

شکست در تحقق اهداف تیمها (Spotify Model)

Spotify از چارچوب Spotify استفاده نمی‌کند. شما هم نباید انجام بدهید!

پیش‌درآمد: نوشتار پیش رو ترجمه‌ای از مقاله Failed Squad Goals به قلم Jeremiah Lee است و نگارنده به دلیل داشتن یک سال تجربه کاری در Spotify Model برخی تجارب شخصی خود را که با نکات مورد اشاره در مقاله اصلی مطابقت داشته را هم به صورت یک مورد مطالعاتی مورد تأکید قرار داده‌است.

از بین جذابیت‌های فرهنگ استارت‌آپی، در عمل تعداد کمی از آنها مانند سرعت و چابکی یک تیم کوچک مطلوب هستند. حفظ این احساس که شرکت در حال رشد است را می‌توان به چالش کشید. در سال 2012 شرکت Spotify روش کار خود را به صورت عمومی منتشر نمود. وقتی در مصاحبه‌ای برای نقش مدیر محصول در Spotify در استکهلم در سال 2017 شرکت کردم، از دیدن مدل Spotify بسیار هیجان زده شدم. با این وجود، استخدام کننده قبل از اولین مصاحبه مرا شگفت زده کرد. او به من هشدار داد که انتظار نداشته باشم که Spotify یک آرمانشهر چابک باشد. من در حالی به شرکت پیوستم که تعداد نیروهای آن در طی 18 ماه سه برابر شده و به 3000 نفر رسیده بود. من فهمیدم که مدل squad مشهور فقط همیشه به صورت یک رؤیا و آرزو مطرح بوده و هرگز به طور کامل اجرا نشده‌است. من شخصا شاهد یک هرج و مرج سازمانی بوده‌ام که در آن مدیریت شرکت به تدریج به روش‌ها و ساختارهای سنتی پیشین بازگشت. جالب اینجاست که وقتی از همکارانم در واحد مدیریت منابع انسانی می‌پرسیدم که چرا با اینکه ما در عمل این را مدل به طور کامل در شرکت به کار نمی‌بریم هنوز در جلسات استخدام آن را به مصاحبه‌شوندگان ارائه می‌دهیم و دست کم قسمت‌هایی که تغییر داده‌ایم را بروز نمی‌کنیم، آنها به شکل طعنه‌آمیزی برخورد می‌کردند که محتوای این مدل برای استخدام عالی هستند.

من دیگر در Spotify کار نمی‌کنم، بنابراین تجربه خودم را مستقیماً با شما به اشتراک می‌گذارم. مدل مبتنی بر Squad در Spotify شکست خورد و اگر شما هم در شرکت خودتان آن را استفاده نمایید شما هم شکست خواهید خورد. هر چند شما لازم نیست برای باور کردن این موضوع به حرف من استناد نمایید. نویسندگان Spotify Model و تعدادی از Agile Coach هایی که در Spotify کار می‌کردند برای سالیان متمادی به مردم گفته‌اند که آن را برای کسب و کارهای خود کپی نکنند. اما متأسفانه، حقیقت به همان سرعت یا به همان اندازه‌ای که بیشتر مردم آنرا باور کنند، گسترش نمی‌یابد. نمونه‌هایی از هشدارهای قبلی را اشاره می‌کنم. Joakim Sundén که بین سال‌های 2011 تا 2017 به عنوان Agile Coach در Spotify مشغول بوده اینگونه اظهار می‌کند:

“حتی در زمان نگارش مدل ما به آن عمل نمی‌کردیم. بخشی از آن جسورانه و بخشی هم تقریبی بود.

مردم واقعاً در تلاشند تا چیزی را که واقعاً وجود نداشت، کپی کنند.”

