

به نام خدا



دانشکده مهندسی کامپیوتر پروژه درس متدولوژی های ایجاد نرم افزار نیمسال اول ۰۳-۰۴

مهندسی متدولوژی برای شرکت «سیستم سازان ره آفرین»



فرض کنید شما به عنوان مهندس متدولوژی به یک شرکت فعال در حوزه ایجاد نرم افزار دعوت شده‌اید. دغدغه‌ها و مشکلاتی که در این جلسه مطرح شده، در ادامه شرح داده شده است. از شما درخواست می‌شود تا یک متدولوژی را مناسب با موقعیت توصیف شده برای این شرکت مهندسی کنید. راهنمایی‌های لازم در فایل پیوست ارائه شده است.

(۱) ما یک شرکت نرم‌افزاری با بیش از ۱۵ سال سابقه فعالیت در صنعت نرم‌افزار هستیم که به صورت تخصصی به تولید نرم‌افزارهای سفارشی برای مشتریان مختلف می‌پردازیم. مشتریان ما شامل طیف گسترده‌ای از استارت‌آپ‌ها، شرکت‌های خصوصی و سازمان‌های دولتی هستند. ساختار سازمانی ما شامل چهار واحد اداری و مالی، واحد کسب و کار، واحد کنترل کیفیت و واحد ایجاد و نگهداری نرم‌افزار است که واحد ایجاد و نگهداری نرم‌افزار بزرگ‌ترین بخش شرکت است.

(۲) در حال حاضر شرکت ما ۱۲۰ نفر عضو دارد که از این تعداد، ۸۰ نفر در واحد ایجاد و نگهداری نرم‌افزار فعالیت می‌کنند. این واحد همزمان بر روی شش پروژه کار می‌کند. سه پروژه در مرحله ساخت و سه پروژه دیگر در مرحله نگهداری هستند. تیم‌های تحلیل و تست نیز به صورت موردی و بر اساس نیاز هر پروژه تشکیل می‌شوند. تیم‌های ایجاد نرم‌افزار از ۱۰ تا ۱۵ نفر تشکیل شده‌اند و هر تیم یک مالک محصول^۱ و یک استاد چابک^۲ نیز دارد. تیم نگهداری نرم‌افزار ما که مسئولیت نگهداری سه پروژه را بر عهده دارد، شامل پنج عضو است و با توجه به شرایط پروژه‌ها فعالیت‌های خود را مدیریت می‌کند.

(۳) ما در تیم‌های ایجاد نرم‌افزار از یک متدولوژی سفارشی استفاده می‌کنیم که دوره‌های کاری آن زمان‌بندی مشخص و ثابتی ندارد و در ابتدای هر دوره مجدداً تنظیم می‌شود. علاوه بر تخمین، اولویت بندی و تخصیص تکالیف، زمان و جزئیات هر دوره کاری توسط مالک محصول و استاد چابک تعیین و به تیم ابلاغ می‌شود. این روش اگرچه انعطاف‌پذیری بالایی دارد، اما گاهی منجر به برگزاری جلسات طولانی و پراکنده می‌شود، به‌ویژه به دلیل آنکه مالک محصول در چند تیم حضور دارد و جلسات روزانه منظم کمتر برگزار می‌شود. در تیم‌های نگهداری، برخلاف تیم‌های ایجاد نرم‌افزار، متدولوژی خاصی تعریف نشده است.

اعضای این تیم وظایف خود را به صورت فردی و بدون اولویت‌بندی مشخص و به دلخواه انتخاب می‌کنند. (۴) در تعدادی از پروژه‌ها تحلیل و طراحی داریم که معمولاً برخی از اعضای تیم ایجاد این کار را انجام می‌دهند ولی معمولاً مدیران سطح بالاتر از نتیجه کار رضایت کامل ندارند چون متوجه برخی بخش‌های آن نمی‌شوند. تحلیل نرم‌افزار معمولاً زمان زیادی می‌برد چون تمام جوانب در آن در نظر گرفته می‌شود

¹ Product Owner

² Agile Master

و مزایا و معایب و انواع روش‌ها و احتمالات مشخص می‌شود و به تیم ایجاد اعلام می‌شود. تیم ایجاد از این کار راضی نیست چون تصمیم‌گیری برای آنها زمان‌بر می‌شود و پروژه را دچار تاخیر می‌کند. گاهی این تحلیل‌ها آنقدر طولانی و مفصل می‌شوند که تیم ایجاد به دلیل دیر رسیدن تحلیل و طراحی، مجبور می‌شود برای رسیدن به زمان‌بندی پروژه کیفیت را کاهش بدهد. گاهی هم زمان زیادی را صرف تصمیم‌گیری برای انتخاب راه‌حل مناسب پروژه‌ها می‌کند.

۵) ما معمولاً پروژه‌ها را با تست شروع می‌کنیم ولی به دلیل زمان‌بر بودن و رسیدن به موقع پروژه تست‌نویسی ادامه پیدا نمی‌کند. در نهایت و بعد از تکمیل پروژه شرکت یک تیم تست تشکیل می‌دهد و گاهی باگ‌های زیادی پیدا می‌شود. شرکت از این مورد راضی نیست چون هم زمان زیادی صرف رفع باگ می‌شود و هم تیم تست برای شرکت هزینه‌بر است.

