

6 دلیل برای اینکه نباید مهندس دواپس (DevOps) را استخدام کنید!



دواپس (DevOps) یک فرهنگ است، نه یک عنوان. بنابراین، شرکت‌ها باید از اصول آن پیروی کنند، نه اینکه تنها یک مهندس دواپس را برای اصلاح مدل فرآیند تولید نرم‌افزار در کل سازمان، استخدام کنند.

ظهور ابزارهای قدرتمند برای این فرهنگ و گسترش روش‌های تولید نرم‌افزار در سازمان‌ها باعث می‌شود که امروزه توسعه‌دهندگان مجهزتر شوند و بتوانند نرم‌افزارهای خود را در تمام حوزه‌ها از جمله گسترش نرم‌افزار، امنیت، فضای ابری و ... با سرعت و کیفیت بالاتری به پایان برسانند.

خارج شدن این فرهنگ جدید از چرخه توسعه می‌تواند یک پارچه‌سازی تیم توسعه و تیم عملیات را ناکام بگذارد. در ادامه ی این مقاله از سایت سماتک دیدگاه یوری زیدنورگ (Uri Zaidenwerg) مدیر دواپس در شرکت Replix بررسی می‌کنیم که چرا به جای ایفای نقش مهندس دواپس در یک سازمان باید بر ایجاد فرهنگ آن تمرکز کنید.

1. دواپس یک فرهنگ است، نه یک نقش

دواپس (DevOps) بهترین عملکرد را به‌عنوان یک فرهنگ در سراسر دنیای فناوری اطلاعات دارد. زیدنورگ می‌گوید: به جای استخدام یک شخص برای ایفای نقش مهندس دواپس در سازمان، باید همه‌ی توسعه‌دهندگان، در فرآیند اجرا و عملیات شرکت کنند.

این به این معنی است که باید توسعه‌دهندگان با شاخص‌های کلیدی عملکرد (KPI) این فرهنگ از جمله: فرکانس استقرار، زمان استقرار، تغییر نرخ شکست، زمان شناسایی، در دسترس بودن و ... آشنا شوند تا دید بهتری درباره‌ی نرم‌افزار پیدا کنند. در انتهای این مقاله درباره‌ی شاخص‌های کلیدی عملکرد (KPI) این دانش جدید به تفصیل توضیح خواهیم داد.

2. توسعه‌دهندگان کدهای برنامه خود را می‌شناسند

توسعه‌دهندگان بهتر از هر شخص دیگری با کدهای برنامه‌ی خود آشنا هستند. تشخیص خطاها در یک محیط نا آشنا مشکل است و فقط برنامه‌نویس‌ها نکات مربوط به هرکد را به‌درستی درک می‌کنند. بنابراین، اگر به مهارت‌های دواپس (DevOps) مجهز شوند، می‌توانند مشکلات را بسیار موثرتر از دیگر افراد حل کنند.

زیدنورگ (Zaidenwerg) در این رابطه توضیح می‌دهد: «نظارت بر در دسترس بودن برنامه‌ای که کدهای آن توسط شخص دیگری نوشته شده، مشکل و چالش برانگیز است.»

اگر یک مهندس دواپس نتواند خطاها را به سرعت به فرآیندهای برنامه مرتبط کند، می‌تواند به دسترسی و SLA ها آسیب برساند. زیدنورگ خاطر نشان کرد که در عوض، تشویق توسعه‌دهندگان به تشخیص مشکلات می‌تواند زمان اصلاح نرم افزار را کاهش دهد زیرا آن‌ها با برنامه بیشتر آشنایی دارند.

3. دیدگاه متفاوت توسعه دهندگان نسبت به محیط اطراف

توسعه‌دهندگان علاوه بر درک نحوه عملکرد برنامه‌های کاربردی، دیدگاه منحصر به فردی نسبت به محیط اطراف دارند. آن‌ها بهتر از هر فرد دیگری متوجه شبکه‌ی پیچیده وابستگی‌ها و اتصالات متقابلی می‌شوند که یک سیستم میکروسرویس را تشکیل می‌دهند.

برای اینکه ببینید چه خبر است، واقعاً باید بدانید که چگونه کدنویسی کنید. در واقع مهندس دواپس (DevOps) باید توسعه دهنده نرم افزار باشد. واگذاری وظایف برنامه نویس‌ها به مدیری که دانش کدنویسی ندارد، می‌تواند توانایی آن‌ها را به شدت محدود کند. مهندس دواپس قادر نخواهد بود که کدها را برای حل مشکلات واقعی یا رسیدگی به درخواست‌ها با اطمینان بررسی کنند.