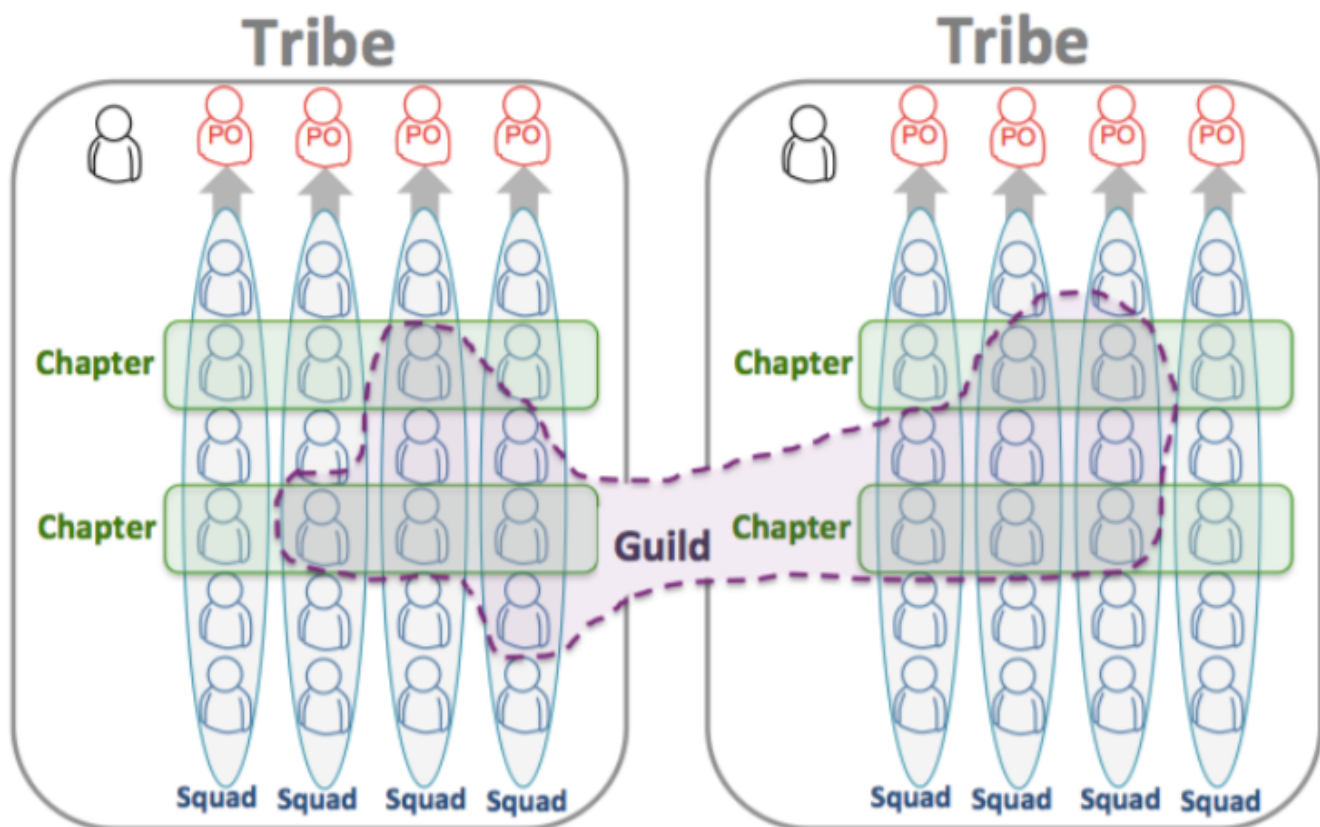
و یا این نقل قول از Anders Ivarsson جزو نویسندگان Spotify Model است:

“این برای من نگران کننده است که نگرش مردم به کارهای ما همانند یک مدلی است که فقط باید آن را کپی و اجرا کرد. ما واقعاً خیلی تلاش می کنیم تا تأکید کنیم که مشکلاتی نیز داریم. اینکه همه چیز درخشان نیست و به خوب پیش نمی رود و تمام squad های ما فوق العاده شگفت انگیز نیستند.”

مروری مختصر بر Spotify Model

شما می توانید در کمتر از 30 دقیقه Spotify Model را از [اینجا](#) مطالعه کنید و یا در صورتی که با آن آشنایی دارید این بخش از نوشتار را رد بشوید و به قسمت بعدی بروید. در این بخش مختصراً به اجزای مدل پرداخته شده است.

