذی‌نفعان باید حینِ روند برنامه‌ریزی هماهنگ باشند و همیشه بِدانند پروژه به چه جهتی می‌رود. برنامه‌ریزی به تیم کمک می‌کند به موعدهای تحویل برسد و سازمان‌دهی خود را حفظ کند؛ نه‌تنها تیم پروژه را متمرکز و هماهنگ می‌کند، بلکه ذی‌نفعان را از پیشروی پروژه آگاه نگه می‌دارد. برنامه‌ریزی هوشمندانه منافع بسیاری دارد. اگر برنامه‌ریزی را قدم اول خود قرار بدهید، می‌توانید مقیاس زمانی واقع‌بینانه و قابل اعتمادی پیدا کنید و با مستندات شفافِ مربوط به رویدادگاه‌ها (مایلستون‌ها) و اقلام تحویلی، پیشبرد امور پروژه را آسان کنید. یک برنامهٔ حرفه‌ای تمام نیازمندی‌های منابع را با جزئیات درنظر می‌گیرد و به‌صورت یک سیستم هشدار عمل می‌کند. وقتی خطر از قلم‌افتادن اقلام تحویلی و وظایف پروژه وجود دارد، با یک سیستم هشدار می‌توانید انتظارات پروژه را شفاف‌سازی کنید. از ابزاری استفاده کنید که خط زمانی‌اش امکان کشیدن و برداشتن (drag and drop) داشته باشد تا بتوانید سریعاً یک برنامهٔ پروژه ایجاد کنید و یک تاریخ نهایی دقیق تعیین کنید. از پروژه‌های تکمیل‌شدهٔ قبلی به‌عنوان پیش‌الگو (template) برای پروژه‌های آینده‌تان استفاده کنید. به‌این‌ترتیب لازم نیست که برای ساخت یک پروژه از اول دوباره کاری کنید؛ افزون بر آن هرچقدر از یک پیش‌الگو بیشتر استفاده می‌کنید، تخمین‌های بودجه‌ای و زمانی‌تان دقیق‌تر خواهند بود.

۰۳ ارتباطات باز

برای موفقیت یک پروژه بررسی دقیق جزئیات و منابع خارجی اطلاعات ضروری است. باز نگه داشتن ارتباطات در تیم قطعاً ضرورت دارد. وقتی کار را طِبق یک جدول زمانی معین به‌پیش می‌برید، تیم باید کاملاً مطلع بماند. درصورت پیشامد مشکلی در بخشی از پروژه، بخش‌های دیگر از آن اثر می‌پذیرند. ارتباطات بهترین روش برای ممانعت از وقوع مشکلات است. به‌علاوه ارتباطات باید به داخل سازمان معطوف بماند. با ثبت تاریخچهٔ پروژه‌های بزرگ می‌توان فرایندهای تجاری و سیاست‌های سازمان را بهبود داد. اگر بیشینهٔ پروژه‌های بزرگ ثبت نشوند، تیم ممکن است که اشتباهات قبلی را تکرار کند. گوش دادن و توجه کردن به ذی‌نفعان عامل اصلی موفقیت است. همچنین، اینکه بدانیم چه‌موقع نه بگوییم بخشی از ارتباطات مناسب است. تیم پروژه هرگز نباید قول چیزی را بدهد که نمی‌تواند تحویل بدهد. اگر، ازاول، کار نامتناسب با توان تیم را نپذیرید، از حجم انبوهی از مشکلات غیرضروری بعدی جلوگیری می‌کنید. همیشه دربارهٔ توانایی‌ها و ناتوانایی‌های تیمتان صادق باشید. ابزاری را بیابید که علاوه‌بر امکان کشیدن و گذاشتن گاهشمار، هنگام تعیین گاهشمار پروژه، بتوان از پروژه‌های قبلی به‌عنوان پیش‌الگو استفاده کرد. به‌این‌ترتیب، هم می‌توانید که با گذر زمان روال‌ها و روندهای کاری را بهتر کنید (تخمین‌هایتان دقیق‌تر شوند و انتظارات مشتری را برحسب تخمین‌هایتان بهتر بسنجید)، و هم ارتباطات بین تمام مشارکت‌کنندگان در پروژه را بهبود بدهید.

۰۴ مدیریت دقیق ریسک

مدیران پروژه می‌دانند که به‌ندرت پروژه‌ها طبق برنامه‌ریزی پیش می‌روند. در فرایند برنامه‌ریزی ایجاد یک ثبتِ ریسک یا risk log و داشتن یک برنامهٔ عملی برای ریسک‌های احتمالی پروژه ضروری است. تمام ذی‌نفعان را حتماً از ثبتِ ریسک مطلع نگه دارید؛ آن‌ها باید بدانند که چطور می‌توانند ثبتِ ریسک را مشاهده کنند. اگر اتفاقی بیفتد، تیم می‌تواند مسئله را سریعاً با برنامهٔ مدیریت، که ازقبل معین شده، حل‌وفصل کند. به‌این‌طریق تیم برای روبه‌رویی با ریسک‌ها اعتمادبه‌نفس پیدا می‌کند و مشتری‌ها را از پیشروی پروژه مطمئن می‌سازد. با داشتن یک پایگاه‌داده از اطلاعات پروژه می‌توان مطئن شد که انگیزه‌های اصلی پروژه حینِ کار ازدست نمی‌رود. اگر یکی از اعضای کلیدی را ازدست دادید، سریعاً می‌توانید عضو جدیدی برای تیم انتخاب کنید.

۰۵ جدیت در اعلام پایان پروژه

اگر پروژه را با جدیت پایان ندهید، امکانش هست که مصرف منابع ادامه بیابد. تیم پروژه باید جدی برخورد کند و با مشتری توافق کند که تمام عوامل اصلی موفقیت پروژه تحویل داده شده‌اند. باید بر تحویل پروژه، آزمون و عرضه توافق کرد و با امضا تأییدشان را گرفت. با نظرخواهی دربارهٔ رضایتمندی می‌توانید تجربیات پروژه را ثبت و مستند کنید تا برای استفادهٔ آینده بتوان به آن رجوع کرد و اطلاعات باارزشی یافت. مدیر پروژه موظف است که مطمئن شود همه‌چیز به‌خوبی پیش می‌رود؛ اما داشتن مدیر پروژه ضامن موفقیت خروجی پروژه نیست. برای رسیدن به موفقیت حقیقی، کل تیم باید به عوامل کلیدی توجه کنند. ماحصل این موفقیت برخورداری از برنامه‌های سازمان‌یافته و پیشرفتهٔ بعدی، و نتیجتاً، بهسازی کیفیت پروژه‌های آینده است.

چگونه یک برنامه‌ریز موفق و عالی باشیم:

آگاهی، ادراک و مسئولیت‌پذیری

سه عامل کلیدی لازم است تا یک برنامه‌ریز عالی شوید: آگاهی، ادراک و مسئولیت‌پذیری

من در مورد موضوع ادراک بسیار صحبت کرده‌ام. حتی جدیدترین کتابم را نرم‌افزار ادراک نامیدم. درواقع من به‌کرات اشاره کرده‌ام که هر چه‌بهتر کاری را درک کنید، بهتر می‌توانید آن را انجام دهید.

بااین‌حال، دو عامل دیگر نیز وجود دارد که در راستای ادراک قرار دارند تا فردی را به یک ارائه کننده‌ی عالی نرم‌افزار مبدل سازند. خلاصه اینکه اگر از مشکلی آگاهی نداشته باشید، ادراکی وجود نخواهد داشت. همچنین اگر برای حل مشکل اقدامی نکنید (که با پذیرش مسئولیت شروع می‌شود) نمی‌توانید کاری در مورد آن انجام دهید.

هرچند در مورد هر یک از این نکات چیزهای زیادی باید بدانید و برای کسی که می‌خواهد ارائه‌دهنده‌ی نرم–افزار بهتری باشد، پیروی از این نکات بسیار مهم خواهد بود. بنابراین می‌خواهم در اینجا هر یک از این موارد را با جزئیات برایتان شرح دهم.

آگاهی

در حقیقت آگاهی نوع به خصوصی از ادراک است. آگاهی لزوماً یعنی اینکه شما بتوانید چیزی را بفهمید و بدانید که وجود دارد. روش‌های زیادی وجود دارد که به برنامه‌نویسی مربوط می‌شود.

اجازه دهید یکی از ساده‌ترین روش‌ها را انتخاب کنیم: اگر از وجود یک باگ نرم‌افزاری آگاه نباشید، احتمالاً نمی‌توانید آن را اصلاح کنید.

گاهی اوقات افراد این روش را به کار می‌برند تا دیگر مجبور نباشند کاری برای حل مشکل انجام دهند. اگر از آن آگاهی نداشته باشند، در مورد آن فکر می کنند و بنابراین مجبور نیستند چیزی در مورد آن بدانند. این کار شاید درست باشد شاید هم نه، اما حقیقت آن است که مشکل وجود دارد چه شما از آن اطلاع داشته باشید یا نه. اگر سیستم شما باگی داشته باشد که میلیون‌ها کاربر را تحت تأثیر قرار دهد، باید از آن آگاهی داشته باشید. در غیر این صورت کاربران شما دچار مشکل می‌شوند. ناراحتی آنان با ناآگاهی شما برطرف نمی‌شود.

