

شهباز توکلی، مدیر توسعه زیرساخت و محصول چارگون جزییات نسل جدید دیدگاه را تشریح کرد: تغییر معماری «دیدگاه» دوره گذار ما به آینده‌ای مترقی است

سرعت رشد تکنولوژی و همزمان با آن تغییر نیازهای کاربران و تمرکز بر سرعت، دقت و سهولت دسترسی موجب شد تا شرکت چارگون، تولید نسل جدید محصولات «دیدگاه» که از چند سال پیش در حال ایجاد زمینه و بستر آن بود را اجرایی کند. این تغییرات که با عنوان نسل جدید دیدگاه 5 کلید خورده و در برخی محصولات از جمله مکاتبات، BPMS، جلسات، کارها و ESS یا پیشخوان کارکنان، اجرایی شده حاصل فعالیت مستمر واحد توسعه زیرساخت و محصول چارگون است. روابط عمومی چارگون درباره فرآیند تغییر معماری دیدگاه و مهاجرت به نسل 5 آن با شهباز توکلی-مدیر توسعه زیرساخت و محصول مصاحبه‌ای کرده است که مشروح آن را در زیر می‌خوانید:

در تغییر ساختاری چارگون واحد مدیریت محصول به واحد توسعه زیرساخت و محصول تغییر نام پیدا کرد. این تغییر ساختاری چه پیامدهایی را در واحد محصول به دنبال داشت؟

هدف از تغییر ساختار و یکپارچگی راهکارها در چارگون این بود که روی بهبود ساختار دیدگاه، تمرکز کنیم تا بتواند خودش را مطابق با تکنولوژی و وضعیت جاری در ترند تکنولوژی دنیا، هماهنگ کند. در حال حاضر تمرکز ما بر این است که زیرساخت و framework را جلو ببریم و کاری با محتوای محصولاتی که روی این زیرساخت آماده می‌شوند، نداشته باشیم.

تیم‌های تولیدکننده محصول اکنون به عنوان مشتریان تیمی که من مسئولش هستم، مطرح‌اند. خیلی از افراد که با من کار می‌کردند الان به خواهان محصول ما تبدیل شده‌اند و مجموعه‌ای از امکانات روی زیرساخت «دیدگاه» را از ما می‌خواهند که آنها را تولید و ارائه می‌کنیم. از طرفی تیمی که در حال حاضر با آنها کار می‌کنم از لحاظ سطح توانمندی (development) در نقطه‌ای هستند که می‌توانند پاسخگو و هدایت‌کننده بسیاری از تیم‌ها باشند؛ به معنی دیگر، پاسخ سوالات فنی بسیاری از تیم‌ها از فرآیند فروش گرفته تا مشتریان، نزد تیم زیرساخت و توسعه محصول است.

از ایده تغییر معماری دیدگاه بگویید؟

ما نزدیک به 18 سال است که با معماری مشخصی بیش از 35 نرم‌افزار را به تدریج تولید کردیم و توسعه دادیم، معماری مورد بحث بسیار ساده و در عین حال کارآمد طراحی شده بود اما در عالم تولید نرم‌افزار 18 سال زمان بسیار زیادی است.

هرچند که این معماری در بخش‌های مختلف در این 18 سال بارها و بارها بصورت جزئی مورد بازنگری و تغییر قرار گرفته بود اما از مدتی پیش به این جمع‌بندی رسیده بودیم که زمان بازنگری معماری در سطحی بسیار وسیع‌تر فرارسیده است. ما می‌دانستیم که معماری قدیمی «دیدگاه» جوابگوی نیازهای ما تا سال 2020 نخواهد بود؛ بنابراین تصمیم گرفتیم که معماری نرم‌افزار را بنحوی بازنگری کنیم که نیازهای استفاده‌کنندگان دیدگاه را برای آن سال و سال‌های بعد از آن پاسخگو باشد. برای این کار پروژه طراحی معماری جدید دیدگاه را به عنوان بزرگترین پروژه سال‌های اخیر چارگون تعریف کردیم و در این پروژه از نحوه ارتباط دیدگاه با دیتابیس، نحوه ارتباط بین لایه‌های مختلف نرم‌افزار، استانداردهای گرافیکی، استانداردهای کاربرپذیری، فریم‌ورک رابط کاربری و نحوه ارتباط بین نرم‌افزارها همه و همه را دوباره از نو طراحی کردیم و در این معماری سعی کردیم همه چیز را به بهترین شکل ممکن و با توجه به استطاعت‌مان بازنگری کنیم. نتیجه این پروژه، معماری جدیدی است که نرم‌افزارهای دیدگاه به مرور خود را با آن، انطباق می‌دهند. در حال حاضر پروژه بازطراحی معماری دیدگاه در آخرین گام‌های خود است و نرم‌افزارهای مختلف راهکارهای دیدگاه در حال انتقال به معماری جدید هستند.