در مدل Spotify تیمها متشکل از گروهی از طراحان و مهندسان نرم افزار با طیف وسیعی از تخصصها هستند و **squad** نامیده می شوند. هدف اصلی طراحی این ساختار تیمی این است که هر تیم بدون وابستگی به تیمهای دیگر بتواند عملکرد موفقی داشته باشد. هر Squad باید بتواند همانند یک مینی استارتاپ مستقل عمل کند و انتظار می رود یک مدیر محصول بتواند به عنوان **مدیرعامل کوچک (Mini CEO)** برای هدایت آن عمل نماید. ساختار مدیریتی در این مدل سنتی است. گروهی از Squad های مرتبط به هم **قبیله (Tribe)** نامیده می شوند و هر قبیله دارای یک **رهبر محصول (Product Director)** است که **رهبر قبیله (Tribe Lead)** نامیده می شود و مدیران محصول عملکرد و دستاوردهای تیم خود را به او گزارش می دهند. مشابه آن برای طراحان و مهندسان نرم افزار هم ساختار مدیریتی مشابهی وجود دارد که خارج از ساختار Squad ها اداره می شوند و آنها با عنوان **Chapter Lead** متمایز می گردند. Chapter Lead ها طراحان و مهندسان نرم افزار با تخصصهای مشابه را در یک بخش از فرآیند توسعه محصول مدیریت می کنند. به عنوان مثال، کلیه افرادی که روی API Backend ها کار می کنند فارغ از Squad ای که در آن مشغول به کار هستند، یک Chapter Lead مخصوص به خود را دارند. به همین ترتیب همه مهندسين تلفن همراه اندرویدی در Chapter دارای یک مدیر متفاوت هستند و به همین ترتیب برای سایر تخصصها هم عمل می شود. هدف این است که مهندسان بتوانند در صورت نیاز بین تیمهای این Chapter جابجا شوند تا به بهترین وجه نیازهای کسب و کار را برآورده کنند بدون آنکه آنها مجبور به تغییر مدیران شوند. شکل زیر تصویری نمادین از مدل Spotify را نشان می دهد.



همانطور که از شکل پیداست Spotify یک ساختار تیمی ماتریسی با ثبات را با انتخاب نام‌های غیرمعمول برای سازماندهی تیم‌های خود امتحان کرده و سپس آنرا به یک مدل ساختار سازمانی تبدیل نموده است.

چرا Spotify Model کار نکرد؟

در این بخش به اختصار به برخی از دلایل شکست Spotify Model اشاره می‌شود.

مدیریت ماتریسی مسأله اشتباهی را حل کرد

تیم‌های چابک Full Stack به خوبی کار کردند، اما مدیریت ماتریسی مهندسان نرم‌افزار مشکلات بیشتری را نسبت به حل مسائل مورد انتظار به بار آورد:

- تیم‌های Spotify نسبتاً با ثبات بودند. فایده عدم تغییر مدیریت (در سطح محصول و chapter) هنگام انتقال افراد از یک Squad به Squad دیگر چندان زیاد نبود. (در اینجا لازم است مترجم متذکر شود در تجربه مشابهی که در یک شرکت ایرانی داشته این باثباتی تیمی در همه سطوح خصوصاً تیم‌های فنی فاکتور انگیزش را از افرادی که در تیم‌ها جابجا می‌شوند سلب می‌کند. چون جابجایی افراد معمولاً یا به دلیل عملکرد بد صورت می‌گیرد و یا عملکرد خوب. در هر دو صورت به غیر از خود فرد - در صورتی که از علت جابجایی مطلع نشود - تیم مقصد نیز از بخشی از دلیل واقعی انتقال ناآگاه است و باید تلاطمی را برای پذیرش عضو جدید و سازگارسازی او با تیم پشت سر بگذارد. همانند وضعیتی که در یک سازمان با ساختار سنتی اتفاق می‌افتد. پس نتیجه می‌گیریم در این خصوص Spotify Model نه تنها

مسأله‌ای را حل نکرده است بلکه به آن افزوده است. نمی‌توان از فردی که جدیداً از تیمی دیگر آمده انتظار داشت در این تیم کارایی بیشتری داشت چون در هر حال مدیریت فنی او در سطح Chapter تفاوتی نداشته و اگر قرار بود کار کردن با این فرد در سطح Chapter جواب بدهد نیاز به تغییر تیم نبود و باید در همان تیم قبلی هم جواب گرفته می‌شد.

▪ مدیران مهندسی (Chapter Leads) در این مدل مسئولیت کمی در پیشرفت شغلی افرادی که آنها را مدیریت کرده‌اند داشتند. همچنین توانایی آنها برای کمک به پیشرفت مهارت‌های افراد نیز محدود بود. یک مدیر مهندسی به اندازه کافی سایر افراد حاضر در تیم را مدیریت نمی‌کند و یا به اندازه کافی در مسائل روزمره Squad ها شرکت نمی‌کند تا به طور مستقل میزان درگیری در تیم را ارزیابی کند.