انواع ظریف‌تری از آگاهی وجود دارد. به‌عنوان مثال، برخی افراد واقعاً از پیچیدگی کد آگاهی ندارند. آنان نمی-دانند کدی که آن را نوشته‌اند پیچیده است. گاه از این موضوع که پیچیدگی، خود یک مشکل است آگاهی ندارند یا حتی از دیدگاه برنامه‌نویس دیگری که آن کد را می‌خواند آگاه نیستند. آنان نمی‌بینند که کد، مواردی را از دست می‌دهد، که به‌سختی خوانده می‌شود یا اینکه روش دیگری برای حل این مشکل وجود دارد. تمامی این موارد جوانبی از آگاهی برای یک برنامه‌نویس هستند. ساده‌ترین روشی که می‌توانید با آن آگاهی خود را به‌عنوان یک برنامه‌نویس بهبود بخشید آن است که بیرون بروید و کدی را که بیرون قلمرو کنترل شما قرار دارد بخوانید. به‌عبارت‌دیگر ممکن است روزانه شما روی کدی کار کنید که ویدئو را رمزگشایی می کند. اما آن ویدئو نیز در جای دیگری تولیدشده است؛ مثلاً کاربران آن را در سیستم شما آپلود کرده‌اند. اگر با مشکلی مواجه شوید که چگونه این ویدئوها وارد رمزگشای شما شده‌اند، شاید بهتر

باشـد بروید و ببینید کـه آن کـد چگونه کار می‌کند.
این کار، سـاده به نظر می‌رسـد اما برای بسیاری از افراد از نظر احساسـی سخت است. فکر کردن بـه آن راحت اسـت. می‌توانید بگویید: همه‌ی کسـانی که آن بیرون هسـتند احمق‌اند من گند زدم و فقـط بایـد روی آن کار کنم. اما چگونه این را می‌دانید درحالی‌که به کد آن‌ها حتی نگاه هم نکرده‌اید؟ شـاید چیزی باشـد که آموختن آن به شما کمک کند.
آگاهی همچنین می‌تواند در مورد دانستن وجود کتاب‌خانه‌هـا (libraries)، چارچوب‌هـا یـا ابزارهایـی باشـد که می‌توانیـد در کدتان به کار بریـد. شـاید همه‌چیـز را در مـورد جدیدتریـن کتابخانه‌هـای مجـازی ندانیـد، امـا اگـر یک سازنده‌ی وب هستید، باید حداقل از وجود آن‌ها آگاهی داشـته باشـید. بنابراین زمانی که باید در مـورد چارچوبـی کـه به کار می‌بریـد تصمیمی بگیریـد، می‌دانیـد کدام یـک مـورد بازدید قرار خواهـد گرفـت. در کل، می‌توانید تنها با تمایل به تجربه‌ی زبان‌های برنامه‌نویسی جدید، الگوهای طراحی جدید، ایده‌هایی در مورد تسـت کردن، چارچوب‌های جدید و غیره، آگاهی خود را بهبود بخشـید. من شـخصاً مایلم در مورد سیستم‌ها و متدهای جدید برنامه‌نویسـی بدانم. زمانی که در مورد یک زبان جدید چیزی می‌شنوم که قدرتمند اسـت، حداقل صفحه اولیه‌ی وب‌سـایت آن را می‌خوانم و گاه حتی در اطلاعات و محتوای آن غرق می‌شـوم. این گونه حداقل از وجود آن چیز مطلـع می‌شـوم و می‌توانم در صـورت لزوم در مورد آن بیشتر بیاموزم. اگر از روش‌های مختلف حل مشکلات آگاه نباشم، به‌عنوان یک مهندس نرم‌افزار، ابزارهای بسـیار کمتری خواهم داشت. افزایـش آگاهی‌هایـم به ایـن روش گزینه‌های انتخابی من را هنگام مواجهه با شرایطی سخت، بیشـتر می‌کند. تمام این موارد شاید ساده به نظر برسـند، امـا ایـن جملـه حقیقت دارد که: شـما می‌توانید با افزایش آگاهی‌تان از برنامه‌نویسی و برنامه‌هـا، توانایـی خـود را به‌عنـوان یـک برنامه‌نویس بهبود بخشید.

ادراک

همان‌طـور کـه قبـلاً گفتم: هر چـه ادراکتان از کاری بهتر باشـد، بهتر می‌توانید آن کار را انجام دهید.
آگاهی یک گام اولیه اسـت، شـما می‌دانید که چیـزی وجـود دارد. پـس از آن، می‌خواهید آن را درک کنید تا به‌خوبی انجامش دهید. اجازه دهید مثالی در مورد ساخت زبان کد نویسی بزنم:

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

به نظر شـما این برنامه چه کاری انجام می‌دهد؟ ظاهـراً پایانـه را پاک و سـپس کلمه‌ی Hello را پرینت می‌کند. اما صادقانه، فقط حدس می‌زنم. مـن بـه مـدارک یـا کـد clear_screen یا write_Hello نـگاه نکـردم. اگـر بـه کد نگاه می‌کردم، می‌دیدم که دستور clear_screen درواقـع کل صفحه‌نمایـش را سـفید می‌کند و write_Hello کلمـه‌ی HELLO را بـه پانزده زبـان و رنـگ مختلـف روی نمایشـگر پرینت می‌نمایـد. امـا مـن هرگز نتوانسـتم بدون نگاه کردن به چیزی آن را بشناسـم تا بتوانم درکش کنم.

ممکن است مانند یک مثال بسیار ساده‌شده به نظر برسد، اما همیشه این اتفاق در برنامه‌نویسی رخ می‌دهد. حالا که می‌دانم clear_screen چه کاری انجام می‌دهد، می‌توانم آن را هر جای دیگری به کار ببرم بدون اینکه مجبور شوم در مورد آن فکر کنم یا زمانی را صرف نوشتن آن نمایم. من می‌توانم برنامه‌ای بنویسم که آنچه می‌خواهم را انجام دهد، به‌جای اینکه برنامه‌ای پر از باگ تولید کنم. من می‌توانم تستی برای آن بنویسم که معنا و مفهوم داشته باشد. تمامی این کارها سادگی، ثبات و کیفیت بالا را افزایش می‌دهند تا درمجموع برنامه‌نویسی خوبی داشته باشیم. این موفقیت، از ادراک ناشی می‌شود.

شکل‌های دیگری از ادراک نیز وجود دارد که در چندین مقاله به آن پرداخته‌ام، به‌ویژه در مقاله‌ی «چرا برنامه‌نویسان گند می‌زنند». این موضوع واقعاً نکته‌ای کلیدی است که یک برنامه‌نویس را می‌سازد و یا در هم می‌شکند. مهارت‌های دیگری نیز وجود دارد که مفید هستند، مانند تایپ سریع، ارتباط خوب، مدیریت مناسب داشته‌ها و بسیاری مهارت‌های دیگر. اما نکته‌ی کلیدی ادراک است. بدون آن شاید بتوانید سریع‌ترین برنامه-نویس دنیا باشید، اما تنها سیستم‌های اشتباه و غیرقابل نگهداری را می‌نویسید. شاید بتوانید جذاب‌ترین فرد در تیم خود باشید، اما درنهایت نمی‌توانید حتی یک خط کد نیز بنویسید. من نمی‌گویم که داشتن سرعت عمل یا جذابیت، خوب نیست، بلکه این‌ها مهارت‌های سودمندی هستند، اما بدون ادراک، این مهارت‌ها شما را به برنامه‌نویسی عالی مبدل نمی‌کنند.

مسئولیت‌پذیری

من در این مورد در مقاله‌ای روی وبلاگ O'Reilly کمی صحبت کرده بودم و به‌طور گذرا بعد از اینکه اولین کتابم منتشر شد در این مورد نوشتم، اما مطمئن نیستم که چه تعداد از خوانندگان سایت codesimplicity.com این پست را خوانده‌اند، اگرچه این موضوع در مورد یک برنامه‌نویس خوب و عالی بودن بسیار مهم و کلیدی است. حال من می‌خواهم چیزی را برایتان شفاف‌سازی کنم. منظور من از داشتن احساس مسئولیت، سرزنش کردن در مورد چیزی که اشتباه شده نیست. این تعریف رایج و معمول مسئولیت‌پذیری است، اما تعریفی که من به کار می‌برم این نیست. منظور من از مسئولیت‌پذیری، تمایل داشتن به پذیرش کنترل نسبت به کار می‌باشد. به‌طور مثال، اگر من نسبت به اتاقم مسئولیت‌پذیر باشم، مایلم آن را تمیز کنم. این بدان معنا نیست که من مجبورم آن را تمیز کنم، بلکه من خودم مایلم این کار را انجام دهم. همچنین تنها این نیست که در قالب کلمات بگویم: بله من کاملاً مایلم اتاقم را تمیز کنم و بروم و سه ساعت دراز بکشم. خیر من واقعاً باید تمایل به این کار داشته باشم. مانند دسر موردعلاقه‌تان که با لذت آن را می‌خورید. من این را نمی‌گویم که شما باید برای هر کاری که انجام می‌دهید به‌اندازه‌ی خوردن یک دسر هیجان‌زده و خوشحال باشید، بلکه منظورم این است که باید یک تمایل واقعی و قلبی نسبت به این کار در شما وجود داشته باشد. پس چگونه می‌توان این مسئولیت‌پذیری را در برنامه‌نویسی پیاده‌سازی کرد؟ خب اغلب من این مثال منفی را مطرح می‌کنم:

من نمی‌خواهم این کد را دوباره بنویسم چون متعلق به من نیست.

این احمقانه است. آیا شما می‌توانید این کار را ویرایش کنید؟ آیا می‌توانید کد منبع را بخوانید؟ آیا مجازید تغییرات را به صاحبانش بفرستید؟ پس از نظر تئوریک قادرید آن را تعمیر نمایید. حال این بدان معنا نیست که شما باید تمام کد را در سرتاسر جهان و همیشه اصلاح کنید، چون گاهی اوقات زمان مناسبی برای رسیدن به اهدافتان نیست. اما از گفتن همین جمله هم نفرت دارم، زیرا در اغلب موارد مردم آن را به‌عنوان بهانه‌ای برای درست و کامل انجام ندادن کارشان به کار می‌برند.

«من نمی‌توانم وقتم را صرف پاک کردن کد آن تیم کنم زیرا موجب می‌شود پروژه‌ام طولانی‌تر شود». یا «خب پاک کردن این کدی که کار من بوده، جزو اهداف تیم من نیست.»

