**میکرو‌سرویس‌ها**

سال‌ها است که ما روش های بهتری برای ساخت سیستم‌ها پیدا کرده‌ایم. ما ازآنچه قبلاٌ رخ داده, استفاده از فن آوری‌های جدید و مشاهده این که چگونه یک موج جدید از شرکت‌های فن‌آوری به روش‌های مختلف عمل می‌کنند تا به مشتریان خود و توسعه دهندگان خودشان کمک کنند‌, یاد گرفته‌ایم.

کتاب طراحی دامنه محور اریک ایوانس به ما کمک کرد تا اهمیت نماینده دنیای واقعی در کد خود را درک کنیم و راه‌های بهتری برای مدل کردن سیستم‌های خود به ما نشان داد. مفهوم CD یا همون تحویل مستمر نشان داد که چگونه میتوانیم به طور موثر و کارآمد نرم‌افزار‌مان را به تولید برسانیم, و در ما این ایده را القا می‌کند که ما باید هر بازرسی را به عنوان یک کاندید آزاد در نظر بگیریم. درک ما از نحوه کار وب ما را به توسعه روش های بهتری برای ماشین‌هایی که با ماشین‌های دیگر حرف می‌زنند, هدایت می‌کند. مفهوم Alistair Cockburn روی معماری شش ضلعی ما را از معماری‌های لایه‌لایه به جایی که منظق تجاری می‌تواند پنهان کند هدایت کرد. پلتفرم‌های مجازی سازی به ما اجازه فراهم کردن تغییر اندازه ماشین‌ها را در اندازه با توجه به زیر ساخت را می‌دهد که به ما راهی برای کنترل این ماشین‌ها در مقیاس می‌دهد. برخی از سازمان‌های بزرگ و موفق مانند آمازون و گوگل از دیدگاه تیم‌های کوچک که چرخه عمر کامل خدمات خود را دارند, حمایت می‌کنند. و اخیراٌ , نت فلیکس به ما روش های ساخت پاد شکننده را در مقیاسی که درک درست 10 سال پیس دشوار بود, به اشتراک گذاشته است.

طراحی مبتنی بر دامنه. تحویل داوم, مجازی سازی تقاضا , اتوماسیون اداری , تیم های مستقل در مقیاس کوچک , میکروسرویس‌ها از این دنیا ظهور کرده اند. آن‌ها قبل از این واقعیت اختراع یا توضیح داده نشده بودند: آن‌ها به عنوان یک گرایش, یا یک الگو از کاربرد دنیای واقعی ظاهر شدند. اما آن‌ها فقط به خاطر همه چیز‌هایی که قبلاٌ رفته‌اند وجود دارند. سراسر این کتاب, من از کار قبلی برای کمک به ترسیم تصویری از نحوه ساخت, مدیریت, و تکامل میکروسرویس‌ها استفاده خواهم کرد.

بسیاری از سازمان‌ها دریافته‌اند که با در آغوش گرفتن ریز‌دانه‌ها و ساختار میکروسرویس می‌توانند نرم‌افزار را سریع تر تحویل دهند و فن‌آوری‌های جدیدتر را بپذیرند, میکروسرویس‌ها به ما آزادی بیشتری می‌دهند تا واکنش نشان دهند و تصمیمات متفاوتی بگیرند, به ما این امکان را می‌دهد که سریع‌تر به تغییرات اجتناب‌ناپذیری که همه ما را تحت تاثیر قرار می‌دهند, واکنش نشان دهیم.

**میکروسرویس‌ها چی هستند؟**

میکروسرویس‌ها خدماتی کوچک و مستقل هستند که با هم‌کار می‌کنند. اجازه دهید با شکستن مسئله را کمی تجزیه و تحلیل کنیم و ویژگی‌هایی را که متفاوت هستند را توضیح دهیم.

**کوچک و تمرکز بر انجام یک چیز**

با نوشتن کد برای افزودن ویژگی جدید کد بیس رشد می‌کنند. در طول زمان, میتوان تشخیص داد که درکجا باید تغییر ایجاد شود چون بیس کد خیلی بزرگ است. با وجود انگیزه برای پاک سازی , کد بیس یکپارچه و ماژولار, همه این موارد در مرزهای فرآیند در حال فروپاشی هستند. کد مربوط به توابع مشابه شروع می‌شود تا همه جا پخش شود ایجاد اشکالات و پیاده‌سازی را سخت‌تر کند.

در یک سیستم یکپارچه, ما با این نیروها با تلاش برای اطمینان از یکپارچگی کد, اغلب با ایجاد انتزاع یا ماژول ها, با این نیرو‌ها مبارزه میکنیم. پیوستگی به منظور داشتن کد مرتبط با یکدیگر یک مفهوم مهم است که ما در مورد میکروسرویس‌ها فکر میکنیم. این توسط رابرت سی تقویت می‌شود. تعریف مارتین از اصل مسئولیت واحد که بیان می‌کند " چیز‌هایی را جمع کنید که به همان دلیل تغییر می کنند, و چیزهایی که به دلایل مختلف تغییر می‌کنند را از هم جدا کنید".

ما مرزهای خدمات خود را بر روی مرزهای تجازی متمرکز می‌کنیم و مشخص می‌کنیم که در کجای کد برای یک قطعه از قابلیت استفاده می‌شود. و با نگه داشتن این خدمات بر روی یک مرز مشخص, ما از وسوسه استفاده از آن برای رشد بیش از حد بزرگ و با تمام مشکلات مرتبط با آن اجتناب می‌کنیم.

سوالی که اغلب پرسیده می‌شود این است که *کوچک چقدر کوچک است؟* دادن تعدادی از خطوط کد مشکل‌ساز است, چرا که برخی از زبان‌ها نسبت به دیگران پر معنی هستند و در نتیجه بیشتر در خطوط کمتری از کد می‌توانند انجام دهند. ما همچنین باید این حقیقت را در نظر بگیریم که می‌توانیم وابستگی‌های چند‌گانه را به دست آوریم که خود شامل بسیاری از خطوط کد هستند. به علاوه, بخشی از حوزه شما ممکن است به طور قانونی پیچیده باشد, و نیاز به کد بیشتری دارد. جان ایوس در RealEstate.com.au در استرالیا a را به عنوان چیزی توصیف می‌کند که می‌تواند در دوهفته بازنویسی شود, یک قاعده سر انگشتی که مفهوم خاصی خودش را درک می‌کند.

یک پاسخ نسبتاٌ بی اهمیت دیگر که میتوانم به آن بدهم, به اندازه کافی کوچک است و نه کوچکتر. وقتی در کنفرانس‌ها صحبت می‌کنید. من تقریباٌ همیشه سوال می‌کنم که چه کسی یک سیستم دارد که خیلی بزرگ است و میخواهید آن را بشکنید؟ تقریباٌ همه دست‌هایشان را بالا می‌برند. به نظر می‌رسد که یک حس خیلی خوب از چیزی که خیلی بزرگ است داشته باشیم. و بنابراین می‌توان اینگونه استدلال کرد که زمانی که یک قطعه کد دیگر خیلی بزرگ نیست, به اندازه کافی کوچک است.