▪ بدون داشتن یک مدیر مهندسی واحد که مسئول مهندسین یک تیم باشد، مدیر محصول به عنوان یک Mini CEO فاقد یک همتای معادل خود (Mini CTO) در تیم بود و مشخصاً هیچ فردی برای تحویل تیم مهندسی به او در سطح تیم وجود نداشت یا کسی که قادر به مذاکره در زمینه اولویت کارها در سطحی معادل با مدیر محصول باشد. (در اینجا لازم است مترجم متذکر شود در تجربه مشابهی که در یک شرکت ایرانی داشته برای حل این عارضه یک رویداد بررسی فنی در سطح Chapter طراحی شده بود که انتظار می‌رفت بتواند از زاویه دید فنی این امکان را برای Squad فراهم کند که از نظر فنی در خصوص اولویت‌های انجام کار مواردی را به مدیر محصول متذکر شود اما این مکانیسم مستلزم برگزار کردن جلساتی با حضور زینفعان فنی بود که لزوماً هم‌گام با تحولات محصول به جمع‌بندی و نتیجه‌گیری نمی‌رسیدند و ادامه توسعه محصول را با مسائل پیش‌بینی نشده‌ای مواجه می‌ساخت.)

▪ هنگامی که اختلافات در تیم فنی به وجود آمد، مدیر محصول برای پیش بردن کار به مذاکره با همه مهندسان تیم نیاز داشت. اگر مهندسان نتوانند به اجماع برسند، مدیر محصول بعضاً باید به همان تعداد مدیران مهندسی مراجعه داشته باشد. زیرا تخصص‌های مهندسی متنوعی در تیم وجود دارد. در یک تیم دارای تخصص‌های طراحی، برنامه‌نویسی backend و برنامه نویسی اندروید دست کم 3 مدیر مهندسی مختلف ممکن است نیاز به مشارکت داشته باشند. و اگر آن دسته از مدیران مهندسی هم نتوانند به اجماع برسند، مسأله رخ داده در سطح یک تیم باید به مدیر مهندسی ارشد (Engineering Director) ارجاع داده شود. (در اینجا لازم است مترجم متذکر شود در تجربه مشابهی که در یک شرکت ایرانی داشته بخش زیادی از این نوع از تعارضات در غیاب یک فرآیند بررسی کامل تا بالاترین سطوح فنی یا منجر به کاهش کیفیت محصول خروجی به دلیل بی‌اعتنایی به یک و یا چند بخش از کار می‌شد و یا هنگامی که بر طبق مدل تمامی سلسله مراتب برای تصمیم‌نهایی طی می‌شد زمان زیادی از ابتدای قضیه گذشته بود که نتیجه آن بعضاً به تولید دیر هنگام جنبه‌های تازه محصول منجر می‌شد.)

“Chapter Lead کسانی هستند که به شما در رشد فردی کمک می‌کنند. آنها در واقع با هیچ تیمی کار نمی‌کنند. آنها گزارشات مستقیمی از همه تیم‌ها دارند. آنها واقعاً هیچ مسئولیتی در قبال تحویل محصول ندارند و هیچ مسئولیتی را هم بر عهده نمی‌گیرند. و در نتیجه راحت‌تر است که صاحب محصول را به عنوان مدیر تیم مشاهده کرد.” (Joakim Sundén)

برای ورود به دنیای اسکرام می توانید از دوره غیر حضوری "آشنایی مقدماتی با اسکرام" شروع کنید

مدرسه اسکرام

دوره غیر حضوری

آشنایی مقدماتی با اسکرام

مدت: ۹۵ دقیقه ویدئو

مدرس: سید مهدی موسیقی

www.ScrumSchool.ir

دروس آموخته از اشتباهات Spotify

- یک تیم متشکل از سه گانه محصول/ طراحی/ مهندسی معمولاً مهندسين بیشتری نسبت به طراحان یا مدیران محصولات دارد. داشتن یک مدیر مهندسی واحد برای مهندسان تیم، امکان پاسخگویی برای تعارضات داخل تیم را ایجاد می کند.
- مدیران محصولات باید از نظر مهندسی دارای یک همتای معادل باشند. مدیران محصولات باید در اولویت بندی کارها پاسخگو باشند. مدیران مهندسی باید نسبت به نحوه اجرای مهندسان تیم پاسخگو باشند که باعث می شود امکان مذاکره در خصوص سرعت و کیفیت با مدیر محصول به وجود آید.