این تئوری‌ها دو ایراد دارند:

اولین ایراد آن است که معمولاً درست انجام دادن کارها هنگامی که کد شما پیچیده و بد باشد، زمان بیشتری می‌برد. اغلب بازسازی و سپس نوشتن کد جدید زمان کمتری از شما می‌گیرد تا اینکه سعی کنید قابلیت و کد موردنظرتان را بر روی آن کتابخانه‌ی ترسناک بنویسید که اصلاً آن‌طور که می‌خواهید کار نمی‌کند. بنابراین شما معمولاً زمان زیادی روی اهدافتان نخواهید گذاشت. بنابراین دیگر تفاوت بین انجام دادن یک کار با انجام دادن واقعی و درست یک کار را خواهید دانست.

منظور از انجام دادن درست و واقعی کار چیست؟ خب

منظورم کاری است که نتیجه داشته باشد، کاری پر از باگ و اشکال نباشد، بتوانید آن را به‌راحتی نگه‌دارید، کل زحمات شما را نابود نکند و بتوانید کاری سودمند و پربازده ارائه دهید.
این‌گونه به مشکل نگاه کنید. بیایید تصور کنیم که دیواری را می‌سازید و از پایین دیوار آجرهای خراب را می‌چینید. سپس چند ردیف دیگر لایه آجر روی آن قرار می‌دهید. اما به لایه‌ی چهارم به بعد که رسیدید، آن آجرهای خراب شروع به شکستن می‌کنند. بنابراین شما آجرهای خراب و شکسته را وصله می‌زنید و روی آن‌ها چندتکه چوب خراب می‌گذارید. چند لایه‌ی دیگر آجر روی آن می‌گذارید و حالا دیوار به کل فرومی‌ریزد. حال آن دیوار را به میله‌های آهنی تکیه می‌دهید و به کارتان ادامه می‌دهید. اگر این روش را ادامه دهید، تمام زندگی و عمر خود را صرف نگه‌داشتن آن دیوار خواهید کرد. شما احتمالاً درنهایت آن را رها خواهید کرد و شخص دیگری نیز به خاطر بروز حادثه‌ای ناگوار (فروریختن دیوار) از بین خواهد رفت. مشکل اصلی درواقع ادراک شما خواهد بود که موجب شده تا از مواد ساختمانی نامناسب برای ساخت دیوار استفاده کرده‌اید.
زمانی که شما کد بی‌کیفیتی را در سیستم کاری خود استفاده می‌کنید، نرم‌افزارتان را به‌نوعی می‌سازید که به سرنوشت آن دیوار مبتلا خواهد شد. اما در این مورد کم‌تر احساس می‌شود، زیرا برنامه‌ها ساختار عظیم فیزیکی ندارند که جلوی چشمتان فروبریزد. اما بااین‌وجود، همان اصول ساخت در این نیز صدق می‌کند؛ یعنی زمانی که مشکلی با افزودن پیچیدگی به آن را حل می‌کنید، پیچیدگی سیستم را افزایش می‌دهید. زمانی که در عوض آن مشکلی را بازبینی و ویرایش می‌کنید، پیچیدگی سیستم کاهش خواهد یافت. من می‌خواهم اشاره کنم که فرقی نمی‌کند چه کسی در وهله اول این مشکل را ایجاد می‌کند. این حقیقت که کس دیگری مسبب آن است، هیچ‌یک از قوانین فوق‌الذکر را تغییر نمی‌دهد. همچنین فرقی نمی‌کند که چه کسی حالا مسئول این پیچیدگی است. اگر آن را بدون برطرف کردن مشکل بازسازی کنید، سیستم پیچیده‌تر می‌شود. اگر مشکل را برطرف کنید، از پیچیدگی سیستم خواهید کاست. شما انتخاب می‌کنید و این قدرت را دارید، می‌توانید مسئولیت‌پذیر باشید. بله گاه می‌توانید شخص دیگری را مسئول درست کردن آن کنید. اما با توجه به تجربه‌ی شخصی من، در اغلب موارد شما باید تمایل داشته باشید خودتان مشکل را مرتفع سازید.
شاید شما همه‌ی این موارد را بدانید. اما بازهم ممکن است فکر کنید که: خب من می‌توانم این کار کوچک را این بار کمی پیچیده‌تر کنم، زیرا فقط همین کار را خودم انجام می‌دهم. می‌دانید، گاهی اوقات شاید حق با شما باشد به‌ویژه اگر اضطراری برای شما به وجود آمده باشد که مجبور شوید کاری را برای مدتی کوتاه به‌صورت سطحی حل کنید. اما اغلب با انجام کار به روشی سطحی با آوری از دیوارتان مواجه خواهید شد که به کابوسی برای شما و افراد دیگری که از آن نگهداری می‌کنند تبدیل می‌شود. این پیچیدگی‌های کوچک و انتخاب‌های ناچیز عدم پذیرش مسئولیت هستند که درنهایت مشکلات و شکست‌های بزرگی را در پی خواهند داشت که مواجهه با آن از کسی ساخته نخواهد بود.
بنابراین زمانی که می‌گویم مسئولیت‌پذیری، بخش اعظم منظور من «تمایل داشتن به تغییر دادن چیزهایی در محیط پیرامونتان» است. قرار نیست این محیط، محیط پیرامون نامحدود باشد. شما می‌توانید در مکان مشخصی خطی بکشید و بگویید: فراتر از این خط، اگر اتفاقی بیفتد مشکل شخص دیگری خواهد بود. به‌طور مثال، در پروژه‌هایم اغلب خطی می‌کشم و می‌گویم: خب من روی کدی که مربوط به بیرون شرکت است کار زیادی انجام نمی‌دهم، زیرا برای من ارزش زمان گذاشتن ندارد. اما زمانی که احساس کنم منجر به بروز پیچیدگی می‌شود می‌توانم باگ‌های فایل‌بندی شده را در برابر زبان‌های برنامه‌نویسی انجام دهم یا اینکه ابزارهایی را بفرستم. به‌عبارت‌دیگر من معمار اولیه‌ی Bugzilla می‌شوم زیرا فکر می‌کنم چیزهای زیادی هست که باید در آن اصلاح شود. البته قرار نیست همه این کار را انجام دهند. اما با پذیرش مسئولیت کدهای بیشتر فراتر از محدوده‌ی پروژه‌تان، برنامه‌نویس بهتری خواهید بود. همچنین هر چه این محدوده را گسترده‌تر کنید، برنامه‌نویس بهتری خواهید شد.
به‌هرحال، من گفتم دو اشتباه در تئوری‌های فوق وجود دارد. می‌توانید کار را به افراد دیگر محول کنید. ما برنامه‌نویسان درواقع همگی عضو یک تیم هستیم.
اگر برای یک کمپانی کار می‌کنید، با درست کردن مشکلات زمانی که از پس آن‌ها برمی‌آیید، به کل آن شرکت کمک خواهید کرد. اگر شما فردی باشید که کارتان را به‌درستی انجام می‌دهید، زمانی که ابزار رایجی یا کدی را اصلاح می‌کنید، به سهم خود در اصلاح جهان کمک کرده‌اید. صادقانه بگویم، این کار از شما برنامه‌نویس بهتری می‌سازد. بهترین برنامه‌نویسانی که من می‌شناسم کسانی هستند که بیشتر از بقیه تمایل دارند مسئولیت بپذیرند تا کاری را به‌درستی انجام دهند.
به‌هرحال شما به‌عنوان یک مصرف‌کننده‌ی API، اغلب در بهترین جایگاه برای بازسازی و ویرایش آن هستید، زیرا بهتر از مؤلفان می‌دانید که چگونه API را باید مصرف کنید. دست‌کم یک باگ را در برابر کد فایل‌بندی کنید و بگویید که با آنچه مشکلی داشتید. در غیر این صورت چگونه مؤلفان می‌فهمند که مشکلی وجود دارد؟ شما تنها ممکن است از آن‌ها انتظار داشته باشید که از طریق علم غیب به آن پی ببرند! ولی باور کنید آن‌ها نمی‌دانند. تجربه‌ی شما می‌تواند برای آنان بسیار ارزشمند باشد.

خلاصه

در کل، اگر می‌خواهید یک برنامه‌نویس بهتری باشید، باید آگاهی، ادراک و مسئولیت‌پذیری خود را بهبود بخشید و سپس برای مدت کوتاهی روی آن تمرکز کنید. اگر اطمینان ندارید، با آگاهی شروع کنید، سپس تمرکز خود را روی ادراک قرار دهید و درنهایت به مسئولیت‌پذیری بیشتر فکر کنید. انجام این کار در جهت و ترتیب دیگر بسیار سخت است، به‌راحتی نمی‌توان مسئولیت کاری را بر عهده گرفت که آن را درک نکرده‌اید و ادراک چیزی که از آن آگاهی ندارید نیز غیرممکن است. بنابراین آگاهی، ادراک و سپس مسئولیت‌پذیری ترتیب درستی است.
اگر آگاهی چیزی است که باید در خود بهبود بخشید، تنها چند کد جدید را بخوانید، یک وبلاگ برنامه‌نویسی جدید بیابید، عناوین کتاب‌ها را جستجو نمایید، با برخی برنامه‌نویسان دیگر در مورد آخرین فناوری صحبت کنید، هر آنچه به شما کمک کند تا نسبت به مشکلات، راهکارها، دانش، الگوها، مردم، سازمان‌ها، اصول یا هر چیز دیگری که به کارتان کمک کند آگاهی بیشتری پیدا کنید.
اگر آنچه می‌خواهید روی آن تمرکز کنید قابل درک است، اسناد بیشتری را بخوانید، زمان بیشتری روی درک هر کاری بگذارید، سؤالات بیشتری از برنامه‌نویسانتان بپرسید، به دنبال کلماتی در دیکشنری بگردید، تعدادی مقاله در مورد فناوری که استفاده می‌کنید بخوانید یا کتابی را مطالعه کنید تا به شما در درک بهتری نسبت به کار بدهد.
درنهایت، مسئولیت‌پذیری تنها با اراده کردن و مصمم شدن به دست می‌آید. زمانی که مشکلی بر سر راهتان قرار بگیرد، تصمیم بگیرید آن را حل کنید تا اینکه آن را کنار بگذارید. وقتی در بیرون از تیمتان مشکلی پیش بیاید، تصمیم بگیرید به حل آن کمک کنید تا اینکه بخشی از آن مشکل شوید. همچنین نوع ویژه‌ای از مسئولیت‌پذیری نیز وجود دارد، زمانی که شخص دیگری مشکلی دارد، تمایل داشته باشید که به وی کمک کنید یا تمایل داشته باشید به او کمک کنید که مشکلش را خودش حل کند. مسئولیت‌پذیری تنها بدان معنا نیست که باید همه‌ی کارها را خودتان انجام دهید. حتی می‌تواند به این معنا باشد که مایل باشید به دیگران کمک کنید کارشان را انجام دهند.
انجام دادن کارهای نام‌برده سخت نیست، تا مادامی که آن را مطابق با گام‌های ساده انجام دهید. شما مجبور نیستید از کل جهان آگاهی داشته باشید، تمام جوانب مشکل را ادراک کنید و یا اینکه تمایل داشته باشید تمامی اجزای نرم‌افزار را تغییر دهید. با یک جزء کوچک آغاز کنید سپس به مراحل بعد بروید، در این حالت یک برنامه‌نویس خوب خواهید شد.