ویژگی‌های مهم معماری جدید دیدگاه چیست؟

در یک کلام مهمترین ویژگی، انطباق با استانداردها و بهترین روش‌های روز دنیا است. در معماری جدید دیدگاه، دیدگاه امن‌تر، سریع‌تر و کاربرپسندتر از هر زمان دیگری است، نرم‌افزارها بار کمتری به سرور تحمیل می‌کنند و سبک‌تر می‌شوند در نتیجه، نرم‌افزار مقیاس‌پذیرتر از قبل می‌شود و می‌توانیم پاسخگوی مشتریانی در ابعاد بزرگتر

باشیم؛ ضمن اینکه رابط کاربری را کاملا بازنگری کرده و آن را حتی در سطح کوچک‌ترین مولفه‌ها، براساس استانداردهای جدیدی که در دنیا معرفی شده‌اند، طراحی کرده‌ایم.

مهمترین محور آن در رابط کاربری‌های روز دنیا چیست؟

5 تا 6 سال پیش وقتی دیدگاه 5 را شروع کردیم، استانداردهای طراحی با امروز بسیار متفاوت بود. ما در زمان بدی و در روزهای پایانی عمر یک استاندارد، کار خود را شروع کردیم؛ یعنی قبل از اینکه نرم‌افزارهای ما به این رابط کاربری منتقل شوند این رابط، منسوخ شده بود. در حال حاضر خود را با متریال دیزاین گوگل، انطباق می‌دهیم که رابط کاربری بسیار سبک، مینی‌مالیستی و زیباییست که کاربران از کار با آن لذت می‌برند.

فرآیند تغییر معماری و جایگزینی تکنولوژی‌های مورد استفاده در آن چگونه عملیاتی شدند؟

همانطور که گفتم یک بخش بزرگ کار، بازبینی معماری نرم‌افزار بود. در این بازبینی، بحث‌های جدید زیادی مطرح شد؛ مثلاً اینکه تولید رابط کاربری را از سمت سرور به کلاینت انتقال دهیم، روش صحبت کردن با دیتابیس‌مان با توجه به حجم دیتا و حجم transaction موجود، با چه روشی باشد که با وجود انجام تغییرات و تولید امکانات جدید روی کدها، نرم‌افزار هم، بار بیش از اندازه به دیتابیس تحمیل نکند.

به عنوان مثال در لایه ارتباط با دیتابیس ORM‌های مشهوری در دنیا وجود دارند مثل Entity Framework، NHibernate، که بسیار کارآمد هستند و امکانات بسیار زیادی را در اختیار برنامه‌نویسان قرار می‌دهند استفاده از آن‌ها هزینه تولید را بسیار پایین می‌آورند ولی مناسب صورت مسئله‌ای با ابعاد آنچه ما می‌خواستیم، نبودند و در واقع به سرورهایی که قرار است میزبان نرم‌افزارهایی که با این ORM ها تولید شده‌اند، بار زیادی تحمیل می‌کنند و در واقع هزینه تامین سخت‌افزار مشتریان را تا حد بسیار زیادی بالا می‌برند. ما به جای استفاده از چنین ORM‌ی، از یک Micro ORM به نام Dapper استفاده کردیم که هرچند به علت کمی توانایی‌ها هزینه زیادی برای تولید نرم‌افزار ایجاد می‌کند اما سبک‌ترین و سریعترین ORM دنیاست که شاخصه اصلی نه امکانات که Performance بسیار بالای آن است.

ما هزینه زیادی را متحمل شدیم برای اینکه بدیهی‌ترین امکاناتی که می‌توان از یک ORM توقع داشت را خودمان تولید کنیم.

مثال دیگر از تغییرات بزرگی که در این معماری اتفاق افتاده حرکت از ASP.net webform نه به ASP.net MVC که به ASP.net Web API است در واقع در خلال این تغییر ما تکنولوژی را انتخاب کردیم که پیچیدگی‌های زیادی را برای ما به همراه دارد؛ اما استفاده‌کننده نهایی را به طرق مختلف راضی نگه می‌دارد. مزیت بزرگ این تکنولوژی این است که سرور به عنوان سرویس‌دهنده، چیزی به نام رابط کاربری را آماده و ارائه نمی‌کند و فقط داده‌هایی که رابط کاربری بلد است با آن کار کند را در اختیار قرار می‌دهد. این معماری نیازمند رابط کاربری بسیار باهوشی است، و به این منظور یک Framework جدیدی برای رابط کاربری طراحی کردیم که با

داده‌ها مثل اپلیکیشن‌های موبایلی کار می‌کند. یعنی به جای آنکه منتظر باشد یک رابط کاربری آماده شده را از سرور تحویل بگیرد، صرفاً منتظر داده می‌ماند و خودش بلد است با داده چه کند و چگونه رابط کاربری را بسازد.