Spotify خودمختاری در تیمها را تثبیت کرد

وقتی یک شرکت کوچک است، تیمها مجبورند طیف گسترده ای از کارها را برای ارائه محصولات انجام دهند و مجبورند بارها و بارها ابتکار عمل را تغییر دهند. هنگامی که شرکت در مسیر تغییر از استارتاپ به مجموعه های بزرگتر هستند، عملکردهای تکرارپذیر (که پیچیدگی های عملکردی آنها در قالب فیچرهای محصول پاسخ داده شده) به تیم های جدیدی منتقل می شوند تا با کاهش حجم کارهای روتین شده بهره وری سازمان افزایش یابد. با وجود تیم های بیشتر، فرکانس نیاز به یک تیم برای پاسخگو بودن در برابر پیچیدگی ها کاهش می یابد. این تغییرات به تیمها این امکان را می دهد تا عمیق تر و طولانی مدت در مورد مشکلاتی که برای حل آنها برنامه ریزی شده اند متمرکز باشند. البته این موضوع سرعت یافتن iteration ها را تضمین نمی کند زیرا هر مسئولیتی که یک تیم برای افزایش تمرکز خود واگذار می کند، به یک وابستگی متقابل جدید تبدیل می شود.

در اینجا مترجم مثالی را جهت شفافیت بیشتر موضوع در تجربه مشابهی که در یک شرکت ایرانی داشته اضافه می‌کند:

“در سازمانی با ساختار Spotify Model که به عنوان مدیر محصول مشغول به کار بودم به دلیل استفاده از متدولوژی Microservices یک مفهوم مالکیت برای برنامه‌نویسان و مهندسانی که برنامه‌نویسی می‌کردند به صورت یک فرهنگ کاری جا افتاده بود که هر فردی مسئول کدی است که مالکیت مایکروسرویس آنرا بر عهده دارد و هر پرورسانی در کد مایکروسرویس دیگران نیازمند دریافت تأیید از مالک آن بود. این موضوع منجر به تقویت یک خرده فرهنگ کار با مسئولیتی محدود در شرکت شده بود و برنامه‌نویسان در اغلب اوقات در تضادی آشکار با منافع محصول تنها به حوزه عمل خود و تغییرات آن حساس بودند و هنگامیکه توسط Agile Coach شرکت فرآیندی جهت ارسال هشدار به واحد عملیات شرکت طراحی شد تا خلاءهای مدیران محصول در مالکیت محصولات و وقایع مربوط به آنها در محیط عملیاتی پوشش بهتری داشته باشد، این خرده فرهنگ به شیوه‌ای عمل کرد در همان ابتدای امر همه مدیران محصول و برنامه‌نویسان بر آن شدند تا مسئولیت ذاتی عملکردهای محصول در تیم محصول را با تولید هشدار در خصوص کوچکترین رخدادهای بدون توجه به حجم هشدارهای تولید شده از سر خود باز کنند و موضوع را به صورت یک روتین که باید در جای دیگری بررسی شود از حوزه کاری خودشان بیرون بیاورند و تا زمان انتشار این مقاله هیچکدام از تلاش‌ها برای اصلاح رویه‌های تولیدی در این زمینه به ثمر ننشسته بود. همانند انتقال ترافیک یک تقاطع شلوغ با باز کردن دوربرگردان‌هایی کمی بالاتر و پایین تر از تقاطع که به جای وابستگی به یک نقطه آن را به دو نقطه منتقل می‌کرد.”

در Spotify روند مشترکی برای همکاری متقابل بین تیم‌ها تعریف نشد. اجازه دادن به هر تیم برای داشتن یک روش کار منحصر به فرد به این معنی است که هنگامی که همکاری با تیم‌های دیگر ضرورت پیدا می‌کند به دلیل روش‌های کاری مجزا برای ایجاد تعامل بین تیم‌ها ضرورتاً نمی‌توان از یک پروتکل استاندارد تبعیت کرد و تعاملات مربوطه لزوماً به روش منحصر به فردی نیازمند خواهد بود و این منجر به کاهش بهره‌وری در سطح سازمان می‌شود.

مدل Spotify زمانی منتشر شد که Spotify هنوز شرکتی بسیار کوچکتر از وضعیت فعلی آن بود. بر طبق گفته Anders Ivarsson (از نویسندگان اصلی این مدل)، قرار بود مفاهیم مطرح شده در این مدل بر اساس تجارب به دست آمده بعدی به روز شوند. اون بخشی از مدل که به خودمختاری تیمی اشاره دارد تکمیل و تشریح شده اما قسمت‌های Alignment و پاسخگویی هرگز تکمیل نشدند.