رویـــداد

کنفرانس PMO 2019

تاریخ: سیزدهم ژوئن ۲۰۱۹

مکان: لندن، انگلستان

کنفرانس PMO رویدادی اســت که توسط جامعه‌ی مدیران پروژه ســازمان‌دهی و برای شــاغلین PMO (مدیریت پروژه) در سراسر جهان طراحی‌شده که بر اداره‌ی سبد محصولات، برنامه‌ریزی و پروژه تمرکز دارد. این کنفرانس در مورد کنکاش عمیق‌تر در این حرفه اســت و شــامل بخش‌هایی برای ســخنرانان متخصص جهانی، نمایشگاه‌های PMO اختصاصی و زمــان زیــادی بــرای ارتباطــات جدیــد PMO می‌باشد.

یک مزیت کلیدی این کنفرانس آن است که تمامی پروژه‌هــای مدیریــت پروژه‌ی رایج و خدمات آن را به نمایــش می‌گــذارد. بنابراین اگــر می‌خواهید در این کنفرانس شــرکت کنید، برای یادگیری موارد زیادی در مــورد متدهــای مدیریــت پــروژه، گرایشــات و تمرینات آماده شوید.

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 امیدپلاک ۶، واحد ۱ شماره تماس: ۷۰-۸۸۶۵۹۴۶ ۰۲۱ پست الکترونیک: info@dadehpardaz.com

انعطاف‌پذیری بیشتر مدیریت Undo Tablespace در MySQL 8.0.2

ویژگی‌هایی از DMR MySQL 8.0.2 را معرفی می‌کنیم که مدیریت undo tablespace را در InnoDB آسان‌تر می‌کنند. بخش اصلی ارتقایی که صورت گرفته این است که می‌توانید undo tablespace-ها را در هر موقع بسازید و مستقر کنید. می‌توانید فایل config را تغییر بدهید و قبل از هر راه‌اندازی (startup)، تعیین کنید که آیا به بازیابی (recovery) نیاز است یا خیر.

innodb_undo_tablespaces

undo tablespace-ها دربردارندهٔ Rollback Segment-ها هستند که آن‌ها نیز undo log-ها را دربردارند. undo log-ها برای 'roll back' یا «عقب‌گرد» تراکنش‌ها و ایجاد نسخه‌های اولیهٔ داده‌هایی استفاده می‌شود که Multi Version Concurrency Control (کنترل همزمانی چندنسخه‌ای) از آن‌ها برای ارائهٔ یک نمای منسجم از پایگاه‌داده درطیِ یک تراکنش استفاده می‌کند.

قبلاً تعداد undo tablespace-هایی که InnoDB به کار می‌گرفت هنگام آماده‌سازی پایگاه‌داده (initialization) تعیین می‌شد. اکنون می‌توان به آن، هرموقع، مقداری بین ۰ تا ۱۲۷ داد؛ چه هنگام راه‌اندازی در فایل config یا در خط فرمان (command line)، یا به‌طور آنلاین با تعیین 'SET GLOBAL INNODB_UNDO_TABLESPACES=n'.

وقتی که undo tablespace-های صفر را انتخاب می‌کنید، system tablespace تمام rollback segment-ها را دنبال می‌کند. این شیوهٔ قدیمی ذخیرهٔ system tablespace قبل از اضافه شدن undo tablespace-های مجزا به نسخهٔ ۵,۶ است. به‌این‌ترتیب تلاش می‌کنیم که از این نحوهٔ استفاده از system tablespace فاصله بگیریم تا مقدار پیش‌فرض برابر ۲ قرار نگیرد. در آیندهٔ نزدیک مقدار کمینه برابر ۲ خواهد بود که به‌این معناست که system tablespace برای هیچ rollback segment-ئی استفاده نخواهد شد. پس لطفاً در فایل‌های config خود قرار ندهید: innodb_undo_tablespaces=۰.

innodb_undo_log_truncate

مقدار کمینهٔ ۲ را برای undo tablespace-ها انتخاب کردیم زیرا برای اینکه حداقل یکی از آن‌ها کوتاه‌سازی شود (truncate)، حداقل به ۲ نیاز داریم. Undo truncation به InnoDB این امکان را می‌دهد که اندازهٔ undo tablespace را بعد از تراکنش‌هایی که بزرگیِ غیرعادی دارند، کوچک‌سازی کند. پیش‌ازاین تنظیمات innodb_undo_log_truncate به‌طور پیش‌فرض OFF قرار داده می‌شد. در نسخهٔ ۸,۰,۲ به‌طور پیش‌فرض ON قرار داده می‌شود.

innodb_rollback_segments

می‌توان، هرموقع در راه‌اندازی چه در فایل config و چه از خط فرمان (command line) و چه به‌طور آنلاین با تعیین 'SET GLOBAL INNODB_ROLLBACK_SEGMENTS=n'، به آن مقداری بین ۱ تا ۱۲۸ داد.

این تنظیمات قبلاً برابر تعداد rollback segment-هایی بود که کل سِرور می‌توانست پشتیبانی کند. اکنون برابر تعداد rollback segment-ها در هر undo tablespace است و ازاین‌رو تراکنش‌های همزمان از rollback segment-های بیشتری استفاده می‌کنند. مقدار پیش‌فرض هنوز ۱۲۸ است.

innodb_undo_logs

این تنظیمات در ۵٫۶، به‌عنوان یک جایگزین یا با نام مستعار innodb_rollback_segments معرفی شد. از لحاظ لغتی گیج‌کننده بود زیرا در InnoDB، Undo Log‌ها در Rollback Segment‌ها ذخیره می‌شدند که file segment‌های یک Undo Tablespace هستند. در v8.0.2 استفاده از این تنظیمات را کنار می‌گذاریم و به‌جای آن لازم است که از Innodb_rollback_segments استفاده کنیم. آخرین نسخهٔ عرضه‌شدهٔ ۵٫۷٫۱۹ هشدارهایی مبنی بر منسوخ شدن آن دارد.

Undo Tablespace Name and Location

undo tablespace‌ها در مسیری (دایرکتوری‌ای) که با تنظیم innodb_undo_directory تعیین می‌شود قرار دارد. اگر از این تنظیمات استفاده نشود، در محل 'datadir' ایجاد می‌شوند. قبلاً اسم‌هایی مثل 'undo۰۰۱' یا 'undo۰۰۲' و غیره داشتند. در ۷۸٫۰٫۲ DMR اسم‌هایی مثل undo_001 و undo_002 و غیره دارند. علت تغییر اسم این است که این undo tablespace‌های جدیدتر یک صفحهٔ سرایند (هدر) جدید دارند که محل‌هایی را که هر rollback segment دارد نگاشت می‌کند. در نسخهٔ ۵٫۶ وقتی که undo tablespace‌های مجزا ارائه شدند، تعداد صفحات سرآیندِ (هدرِ) rollback segment آنِ‌ها در system tablespace پیگیری شدند که تعداد rollback segment‌ها را برای کل نمونه (instance) به ۱۲۸ محدود کرد. به دلیل اینکه هر undo tablespace می‌تواند rollback segment‌های خودش را با این صفحهٔ جدید پیگیری کند، واقعاً انواع جدیدی از undo tablespace‌ها محسوب می‌شوند که باید مقررات نام‌دهی متفاوتی داشته باشند. به این دلیل است که اکنون innodb_rollback_segments ، به‌جای تعداد کل نمونهٔ MySQL، تعداد rollback segment‌ها را برای هر undo tablespace تعریف می‌کند.