4. دواپس باید به یکپارچه سازی تیم توسعه و تیم عملیات کمک کند



واگذاری مسئولیت‌های این فرهنگ جدید به یک تیم متمرکز بر خلاف هدف اصلی شکل‌گیری آن است. زیدنورگ (Zaidenwerg) می‌گوید: «دواپس (DevOps) برای کوچکتر کردن شکاف بین تیم توسعه و تیم عملیات اختراع شد.» بنابراین، در نظر گرفتن نقش جداگانه‌ی مهندس دواپس، برای این فرهنگ با اصول پیدایش آن در تناقض است.

تام مارش (Tom Marsh)، توسعه‌دهنده اصلی Novartis در Quora می‌گوید: «دلیل اصلی تشکیل دواپس این است که تیم‌های توسعه‌دهنده قدرت کنترل عملیات را داشته‌باشند، نه اینکه شما متخصصانی را استخدام کنید که اسکریپت‌ها را بنویسند.»

5. یک مهندس دواپس نمی‌تواند با سیل مبارزه کند

همانطور که احتمالاً می‌دانید، به تعداد کافی مهندس دواپس (DevOps) وجود ندارد. تحقیقات Quora نشان می‌دهد که نسبت مهندس‌های دواپس به توسعه دهندگان 1 به 10 تا 1 به 12

است. زایدنورگ می گوید که این نسبت در بعضی از سازمان‌ها بیشتر می‌باشد یعنی در حدود 5 به 50 تا 6 به 120 اما او معتقد است که نسبت صفر به صفر، یک عدد بهینه است.

یک مهندس دواپس به تنهایی نمی‌تواند در برابر موج درخواست‌ها و استقرار برنامه‌ها، پایداری کند. آیا می‌توانید تعداد کدهای نوشته شده را تصور کنید؟. زایدنورگ در این رابطه توضیح می‌دهد: « میزان آتش سوزی که یک مهندس دواپس باید خاموش کند، بسیار زیاد است.» وقتی یک مهندس دواپس به تعطیلات می‌رود چه اتفاقی می‌افتد؟ برای جلوگیری از تاخیر، او توسعه‌دهندگان را تشویق می‌کند تا فرآیند CI/CD خود را از انتها به انتها انجام دهند.

6. اگر وظایف مهندس دواپس به صورت اتوماسیون انجام شود، چه اتفاقی می‌افتد؟

یکی دیگر از اهداف شکل‌گیری فرآیند DevOps و SRE، معرفی اتوماسیون و انجام کارها به شکل خودکار است. اما در این صورت، برای مهندس دواپس (DevOps) چه اتفاقی می‌افتد؟ آیا آن‌ها می‌نشینند و سال‌ها پس از آن، سود حاصل از کار اولیه خود را به دست می‌آورند؟ بنابراین، بهترین و منطقی‌ترین گزینه این است که برنامه نویس‌ها اتوماسیون را در گردش کار خود وارد کنند.

چگونه می‌توانید شاخص‌های کلیدی عملکرد (KPI) دواپس را ردیابی کنید؟

اگر وظایف این فرهنگ جدید را به یک مهندس دواپس (DevOps) محول نکنیم، چگونه می‌توانیم توسعه‌دهندگان را تشویق کنیم که از اصول این فرهنگ پیروی کنند؟ پاسخ زایدنورگ این است که: «به جای سنجش موفقیت با استفاده از معیارهای خروجی توسعه سنتی مانند: بهره‌وری، اسکرام، تعداد وظایف تکمیل شده و.. به فرهنگی نیاز داریم که کار توسعه‌دهندگان را با استفاده از شاخص‌های کلیدی عملکرد (KPI) این دانش جدید اندازه‌گیری کند.

شاخص‌های کلیدی عملکرد (KPI) زیادی وجود دارند که باید در نظر گرفته شوند. در ادامه برخی از آن‌ها را معرفی می‌کنیم:

- **فرکانس استقرار:** هرچند وقت یکبار قابلیت‌ها و ویژگی‌های جدید منتشر می‌شوند؟
- **زمان استقرار:** استقرار یک برنامه چقدر طول می‌کشد؟
- **اتوماسیون:** فرآیند ساخت چقدر خودکار است؟

- **در دسترس بودن:** آیا این برنامه در سطح جهانی در دسترس است و عملکرد بالایی دارد؟
تأخیر و آپتایم (Uptime) آن چقدر است؟
- **مانیتورینگ:** آیا به طور فعال خطاها و مشکلات نرم افزار را رفع می کنید؟
- **بهینه سازی ها:** آیا برنامه مقرون به صرفه است؟ آیا می تواند کارآمدتر باشد؟ آیا منابع بیکار در حال اجرا هستند؟
- **خط مشی به عنوان کد:** آیا سیاست های امنیتی به صورت کد و خودکار هستند؟ زیدنورگ می گوید: «امنیت باید با کد نوشته شود، اگر به صورت دستی انجام شود، رضایت بخش نخواهد بود.»
- **شاخص DRP:** آیا برنامه انعطاف پذیر است؟ طرح بازیابی بلایا (DRP) چیست؟