دروس آموخته از اشتباهات Spotify

- خودمختاری مستلزم وجود Alignment است. اولویت‌های شرکت باید توسط رهبری آن تعریف شوند. خودمختاری به معنای این نیست که تیم‌ها هر کاری را که می‌خواهند انجام بدهند.
- فرآیندهای همکاری متقابل تیمی باید تعریف شوند. خودمختاری به معنای رهاسازی تیم‌ها برای خودسازماندهی هر مشکلی نیست.

- چگونگی اندازه‌گیری موفقیت باید توسط مدیریت تعریف شود تا افراد بتوانند به طور مؤثر در اولویت‌بندی‌های وابستگی‌های متقابل در سطح تیم به مذاکره بپردازند.
- خودمختاری مستلزم پاسخگویی است. مدیریت محصول برای تولید ارزش پاسخگو است. تیم هم پاسخگوی پیشرفت‌های “انجام شده” است. تیم‌های بالغ می‌توانند خودمختاری خود را با توانایی در بیان ارزش تجاری، ریسک‌ها، یادگیری و اقدامات بهینه آتی مشخص کنند.

Joakim Sundén که بین سال‌های 2011 تا 2017 به عنوان Agile Coach در Spotify مشغول بوده اینگونه اظهار می‌کند:

“اگر قرار بود یک کار متفاوت را انجام دهم، می‌گویم ما نباید خیلی به خودمختاری توجه می‌کردیم.”

همچنین Jason Yip که سابقه همکاری به عنوان Agile Coach با Spotify را دارد در سال 2015 اینگونه اظهار می‌کند:

“اگر روش‌های متناقضی برای کار کردن دارید، حرکت برای افراد دشوارتر است. این یک موضوع دو طرفه است یعنی اگر حرکت برای افراد دشوار باشد، به احتمال زیاد شما روش‌های ناسازگار برای کار کردن دارید. این یک روند است که ناگهان برای شما آشکار می‌شود: شما واقعاً در همان شرکت کار نمی‌کنید، در حال کار کردن برای انواعی از خرده فرهنگ‌های عجیب و غریب هستید.”

همکاری یک شایستگی مفروض بود نه واقعی

در حالی که Spotify به تیم‌ها اجازه می‌داد تا نحوه کار خود را کنترل کنند، بسیاری از افراد درک درستی از شیوه‌های چابک در توسعه محصول نداشتند. این امر باعث شد تیم‌ها با آمیدی واهی به دنبال پیدا کردن ترکیبی از روش‌های iterative که به آنها در بهبود نتیجه کمک کند، باشند. افراد فاقد یک زبان مشترک بودند تا به طور مؤثر در مورد مشکلات فرآیند، آموزش برای حل آنها و تجربه برای ارزیابی عملکرد صحبت کنند. مدل Spotify واقعاً چابک نبود؛ فقط آبشاری نبود.

یک عارضه دیگر Agile Coach ها بودند که Spotify برای آموزش تیم‌ها و پیشرفت‌های فرآیند از مشاوره آنها استفاده می‌کرد. در حالی که کاملاً مشخص بود تعداد مربیان کافی برای کمک به هر تیم وجود نداشت. درگیری Agile Coach با یک تیم به ندرت به اندازه کافی طولانی بود که بتواند یک پروژه بهبود را تکمیل کند یا به تیم در ارزیابی عملکرد کمک کند. علاوه بر این، آنها در قبال هیچ چیز پاسخگو نبودند.

“کنترل بدون شایستگی آشفته است” David Marquet

دروس آموخته از اشتباهات Spotify

- همکاری مهارتی است که نیاز به دانش و تمرین دارد. مدیران نباید فرض کنند که افراد بدون

آموزش و تمرین درک درستی از شیوه‌های چابک دارند.

- هنگامی که یک شرکت به اندازه کافی بزرگ شد، تیم‌ها برای هدایت برنامه‌ریزی در تیم و ایجاد ساختار بین تیم‌ها به پشتیبانی اختصاصی نیاز دارند. مدیریت برنامه (Management Program) می‌تواند برای فرآیند برنامه ریزی مسئول و پاسخگو باشد. داشتن مدیران برنامه اختصاصی به تیم‌ها امکان انطباق رفتاری مشابه آنچه مدیران محصول و مدیران مهندسی با شایستگی‌های مربوطه خود انجام می‌دهند را می‌دهد.