Automatic Upgrade

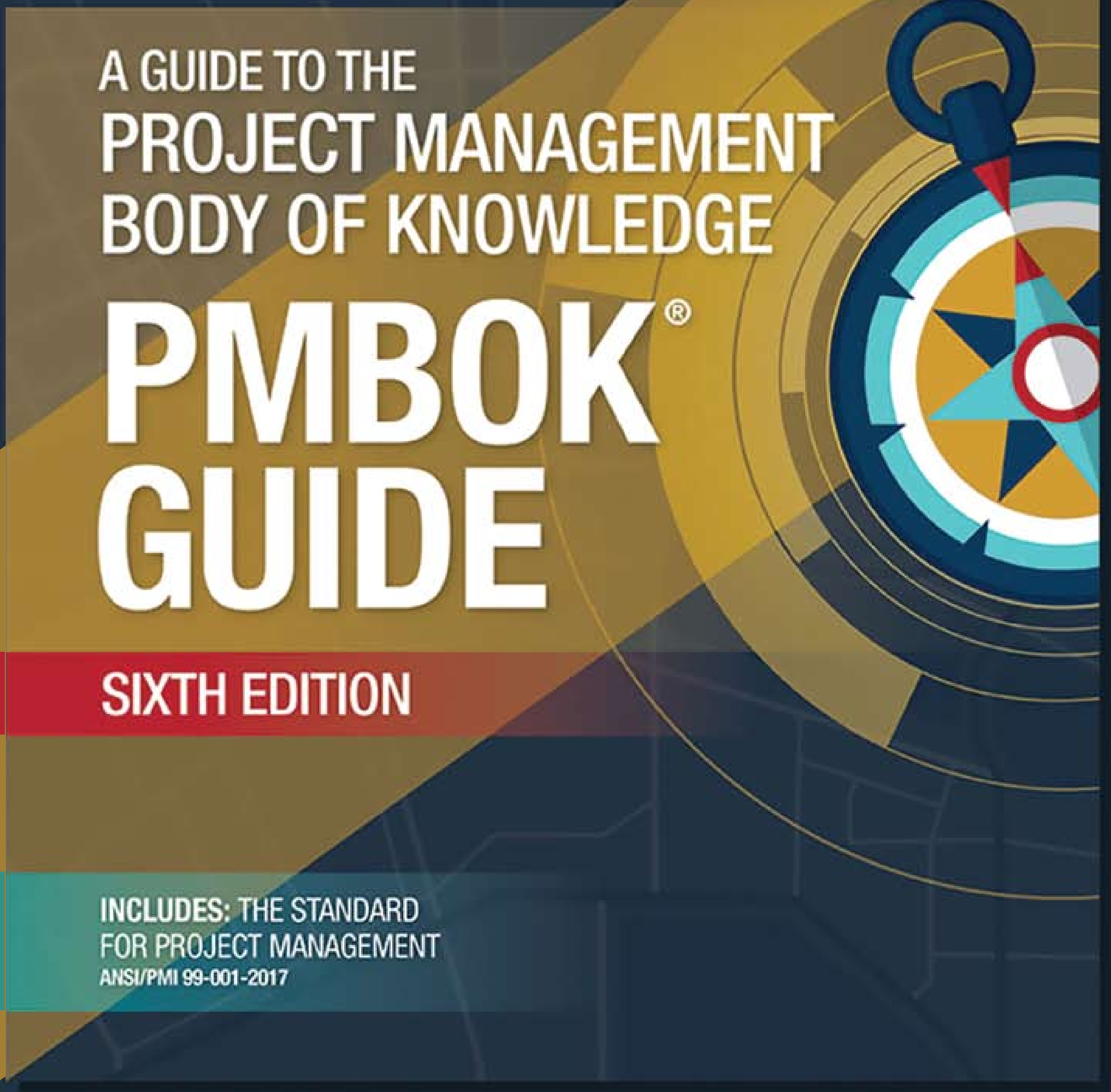
قبل از این تغییر، system tablespace تمام rollback segment‌ها را پیگیری می‌کرد، چه در system tablespace و چه در undo tablespace‌ها. اگر MySQL ۸٫۰٫۲ را بر پایگاه‌دادهٔ موجود که از system tablespace برای پیگیری rollback segment‌ها استفاده می‌کند راه‌اندازی کرده باشید، حداقل دو undo tablespace جدید به‌طور خودکار تولید می‌شوند. این کار رایج خواهد شد زیرا مقدار پیش‌فرض قبلی برای innodb_undo_tablespaces برابر ۰ بود. پایگاه‌داده‌های MySQL ۵٫۷ متحمل یک روند به‌روزرسانی می‌شوند که DDِ جدید را از فایل‌های قدیمی FRM می‌سازد. بخشی از این فرایند ایجاد حداقل دو undo tablespace جدیِد InnoDB است. همچنان می‌تواند از rollback segment‌های موجود و undo tablespace‌های تعریف‌شده در system tablespace استفاده کند، اگر هنگام راه‌اندازی در خودشان undo log‌ها را داشته باشند. اما این کار rollback segment‌های قدیمی را برای هیچ‌کدام از تراکنش‌های جدید تعیین نخواهد کرد. بنابراین وقتی undo recovery (دوباره کاری بازیابی) تمام شد، و دیگر به این undo log‌ها نیاز نبود، undo tablespace‌های قدیمی حذف می‌شوند.

راهنمای جدید

PMBOK

ویرایش ۶

امتحان PMP مؤسسهٔ مدیریت پروژه این بار نیز در نتیجهٔ ویرایش جدید کتاب «راهنمای پیکرهٔ دانش مدیریت پروژه (پم‌باک)» تغییر می کند. در این مقالهٔ برخی از تغییرات نسخهٔ ششم راهنمای پم‌باک را بررسی می کنیم.



چه‌چیز در ویرایش جدید راهنمای پم‌باک جدید است؟

اول اینکه تغییرات فرایندها و نحوهٔ سازمان‌دهی آن‌ها تغییر قابل توجهی نکرده است. چند فرایند برای سازگاری بیشتر مجدداً نام‌گذاری و چند فرایند جدید اضافه شده‌اند. اما ساختار اصلی حوزه‌های دانش (Knowledge Areas یا KA) و گروه‌های فرایند (Process Groups یا PG) تغییر نکرده‌اند. برای اطمینان جزئیات خروجی‌ها و ورودی‌ها بهبود یافته‌اند اما خواندن آن نیاز به دقت و حوصلهٔ زیادی دارد؛ پس دربارهٔ تغییرات جالب آن را صحبت می کنیم.

سه بخش جدید در ابتدای هر KA اضافه شده‌اند: گرایش‌ها و روش‌های درحال‌ظهور، هم‌سازسازی ملاحظات و ملاحظات محیط‌های تطبیقی-چابک. گرایش‌ها و روش‌های درحال‌ظهور شامل ابزاری مثل زمان‌بندی‌های کسب‌شده در حوزهٔ دانش مدیریت پروژه (Cost Management KA) است که خودش یک روش جدید نیست اما در ویرایش پنجم راهنمای پم‌باک ذکر نشده بود. مثال دیگر، کاربرد چندین پیمانکاران و فرم استاندارد قرارداد است که مناسبِ مدیریت کارپردازی در ابرپروژه‌های بین‌المللی بزرگ هستند. ایدهٔ تیم‌های خودسازمان‌ده به‌عنوان یک روش درحال‌ظهور در مدیریت منابع (که قبلا HR نامیده می‌شد) تعریف شده است. چشم‌انداز وسیع تری از تعریف ذی‌نفع ارائه شده و به مشارکت واداشتن تمام اعضای تیم با ذی‌نفعان، به‌عنوانِ یک گرایش در حوزهٔ دانشِ مدیریت ذی‌نفعان مطرح شده‌است. هم‌سازسازی ملاحظات به موضوعاتی شفافیت می‌بخشد که در ویرایش‌های قبلی راهنمای پم‌باک به آن اشاره‌ای شده‌بود. سؤال‌های مطرح‌شده به فهمیدن نحوهٔ کارکردِ فرایندهای مربوط‌به ذی‌نفعان و محدودیت‌های منحصر به یک سازمان کمک می کند. به‌طورمثال دربارهٔ مدیریت ارتباطات، اعضای تیم پروژه به چند زبان صحبت می کنند؟ و آیا اعضای تیم از لحاظ جغرافیایی پراکنده‌اند؟ چقدر دربارهٔ نیازمندی‌ها اطمینان وجود دارد و چقدر بر فرایندهای مدیریت محدوده تأثیر می گذارند؟ ابعاد و پیچیدگی پروژه چه تأثیری بر فرایندهای مدیریت ریسک دارد؟ ایدهٔ هم‌سازسازی در راهنمای پم‌باک تازه نیست؛ اما افزودن این بخش‌ها با سؤالاتی که مختص هر حوزهٔ دانش است هم‌سازسازی را بیشتر به‌چشم می‌آورد و به این مستند حس انعطاف‌پذیری بیشتری را می‌دهد؛ تا راهنما کمتر مثل یک روشگان (متدولوژِی) تجویزی، که اغلب اوقات اشتباه فَهمیده می‌شود، به‌نظر برسد.

ملاحظات محیط‌های چابک/تعاملی سومین بخش جدید در فصل KA است. در این بخش‌های مختصر و سطح‌بالا، مفهومِ چرخهٔ حیاتِ پروژه برای درک فرایندها به کار برده می. شود. در این بخش‌ها، به تأثیر عدم اطمینان بر فرایندها و مزایای اموری چون بخش‌بخش کردنِ کار و بازخوردگیری مدام اشاره شده است. با این بخشِ مختصر، خواننده‌ای که نحوهٔ به‌کارگیری فرایندها و ابزار را ارزیابی می کند می تواند نوع پروژه‌ها را به‌خاطر بسپارد. البته جزئیات مربوط‌به پروژه‌های محیط‌های تطبیقی در راهنمای روش چابک (Agile Practice Guide)، که در همکاری اتحادیهٔ چابک (Agile Alliance) تهیه شده و اکنون با ویرایش ششم PMBOK عرضه می‌شود، مطرح می‌شوند. علاوه بر این بخش‌های جدید، بر چند مفهوم قدیمی بیشتر تأکید شده و چند مصنوعات و مفهوم جدید مطرح شده‌اند. مثلاً در نسخه‌های قبلی، بر نقش مدیر پروژه بیشتر تأکید می‌شد؛ زیرا زیربنای آن در سال‌های قبل و در Talent Triangle PMI بنا گذشته شده بود. به مهارت‌های رهبری و مهارت‌های نرم مانند مذاکرهٔ سیاست‌ها بیشتر پرداخته شده تا تأکیدی باشد بر هم‌راستایی پروژه‌ها با کسب‌کار و تحویل ارزش تجاری آن. درکل راهنمای PMBOK در هر نسخه منسجم‌تر می‌شود. افزون‌بر‌آن، نمودارها تمیزتر و فهمیدنشان آسان‌تر شده‌است. خواندن صفحات خاکستری چندان دلچسب نیست اما به‌دلیل جلوگیری از تکثیر غیرمجاز و حفظ مالکیت معنوی ناگزیر است. بی شک، پم‌باک به رشد خود ادامه می‌دهد، هرچند ۷۵۶ صفحهٔ آن به‌نظر وحشت‌آور باشد! در مقاله‌های آینده، راهکاری برای درک بهتر ویرایش جدید PMBOK ارائه می کنیم. یادآوری می‌شود که کتاب «راهنمای پیکرهٔ دانش مدیریت پروژه، ویرایش پنجم»، به ترجمهٔ آقای نادر خرمی راد، و همین‌طور آقایان مهدی ابراهیمی، علی بیاتی و فرشید ادهمی، انتشارات آریانا، به زبان فارسی موجود است.