از میان شاخص های کلیدی عملکرد این فرهنگ جدید، بهینه سازی متمایز است، زیرا افزایش هزینه های رایانش ابری این روزها یک مانع بزرگ برای بهینه سازی نرم افزارها است. زیدنورگ توسعه دهندگان را تشویق کرد تا هزینه های محیط ابری را پیگیری کنند. او می گوید: وقتی نموداری از هزینه های روزانه دارید، می توانید از این دید برای ایجاد سیستم های پشتیبانی که بتوانند منابع بی کار را به صورت خودکار حذف کنند، استفاده کنید. دیگران این ویژگی را به عنوان BizDevSecOps توصیف کرده اند.



قصد ما از عنوانی که برای این مقاله استفاده کرده‌ایم این نبود که بخواهیم ارزش افرادی را که به عنوان مهندس دواپس (DevOps) کار می‌کنند، کم رنگ کنیم، برعکس، استخدام فردی با این تجربه نادر می‌تواند به سازمان‌ها در معرفی اصول این دانش جدید کمک زیادی کند. یک مهندس دواپس می‌تواند به عنوان الگویی برای کمک به پرورش این فرهنگ عمل کند.

علاوه بر استفاده از شاخص‌های کلیدی عملکرد (KPI)، زایدنورگ یک رویکرد از بالا به پایین را برای گسترش اصول این فرهنگ توصیه می‌کند. این روش باید توسط CTO (معمار یا مدیر ارشد فناوری) و با توجه به اندازه شرکت، تایید و هدایت شود. این فرهنگ جدید مخصوصاً برای شرکت‌های SaaS یا IaaS بسیار مهم است.

انتشار بهترین شیوه‌های مدرن ابری نیز برای ایجاد این فرهنگ دارای اهمیت است. هنگامی که از محیط سخت‌افزاری به فضای ابری منتقل می‌شوید، اغلب با فناوری مشابه (متعادل کننده‌های بار، فایروال‌ها، مناطق در دسترس و ...) مواجه می‌شوید که دارای نام‌ها و فرآیندهای پیاده‌سازی متفاوت

برای هر عملکرد هستند و این تفاوت‌های ظریف بین ارائه‌دهندگان خدمات ابری (CSP) باعث بروز مشکلات و پیچیدگی می‌شود.

خانه چیست؟

زایدنورگ یک برنامه نرم افزاری را به خانه شخصی تشبیه می‌کند. صاحب خانه همه جزئیات را می‌داند از جمله این که: ظروف کجا هستند، نحوه کار با نازل دوش، زباله‌ها در چه روزی جمع آوری می‌شوند و... . یک غریبه را به خانه جدید دعوت کنید، در این صورت آن‌ها دچار مشکلات فراوانی خواهند شد.

به طور مشابه، دعوت از یک فرد خارجی برای مدیریت یک زیرساخت خارجی، آن‌ها را دچار مشکل می‌کند. زایدنورگ معتقد است: «اگر به شخص دیگری اجازه دهید با زیرساخت‌ها سروکار داشته باشد، دو گروه دارید که هیچ کدام با کارهای دیگری آشنایی ندارد.»

به طور خلاصه دواپس (DevOps) یک فرهنگ است، نه یک نقش. در نظر گرفتن آن به عنوان نقش مهندس دواپس می‌تواند باعث شکست هدف و کند شدن حرکت تیم شود. در عوض، استفاده از شاخص‌های کلیدی عملکرد (KPI ها) و توانمندسازی توسعه دهندگان با ابزارهای این دانش جدید می‌تواند به آن‌ها کمک کند تا خانه خود را مرتب نگه دارند.

البته این فرهنگ جدید در حال تکامل است و هر سازمانی درک خاص خود را از معنای پیاده‌سازی این دانش جدید خواهد داشت و همان طور که گفتیم تعداد افرادی که به این فرهنگ تسلط دارند، بسیار محدود هستند. به همین دلیل به شما پیشنهاد می‌کنیم که به سرفصل دوره دواپس در سماتک سربزنید تا علاوه بر آشنایی بیشتر با این فرهنگ جدید، بخشی از فیلم آموزشی این دوره را مشاهده کنید.

نوشته شده توسط: نرگس گرامی

منابع: وب سایت DevOps.com