یک متدولوژی که تغییر آن دشوار است

هنگامی که اسکرام معانی جدیدی را برای مجموعه‌ای از کلمات مانند Sprint و Burn-Down وارد کرد، این کار را به این دلیل انجام داد که داشت مفاهیم جدیدی را ارائه می‌داد که به اسامی جدید نیاز داشتند. Spotify برای توصیف نحوه کار خود، واژگان Missions Tribes Squads Guilds و Chapter Leads را معرفی کرد و این توهم را ایجاد نمود که مفاهیمی را ارائه می‌کند که استفاده از کلمات غیرمعمول برای آنها به درک مفهوم کمک می‌کند. این در حالی است که اگر مترادف‌های غیرضروری را از ایده‌های پشت آنها حذف کنیم، مدل Spotify چیزی بیشتر از مجموعه‌ای از تیم‌های عملکردی با خودمختاری بیش از حد نیست و ساختار مدیریتی برای ارائه ندارد. لذا با استفاده از آن سقوط نکنید. شاید اگر Spotify با عناوین واقعی خود به این ایده‌ها اشاره می‌کرد و آنها به جای مواجهه با تغییر در هویت فرهنگی خود، به سادگی برای یافتن فرآیندهای داخلی (که به خوبی کار می‌کنند) وقت صرف می‌کردند، می‌توانستند بهتر و عادلانه‌تر آنها را ارزیابی کنند و به نتیجه‌ای که واقعا کار می‌کند می‌رسیدند.

دروس آموخته از اشتباهات Spotify

- بیشتر کسب و کارها تنها می‌توانند برخی از زمینه‌های نوآوری را حفظ کنند. فرآیندهای داخلی به ندرت می‌توانند محلی برای ارائه راه‌حل‌های نوآورانه‌ای باشند که یک شرکت را در بازار از دیگران متمایز نماید. مطالعه گذشته به کسب و کارها اجازه می‌دهد تا زمینه‌های بهتری را برای نوآوری انتخاب کنند.
- بهینه‌سازی را برای درک بهتر استفاده کنید. هر چیز جدیدی که شخص باید یاد بگیرد تا بتواند در سازمان شما مؤثر باشد باید بر اساس ارزش آن ارزیابی شود.
- Teams Departments Business Units و Managers مفاهیمی هستند که به طور مؤثر نقش‌ها و مسئولیت‌های ساختار سازمان را بدون نیاز به مترادف‌های Spotify آن نمایندگی کنند بدون آنکه از روشی که کار سازنده آنها را ناکام می‌کند تبعیت کرده باشند.

به جای آن این را استفاده کنید!

(البته این شوخی است! هیچ راه‌حل جایگزین آنی وجود ندارد)

شما ممکن است مدل Spotify را کشف کرده باشید زیرا سعی داشتید نحوه ساختار تیم‌های خود را

بفهمید. اما به این مقدار اکتفا نکرده و به تحقیق ادامه دهید. رهبران شرکت‌هایی که آزمایش‌های طولانی‌تری را تحمل کرده‌اند، چیزی بیشتر از Spotify منتشر کرده‌اند. انسان‌ها تا زمانی که انسان بوده‌اند تلاش کرده‌اند بفهمند که چگونه با هم همکاری کنند. عصر صنعتی و عصر اطلاعات برخی از محدودیت‌ها را تغییر داده است، اما کارهای پژوهشی در حوزه تئوری‌های سازمان، حقایق مستقل از زمان درباره آنچه انسان‌ها برای موفقیت در یک مجموعه ضروری می‌دانند، پیدا کرده‌اند. آنچه واضح است این است که Spotify در سال 2012 و هنگام انتشار این مدل متوجه نشده که چگونه سرعت و چابکی یک تیم کوچک را در یک سازمان بزرگ حفظ کند. اما این شرکت فراتر از مدل مشهور خود تکامل یافته و به دنبال پاسخ‌های بهتری بوده است. شما هم باید چنین کنید و به دنبال یافتن پاسخ‌های مطلوب کسب و کار خودتان باشید.

مقاله اصلی: <https://www.jeremiahlee.com/posts/failed-squad-goals>

مدرسه اسکرام مرجع حرفه ای آموزش اسکرام