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 مشغولِ وارد کردن جزئیاتِ 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؛ ازین‌رو بعداز 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 به‌بار 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، عنوانِ «جلوتر از زمان‌بندی» را می‌گیرد؛ چراَ 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 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.

بایدَ خودتان آستانهٔ این مقادیر را، که درصورت هر تغییری برحسب آن تغییر می‌کنید، از همان ابتدای کار تعیینِ 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.

سوز هینتون

توسعه‌دهندهٔ مایکروسافت، هکر سخت‌افزار / کاربر مک، وب، ویندوز

■ خودتان را معرفی کنید، چه می‌کنید؟

من سوز هینتون (Suz Hinton) هستم. در ملبورنِ استرالیا به دنیا آمدم و بزرگ شدم؛ در آنجا کم‌کم فهمیدم که خورهٔ رایانه و فناوری‌ام و ۷ سال پیش به ایالات متحده آمدم و درحال‌حاضر در نیویورک کار می‌کنم. الآن در مایکروسافت توسعه‌دهندهٔ نرم‌افزار هستم. با دو شرکت و جمعیت برای ساخت بخش خدمات ابری Azure و دیگر ابزار مایکروسافت کار می‌کنم. این کار برای من ترکیبی است از برنامه‌نویسی، سخنرانی عمومی، مستندنویسی و ساختن پیش‌نمونه‌های (prototype) جالب؛ در این کار سرعت آموختن و آموزش دادنم زیاد است و هیچ‌وقت دستم از پروژه‌هایی که دوست دارم انجام بدهم خالی نمی‌شود. غیر از وظایفم به ساختن چند کتابخانهٔ سخت‌افزاری JavaScript مشغولم که خیلی دوستشان دارم. برای خوشحال کردن دیگران وسایل الکترونیکیِ درست می‌کنم. خواندن برگه‌های فنی را واقعاً دوست دارم؛ دوست دارم از موضوعات مربوط به دسترس‌پذیری و سمت کاربر (fronend) سر در بیاورم. اکثراً آخر هفته‌ها در توئیتر یک برنامهٔ زنده کدنویسی دارم که مخاطبان بسیاری پیدا کرده است.

■ از چه سخت‌افزاری استفاده می‌کنید؟

از خانه کار می‌کنم، از جایی که مطالعه می‌کنم. یک میز قدیمی IKEA دارم که ۵ سال پیش خریدمش. بر یک صندلیِ نصف‌قیمت Areon Chair می‌نشینم که واقعاً از پشتم در ساعات طولانی کار حفاظت می‌کند. تمام روز با صفحه‌نمایش ۲۷ اینچی Acer K272 کار می‌کنم. از یک موشوارهٔ (موس) بی‌سیم Perixx PERMICE-712 و یک صفحه‌کلید مکانیکی Leopold M۶۶۰ استفاده می‌کنم که خیلی دوستش دارم؛ کلیدهایش قهوه‌ای و آلبالویی بود اما اخیراً روکش صفحه‌کلیدم را ارغوانی کرده‌ام. به این صفحه‌کلید افتخار می‌کنم و از نوشتن نرم‌افزار در آن لذت می‌برم.

از دو رایانهٔ مختلف استفاده می‌کنم. رایانهٔ کارم Microsoft Surface Book و رایانهٔ شخصی‌ام Apple MacBook Pro است. برای کار در شبکه‌های مجازی از یک میکروفون Blue Yeti و وب‌کم Logitech C920 Pro و نرم‌افزار OBS Studio streaming استفاده می‌کنم. یک چاپگر سه‌بعدیِ Printrbot Plus Metal دارم. از آن به‌طور بی‌سیم با OctoPrint که بر Raspberry Pi اجرا می‌شود کار می‌کنم. هویه‌ام Hakko FX-888D است. قهوه‌خردکنم Bodum BISTRO است و قهوه‌هایم را با Aeropress درست می‌کنم. شیر سویا در قهوه‌ام می‌ریزم زیرا طعم شیر سویا را در قهوه خیلی دوست دارم. تنها خوردنی‌ای که در آن افراط می‌کنم قهوه است و غیر از قهوه خیلی غذا دوست ندارم.

■ از چه نرم‌افزاری استفاده می‌کنید؟

برای ویرایش کد از vim استفاده می‌کنم. در MacBook از iTerm استفاده می‌کنم. در Surface Book با Bash on Ubuntu که ConEmu را اجرا می‌کند زندگی می‌کنم. مرورگرم بیشتر Chrome و Edge است. تمام زندگی‌ام در OneNote است. در آن کل لیست وظایفم، پست‌های وبلاگم، پروژه‌های درازمدت، و اهداف شخصی و هر چیزی را که فکرش را بکنید ذخیره می‌کنم. بدون تقویم Outlookام نمی‌توانم زندگی کنم، چون با آن می‌توانم بدانم که در هر زمانی باید کجا باشم. با برنامهٔ کاربردی Spotify روزانه به موسیقی گوش می‌دهم و هدفون موردعلاقه‌ام Sennheiser PX200IIs است. زبان‌های موردعلاقهٔ برنامه‌نویسی‌ام JavaScript و Python هستند.

■ امکانات یا سیستم رؤیایی شما چیست؟

دوست دارم در اتاقی کار کنم که پنجره‌های بزرگ داشته باشد و بیرون از آن پر از درخت باشد. موقع فکر کردن، نگاه کردن به پرنده‌ها و حیوانات را دوست دارم. حداقل یک میز سفید (واقعی) می‌خواهم. اگر یک اتاق جدا برای لحیم‌کاری و ساختن چیزها داشته باشم بد نیست. اخیراً برنامه‌نویسی و لحیم‌کاری را روی یک میز می‌کنم، برای همین میزم خیلی جادار و کاربردی نیست. دوست داشتم چند تا صفحه‌نمایش داشتم، مثل هکرهای کلیشه‌ای داستان‌های هالیوودی! الآن برای صفحه‌نمایش‌هایم جا ندارم. بیشتر فضایم را با گل‌وگیاه یا کارهای هنری دوستانم پر کرده‌ام. از همه بیشتر به یک محیط آرام‌تر نیاز دارم. به شلوغی نیویورک عادت کرده‌ام اما دلم برای سکوت و آرامش هم تنگ شده!

چگونه با خلأهای آموزشی مربوط به مدیریت خانم‌ها در محل کار مواجه شویم

تنها ۵ درصد از ۵۰۰ کمپانی خوش‌شانس توسط بانوان رهبری و هدایت می‌شود. اگرچه تلاش‌ها برای بهبود بخشیدن به استفاده از خانم‌ها در نقش‌های رهبری و مدیریتی از دوران ابتدایی شکل می‌گیرد و در هر سطحی از کار ادامه می‌یابد، اما تاکنون شرایط، تغییر چشمگیری نیافته است. بر اساس تحقیق جدید از کمپانی یادگیری آنلاین D2L، این عدم پیشرفت می‌تواند به دلیل تفاوت‌های چشمگیر در روش‌هایی آموزش مردان وزنان در شرکت‌ها می‌باشد.

پس از تحقیق و مطالعه روی ۱۰۰۰ کارگر شرکت‌ها در صنایع مختلف در فوریه ۲۰۱۸، تحقیق D2L آشکار کرد که در کل، زنان نسبت به فرصت‌های آموزشی موجود در سازمان‌هایشان تقریباً در تمامی انواع یادگیری، آگاهی کم‌تری دارند و اطلاعات کمتری به آنان انتقال داده می‌شود.

بر اساس این تحقیق، به‌طور میانگین، ۵۶درصد از مردان می‌گویند که شرکت آنان آموزش مهارت‌ها را ارائه می‌کند، حال آنکه درصد زنانی که از این آموزش‌ها اطلاع دارند ۴۲ درصد است. این اختلاف در اطلاع و آگاهی به‌ویژه در صنعت بانکداری صادق است، هرچند این بخش فرصت‌های آموزشی بیشتری دارد. این تحقیق دریافت که ۷۴ درصد از مردان و ۵۸ درصد از زنان شاغل در بانک‌ها اطلاع دارند که مجموعه‌ی آنان به آموزش مهارت‌ها می‌پردازد. در مقایسه، صنایع خرده‌فروشی، امور مسافرتی و مشاور املاک، اختلاف کمتری بین زنان و مردان در این زمینه‌دارند؛ ۵۶ درصد از مردان و ۴۶ درصد از زنان در بخش خرده‌فروشی، امور مسافرتی و مشاور املاک مطلع هستند که مجموعه‌شان آموزش‌هایی برای افزایش مهارت ارائه می‌کند.

بر اساس این تحقیق، در مورد به اشتراک‌گذاری اطلاعات مربوط به مهارت‌ها، دانش و آموزش نیز اختلاف مشابهی وجود دارد. مردان بیشتر معتقدند سازمانی که برای آن کار می‌کنند، تخصص‌های لازم را به‌خوبی به آنان آموزش می‌دهد (۷۳ درصد از مردان و ۵۵ درصد از زنان). اما این اختلاف زمانی که صحبت از تفاوت نسل‌ها می‌شود، دیگر مدنظر قرار نمی‌گیرد. کارکنان قدیمی، کارگرانی که ۵۴ سال به بالا هستند، در حقیقت حداقل تفاوت بین اعتقاد مردان (۶۳ درصد) وزنان (۸۳ درصد) در به اشتراک گذاشته شدن اطلاعات را دارند. اما نیروهای جوان (بین ۱۸ تا ۳۷ سال) اختلاف بسیار گسترده‌ای بین مردان (۸۳ درصد) وزنان (۵۵ درصد) را در این زمینه نشان می‌دهند.