چنین رابط کاربری در صورتیکه به درستی طراحی نشود استعداد این را دارد که سنگین شود و کارکرد کلاینت‌ها را دچار مشکل کند. برای حل این مشکل ما کتابخانه‌های مختلفی را مورد ارزیابی قرار دادیم از کتابخانه مورد استفاده در دیدگاه 5 به نام Knock-out شروع کردیم و کتابخانه‌های دیگری مثل AngularJS 1&2، حتی preview نسخه 3 و 4 آن، یا کتابخانه مشهور دیگری به نام vue را بررسی کردیم و در نهایت یک library جاوا اسکریپتی را تحت عنوان React را انتخاب کردیم که در بین کتابخانه‌های رابط کاربری جزو بهترین‌ها و سبک‌ترین‌هاست.

حال با استفاده از این کتابخانه، با اینکه بار آماده‌سازی رابط کاربری را از سمت سرور حذف می‌کنیم و به سمت کلاینت می‌بریم؛ اما انتظار داریم که بار زیادی هم به کلاینت‌ها تحمیل نکنیم و مرورگرها بتوانند صفحاتشان را بدون مشکل آماده ارائه کنند. به نظر هرچند از این گفتگو این طور به نظر می‌رسد که این‌ها نقاط شاخص کار ما در تغییر معماری «دیدگاه» است؛ اما در واقع به اعتقاد من نقطه شاخص تفکری است که منجر به این انتخاب‌ها و این تغییرات شده است.

با وجود تغییر معماری و تلاشی که برای استفاده از

پیشرفته‌ترین تکنولوژی‌ها شده، می‌توان ادعا کرد که یک محصول متری وارد بازار کرده‌ایم؟

اصولا در کامپیوتر عمر تکنولوژی‌ها کوتاه است و این یک معنی خطرناک در پس خود دارد. آن هم این است که وقتی تکنولوژی را به خدمت می‌گیرید، اگر در مدت زمان کوتاهی بتوانید آن را به طور گسترده در نرم‌افزارهای خود به کار بگیرید می‌توانید ادعا کنید یک نرم‌افزار متری دارید؛ اما اگر مدت زمان مهاجرت آن طولانی شود، روی مجموعه‌ای کار می‌کنید که در زمان شروع کار متری بود ولی در انتهای آن، متری به حساب نمی‌آید. ما در چارگون با توجه به تجربیاتمان در تغییر معماری دیدگاه 4 و 5، لایه‌های مختلف نرم‌افزار را به شدت از هم مجزا و بی‌خبر نگه داشته‌ایم؛ در نتیجه با جرات می‌شود گفت، هزینه پرداختی برای منتقل شدن به تکنولوژی احتمالی بعدی نسبت به هزینه‌ای که تیم‌ها برای انتقال به معماری جدید متحمل می‌شوند خیلی کمتر است و از این جهت، تغییر معماری جدید «دیدگاه» برای ما دوره گذار به آینده‌ای متری است.

گفتید تکنولوژی‌های قبلی دیدگاه 5-6 سال عمر کرد و اکنون در حال خارج شدن از دور است. سرعت شیف‌ت کردن به معماری جدید باید چه قدر باشد تا تکنولوژی را

از دست ندهد؟

عمر تکنولوژی در دنیا کمتر از 3 سال است و مرتبا کوتاهتر می شود؛ یعنی اینکه باید در مدت کوتاهی این کار را انجام دهیم. ما باید در چنین مدتی و البته تا حد ممکن کوتاهتر از آن، نرم افزارمان را با تمام امکاناتش به مشتریان ارائه شده داشته باشیم. البته باید مجددا تاکید کنم که تغییرات صورت گرفته در معماری دیدگاه و تفکیک لایه ها به شکل جدید و بازطراحی نحوه ارتباط بین لایه های مختلف نرم افزار کمک می کند استفاده از تغییرات جدید در این معماری در بازه های زمانی بسیار کوتاهی امکان پذیر باشد؛ بنابراین از این به بعد تغییرات را با معماری جدید منطبق می کنیم و از آنجا که این بازه زمانی خیلی کوتاه خواهد بود؛ می توانیم همواره روی خط تکنولوژی باقی بمانیم.