مردان از برنامه‌های یادگیری و توسعه‌ی خود نسبت به زنان، رضایت بیشتری دارند؛ چراکه ۷۵ درصد از مردان از این جریان راضی هستند درحالی که ۵۵ درصد از زنان از آموزش‌ها رضایت دارند. احتمالاً بدان دلیل است که زنان می‌گویند دسترسی کمتری به برنامه‌ی یادگیری و توسعه در محل کاردارند. ۱۶ درصد از زنان گزارش کرده‌اند که دسترسی ندارند، حال آنکه تنها ۴ درصد از مردان چنین موردی را گزارش کرده‌اند. به نظر می‌رسد که میزان دسترسی، در برنامه‌های یادگیری آنلاین بهتر می‌شود، اما همچنان اختلافی باقی می‌ماند: ۶۴درصد از مردان می‌گویند که به آموزش‌های آنلاین دسترسی دارند درحالی که این میزان در زنان ۴۸ درصد گزارش‌شده است.

در مورد انواع خاص آموزش مهارت‌ها، تجربیات نیز بین زنان و مردان فرق می‌کند. در مورد مهارت‌های فنی، ۶۸ درصد از مردان و ۴۷ درصد از زنان می‌گویند که به این نوع آموزش دسترسی دارند، درحالی که ۲۳ درصد از مردان و ۳۷ درصد از زنان دسترسی نداشته‌اند اما آنان دوست داشتند که از این مزیت برخوردار شوند.

این تحقیق همچنین نشان می‌دهد که در دسترسی به کلاس‌های آموزش قابلیت تطابق یا انعطاف‌پذیری (۵۳ درصد از مردان و ۴۵ درصد از زنان)، خلاقیت،

منطق و آموزش حل مسئله (۵۳ درصد از مردان و ۴۰ درصد از زنان) و نهایتاً آموزش هوش روابط بین افراد یا هوش عاطفی (۵۴ درصد از مردان و ۴۰ درصد از زنان) نیز تفاوت‌هایی بین زنان و مردان وجود دارد. در کلاس‌های آموزش آنالیز، قضاوت و تصمیم گیری (۵۴ درصد از مردان و ۴۳ درصد از زنان)، مدیریت منابع (۵۳ درصد از مردان و ۴۰ درصد از زنان)، و آموزش ارتباطات کتبی و شفاهی (۵۶ درصد از مردان و ۴۰ درصد از زنان) نیز صدق می کند.

نقش شرکت‌ها در کم کردن اختلافات

مسئول این اختلافات آگاهی و دسترسی در زنان و مردان چیست؟ چریل آینوا، مدیر ارشد اجرایی D2L از مسئولیتی که در چگونگی ارتباط و تعامل مجموعه‌ها با مردان نسبت به چگونگی ارتباط آن‌ها با زنان، بیشتر صحبت می کند. این نکات عبارت‌اند از تفاوت در نحوه‌ی تربیت و اجتماعی شدن زنان و مردان از تولد که می–تواند در بازداشتن زنان از مزیت آموزش، یادگیری و درنتیجه فرصت‌هایی برای پیشرفت، تجلی یابد.

مسئله‌ی مهم برای تصمیم گیرندگان امروزی بخش فناوری اطلاعات تنها این نیست که به فضای ابری مهاجرت کنند، بلکه مهاجرت دادن چه مواردی و در چه زمانی، مسئله‌ای مهم می‌باشد. پاسخ به این سؤالات را در اینجا بیابید.

آینوا می گوید: این کار لزوماً از قصد و عمدی نیست. وی این جریان را دریکی از نقش‌های قبلی خود تجربه کرده است.

وی می گوید: تفاوت‌هایی در نحوه‌ی تعامل مردان وزنان در محل کار وجود دارد. من موقعیتی داشتم که در آن دو گزارش مستقیم دریافت می کردم، یک گزارش از یک مرد و دیگری از یک زن. ما این فرصت را داشتیم تا یکی از آنان را به نقش و سِمت رهبری (مدیریت) منسوب کنیم و صادقانه بگویم آن زن صلاحیت بیشتری داشت. بنابراین ما با تیم اجرایی، نشستی برگزار کردیم و هر دو را مدنظر قراردادیم. مرد بیشتر بر این تمرکز داشت که گام بعدی چیست؟ من چه مهارت‌هایی را باید تغییر دهم؟ بهترین فرصت برای پیشرفت و بهبودی کجاست؟ درحالی که زن دائماً به من می گفت: من در این نقش آموزش‌های بسیاری دیده‌ام، من اطلاعات کاملی دارم و از این بابت خوشحالم، اما اگر فرصتی ایجاد شود، به پیشرفت نیز فکر می کنم.

تصمیم به پیشرفت ساده به نظر می رسید: مرد برای رهبری و مدیریت، پیشرفت می کند؛ زیرا زن از نقشی که داشت خوشحال و راضی بود و شرکت با حفظ هردوی آن‌ها سود می‌برد.

وی افزود: چند هفته بعد، آن زن به دفتر من آمد و استعفا داد. من بسیار متعجب شدم، تا اینکه بیشتر در این مورد با دیگران صحبت کردم. آنچه اتفاق افتاد چیزی است که ما روزانه شاهد هستیم: مردان معتقدند اگر ۶۰ الی ۷۰ درصد از مهارت‌ها و تجربیات لازم را داشته باشند، شایسته‌ی آن کار خواهند بود. زنان احساس صلاحیت نمی کنند مگر اینکه ۱۲۰ درصد آمادگی را در خود ببینند. آن زن بااینکه می‌دانست شایستگی بیشتری نسبت به مرد دارد، از بودن در آن نقش، ناراضی و بی‌میل شده بود، بنابراین زمانی که فرصتی پیش آمد، استعفا داد. ما نیز او را از دست دادیم؛ زیرا از چندوچون اتفاقات آگاهی نداشتیم.

لازم است بدانید که در این سفر کجا هستید و تا چه حد موفق بوده‌اید. بیاموزید که چگونه پیشرفت را دنبال می کنید و اهداف IT و زمان‌بندی واضحی دارید.

شاید بسیاری ترجیح می‌دادند آن زن را به دلیل آنکه از پس کارش برنیامده سرزنش کنند. اما آینوا می گوید که مقصر اصلی عدم ادراک مدیریت او بوده است. او این شیوه را متد مدیریت چرخ‌دنده‌ای (مکانیکی) می‌نامد و از آن به‌عنوان مانعی اصلی برای پیشرفت زنان در نقش‌های رهبری و مدیریتی یاد می کند.

وی می‌افزاید: رهبران زمانی که تنها به مردم چرخ‌دنده‌ای توجه دارند به‌اشتباه احساس امنیت می کنند، افرادی که در مورد آنچه دوست دارند، می‌خواهند بروند و جایگاهی که باید به دست آورند

خیلی صحبت می‌کنند. مردان اغلب به این روش، تربیت اجتماعی می‌شوند، اما زنان نه. زنان در فهرست اولویت، خود را آخر قرار می‌دهند و اولویت را به هر چیز یا هر کس دیگر می‌دهند: خانواده، همسر، کودکانشان و در محل کار هم تیم و همکارانشان را در اولویت بالاتری نسبت به خود قرار می‌دهند. آنان نمی‌خواهند آن فرصت پیشرفت و ارتقا را به دست آورند! بنابراین زمانی که فرصتی ایجاد می‌شود، مدیریت مجموعه در ابتدا به سراغ افرادی می‌رود که بلندترین صدا را دارند و بیشتر در مورد افکار خود حرف می‌زنند؛ در نتیجه افراد بسیار خوبی را که علاقه‌ای به داوطلب شدن و خود نشان دادن ندارند، نادیده می‌گیرند.

آینوا اظهار داشت: از بین بردن شکاف و فاصله و نیز آسان کردن دسترسی به یادگیری و آموزش، با آگاهی از این عوامل و استفاده از آن‌ها به نفع خودتان آغاز می‌شود. درس‌هایی وجود دارد که باید از تأکید کمپانی‌ها بر استفاده از شخصیت‌های مختلف، آموخت.

وی در ادامه گفت: یکی دیگر از دلایل این تفاوت آن است که به زنان یاد داده‌اند بیشتر درون‌گرا باشند تا برون‌گرا. آنان انرژی خود را به روش‌های مختلفی به دست می‌آورند، بنابراین اگر این حقیقت را بدانید، می‌توانید اطلاعات و روش خود را بر اساس این حقیقت پیش ببرید تا به بیشترین موفقیت برسید.

یک رویکرد شخصی بهترین گزینه است

مدیران به‌جای اینکه تعداد زیادی ایمیل در مورد یادگیری و آموزش برای تمام شرکت ارسال کنند، باید تلاش کنند تا مستقیماً از گزارشات خود کمک بگیرند و تک‌تک افراد را آگاه سازند که این فرصت‌های یادگیری و آموزش تا چه حد و به چه دلیل می‌تواند مفید باشد. آینوا متذکر شد: وقتی که من معلم بودم، ما تعدادی زن فنی ارشد داشتیم که بسیار عالی بودند اما در حالی که مردان در رویدادها و اجتماعات صحبت می‌کردند، می‌خواستیم مطمئن شویم که زنان نیز این‌گونه هستند یا نه. ما این فرصت را داشتیم که در مجموعه کنفرانس‌های زنان فنی Grace Hopper Celebration (GHC) شرکت کنیم و سعی داشتیم تجسم کنیم که چگونه می‌توان به بهترین شکل ممکن زنان را بر آن داشت تا پیشنهادات خود را ارائه کنند. من به تمام مدیران ایمیل فرستادم و گفتم: آیا ممکن است نیروی ارشد زن متخصص فنی خود را تشویق کنید تا یک پیشنهاد ارائه دهد؟

وی افزود: ما تعدادی از پیشنهادات زنان را پذیرفتیم. چارچوب‌بندی یادگیری و به وجود آوردن فرصت‌هایی برای تقدیر از اجتماعی کردن زنان می‌تواند کمک شایانی به تقویت این حس در زنان کند.

همچنین ادامه داد: فراموش نکنید که زنان به گونه‌ای تربیت شده‌اند که می‌پندارند تلاش برای دستیابی به این موفقیت‌ها به‌نوعی خودخواهی تلقی می‌شود و موجب پایین کشیدن سایرین خواهد شد. بنابراین موفقیت بزرگ‌تر را در چارچوب‌بندی این موارد می‌دانم زیرا شما در قبال رشد و توسعه‌ی این مهارت‌ها مسئول هستید و باید این دانش را به کل تیم نسبت دهید، به‌جای اینکه آن را یک موفقیت انفرادی بدانید. همچنین زمانی که من این کار را انجام دادم تعداد خانم‌هایی که در این دوره‌های آموزشی شرکت می‌کردند، افزایش یافت. اگرچه هدف نهایی شکستن آن موانع اجتماعی شدن و تشویق تمامی افراد در بهره‌مندی از این فرصت‌ها برای اصلاح و رضایت شخصی می‌باشد، اما ما هنوز به آن نقطه نرسیده‌ایم. هرچند تاکنون برقراری تعادل آن سیستم‌ها برای سوق دادن زنان بیشتری به‌سوی رهبری، گام بسیار مفید و مثبتی می‌باشد. با ورود به عصر جدیدی از کار، آموزش پس از دانشگاه برای زنان ضروری‌تر می‌شود تا در مشاغل خود رشد بیشتری داشته باشند. به‌عبارت‌دیگر آموزش حین کار امری ضروری است و کمپانی‌هایی که به تنوع (به‌ویژه در رهبری) اهمیت می‌دهند باید اطمینان یابند که مردان و زنان فرصت‌های برابری در بهره‌مندی از این دوره‌های آموزشی دارند. برای اطمینان از اینکه یادگیری در محل کار شامل همگان می‌شود، کمپانی‌ها باید گام‌هایی بردارند تا به‌طور ویژه به زنان آگاهی و قابلیت دسترسی دهند.

محمدرضا اسماعیل طهرانی

فرآیند پیشبرد یک پروژه از شروع تا پایان

به‌عنوان مدیر دفتر مدیریت پروژه شرکت داده‌پرداز پویای شریف، فرآیند پیشبرد یک پروژه از زمان شروع تا پایان آن را به‌تفکیک عنوان خواهم کرد و در شماره‌های بعدی نیز به تشریح برخی از این مراحل می‌پردازم. هدف اصلی این مقاله ارائهٔ یک شیوهٔ مدون و مشخص است که بارها و بارها مورد آزمون قرار گرفته و پروژه‌های موفق زیادی از این فرآیند به‌دست آمده‌اند. اولین موضوعی که لازم است در تمام شرکت‌های نرم‌افزاری مورد بررسی و بحث اساسی قرار گیرد، تعیین شیوه و روش کاری است. مسئلهٔ اصلی که در این میان مطرح است هم‌گام حرکت کردن با روش‌های روز است؛ زیرا، بدون تعارف، برخی از این متدها علی‌رغم موفقیت‌های زیادی که در تئوری دارند، در عمل، با توجه به موضوعاتی چون مسائل فرهنگی که همهٔ طرف‌های پروژه را در برمی‌گیرد، به موفقیت چندانی دست نمی‌یابند. به‌عنوان مثال، قطعاً عدم ضرورت مستندسازی کل پروژه در مرحلهٔ قبل از شروع در روش چابک (Agile)، موجب سوءتفاهم کارفرماهای نرم‌افزاری می‌شود و عملاً برای پیمانکار نیز این مشکل منجر به کاهش میزان کیفیت کار جهت تسریع در تحویل پروژه خواهد شد که با این رویکرد، هردو طرف متضرر خواهند شد. لذا باید به‌صورت واقع‌بینانه به انتخاب روش بپردازیم. تجربه نشان داده ما به مستندسازی نیاز مبرم داریم و نباید از این موضوع غفلت شود. این مستندسازی یک استاندارد و ساختار مشخص دارد که غالباً به آن SRS می‌گوییم که در شماره‌های بعدی بیشتر در خصوص آن توضیح می‌دهیم.
فرض بر اولین زمان تماس کارفرما جهت ارائه نیازمندی و سفارش خود گذاشته و مراحل را به‌شرح زیر توضیح خواهیم داد:

۱. تماس کارفرما با مجموعه جهت ارائهٔ نیازمندی: کارفرماها در این مرحله به دو گروه تقسیم می‌شوند:

- **کارفرمایی که تیم فنی دارد:** تیم فنی به تیمی گفته می‌شود که توانایی آنالیز و تحلیل نیازمندی و مستندسازی آن را داشته باشد. با این حال، کارفرما با آگاهی به آنچه نیاز دارد به پیمانکار درخواست زمان و هزینه اجرا می‌دهد.
- **کارفرمایی که تیم فنی ندارد:** این موضوع باعث می‌شود تا تصمیم‌گیری در خصوص اینکه مستندسازی به‌صورت مشاوره‌ای یا برون‌سپاری انجام شود، مورد بحث و بررسی قرار گیرد.

۲. آماده شدن مستند طبق یک ساختار مشخص: برای پروژه‌ها مستندات متفاوتی وجود دارد که به‌اختصار در زیر به آن اشاره شده است:

- **مستند RFP:** شرح نیازمندی‌های کلی مشتری به زبان خود ایشان که غالباً هیچ زاویه فنی ندارد.
- **مستند پروپزال:** اضافه شدن زمان و هزینه تخمینی طبق موارد مندرج در RFP که اصلاً نهایی نیست.
- **مستند CRS:** مستندی که از روی RFP نیازمندی را به‌صورت ماژول‌های مجزا و تفکیک‌شده تقسیم می‌کند.
- **مستند SRS:** مستندی که از روی CRS با جزئیات، هر ماژول و ویژگی‌های آن را تشریح می‌کند.
- **مستند وب استراتژی:** مستندی که از روی SRS به طرح‌های بخش‌به‌بخش پروژه می‌پردازد.
- **مستند SDD:** مستندی که قبل از پیاده‌سازی به‌صورت فنی به راه‌حل‌های مسئله می‌پردازد.

۳. اعلام ریز متره پروژه: این بخش به‌صورت کلی به دو قسمت تقسیم می‌شود:

- **اعلام زمان توسط دفتر مدیریت پروژه:** در این قسمت نقش‌های موردنیاز در پروژه اعم از برنامه‌نویس، مدیر پروژه، کارشناس کنترل پروژه، امنیت، طراحی، پشتیبانی و غیره، تعیین می‌شوند و همچنین برای بخش‌های قیدشده در مستند SRS، اعلام ساعت موردنیاز انجام می‌شود.
- **اعلام هزینه توسط تیم مالی:** برحسب زمان‌های اعلام شده توسط دفتر مدیریت پروژه، هزینه به تفکیک بخش برای کارفرما محاسبه می‌شود.

۴. عقد قرارداد: در صورت موافقت، مراحل حقوقی عقد قرارداد صورت می‌گیرد که هر قرارداد دو قسمت دارد:

- **بخش حقوقی:** که تعهدات حقوقی طرفین مورد بررسی قرار می‌گیرد.
- **بخش فنی:** که عین مستند SRS به‌عنوان محل توافق در قرارداد قید می‌شود.

۵. ارجاع قرارداد به دفتر مدیریت پروژه
۶. تشکیل تیم فنی موردنیاز برای پروژه توسط تیم مدیریت پروژه
۷. اولین جلسهٔ بررسی نیازمندی‌ها و رفع ابهام تیم تشکیل‌شده که به‌صورت داخلی است؛
۸. برنامه‌ریزی اولیهٔ تحویل پروژه به تفکیک بخش در مستند
۹. ارسال ایمیل درخواست جلسه جهت معارفه تیم و ارائه جدول زمان‌بندی پیشبرد پروژه؛ در این جلسه کارفرما به‌همراه پیمانکار می‌تواند اولویت‌بندی‌های تحویل را اصلاح نماید و در نهایت خروجی این جلسه باید یک برنامه‌ریزی تحویل خرد خرد پروژه به کارفرما باشد.
۱۰. شروع تحلیل و مستندسازی SDD
۱۱. هم‌زمان با شروع مستند SDD تیم گرافیک با کارفرما در حال مذاکره و برگزاری جلسات برای تنظیم طرح‌های پروژه هستند.
۱۲. نهایی شدن طرح‌ها و تنظیم مستند وب استراتژی
۱۳. تنظیم نمونهٔ اولیه پروژه بر حسب طرح‌های تنظیم‌شده و نهایی‌شده توسط تیم گرافیک و کارفرما
۱۴. ارجاع طرح‌های نهایی و تأییدشده به تیم فنی جهت شروع کدنویسی
۱۵. شروع کدنویسی پروژه
۱۶. تحویل بخش به‌بخش پروژه طبق برنامه‌ریزی صورت گرفته.

دقت شود که خیلی از مراحل بالا به‌صورت موازی و هم‌زمان پیش می‌رود، لذا دغدغه‌ای برای دیرکرد و یا طولانی بودن پروسه نخواهیم داشت.
به‌علاوه اینکه مستندات مطرح‌شده هیچ کدام خللی در پیشبرد پروژه از نظر فنی نخواهند داشت و صرفاً با هدف آینده‌نگری، توسعهٔ سریع‌تر و البته پشتیبانی بهتر، تنظیم می‌شوند. در شماره‌های آتی مجله، در مورد بندهای مختلف این مقاله توضیحات کامل‌تر و شفاف‌تر ارائه می‌شود. سعی خواهیم کرد، راه‌حل‌های هر کدام و ابزارهایی را که طی کردن این فرآیند را تسریع می‌کند ارائه کرده تا تعامل بیشتری برای موفقیت پروژه‌ها با خواننده‌های خود داشته باشیم.

خدمات دولت هوشمند

www.iran.gov.ir

مدرس آنلاین

هر زمان و هر مکان، پای درس بهتـــرین 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