۶) نگهداری یکی از پروژه‌های قدیمی شرکت تنها توسط یک نفر انجام می‌شود. چون کد زنی این پروژه کاملاً توسط خود این فرد انجام شده و فهم آن برای دیگران کمی دشوار است. در صورت خروج این فرد از شرکت مشخص نیست پروژه چه سرنوشتی خواهد داشت. این پروژه تست و مستندات کافی ندارد.

۷) برخی مدیران فنی در شرکت اعتقاد دارند چیزی را نباید از کد پاک کنیم چون ممکن است یک روز لازم شود. مثلاً در یکی از پروژه‌ها برای یک الگوریتم چند پیاده‌سازی آزمایشی انجام دادیم و در نهایت همه را در کد اصلی ادغام کردیم چون شاید روزی لازم شوند. پیمان‌هایی که منسوخ می‌شوند را هم چون ممکن است دوباره استفاده شوند از کد حذف نمی‌کنیم.

۸) واحد کسب‌وکار شرکت معمولاً جلسات با مشتریان را برگزار و در نهایت نتایج را به مالک محصول می‌دهد. گاهی این واحد به جای دیدگاه مشتری دیدگاه خود را از نیازمندی مشتری به مالک محصول منتقل می‌کند که بعد از انجام کار و تحویل به مشتری مشخص می‌شود مشتری نیازمندی دیگری داشته است.

۹) یکی از چالش‌های شرکت، تکنولوژی تولید محصولات است. یکی از مشتریان علاوه بر نسخه وب، نسخه موبایل و دسکتاپ نرم‌افزار را هم می‌خواست. مدیران تصمیم گرفتند به جای پیاده‌سازی جداگانه هر کدام با تکنولوژی خاص خود، همه را با کمک فریم‌ورک Flutter پیاده‌سازی کنند این کار هزینه‌های شرکت را کاهش داد چون همه نسخه‌ها تنها با یکبار کدنویسی پیاده‌سازی شدند و اعضای تیم‌ها هم نیاز نبود زبان برنامه‌نویسی جدیدی یاد بگیرند ولی مشتری از نتیجه کار راضی نبود.

۱۰) نیروی‌های قدیمی شرکت گاهی شرکت را دچار چالش می‌کنند. مثلاً مسئول زیرساخت و امنیت شرکت که تنها با پایگاه داده SQL آشنایی دارد در یکی از پروژه‌ها که به دلیل پیشنهاد تیم قرار شد پایگاه داده NOSQL استفاده شود با استفاده از آن مخالفت کرد و تیم مجبور شد از پایگاه داده SQL استفاده کند.

۱۱) زمانی شرکت ما برای پروژه‌ها معماری خاصی نداشت اما بعد از اجرای چندین پروژه از معماری موفق آن برای دیگر پروژه‌ها استفاده می‌کنیم که این کار باعث شده شرکت برای معماری جدید هزینه‌ای نکند. این معماری، بعد از جلسات متعدد بین همه اعضای تیم، به صورت جمعی مورد قبول واقع شده‌است. اما متأسفانه چندین پروژه هم بوده که معماری در پایان پاسخگوی نیاز به مقیاس‌پذیری نبوده و در نتیجه

به شرکت هزینه تحمیل شده است. در این موارد متاسفانه معمار سیستم می گوید من از ابتدا هم مخالف بودم و به نظر جمع احترام نمی گذارد.

۱۲) برای برخی پروژه‌های شرکت که برای سازمان‌های دولتی هستند در جلسات طراحی حتی مدیران شرکت هم حضور پیدا می کنند و نظرات خود را ارائه می دهند و تیم طراحی هم گاهی مجبور به قراردادن نظرات آنها در طراحی می شود. برنامه نویس‌ها معمولاً دوست ندارند عضو تیم‌های ایجاد این پروژه‌ها باشند. برای این پروژه‌ها معمولاً برنامه ریزی و زمان بندی دقیقی با جزئیات کامل مشخص می شود. گاهی پیش آمده تیم برای اینکه پروژه را سر موعد تحویل دهد بخشی از الگوریتم را پیاده سازی نکرده که در نهایت مشتری ناراضی شده است.

۱۳) در پروژه‌های شرکت چون تیم‌های مختلفی روی آنها کار کرده اند استایل‌ها و استانداردهای کد نویسی متفاوت دارند و معمولاً استفاده از پیمان‌های آنها در دیگر پروژه‌ها دشوار است. مثلاً در یکی از پروژه‌ها نیاز به پیمان‌پرداخت بود و این مورد در پروژه‌های قبلی وجود داشت ولی نتوانستیم از آن در پروژه جدید استفاده کنیم. قبلاً برای پروژه‌ای از یک پیمان‌بیرون شرکت استفاده شده بود که بعد از بروزرسانی آن کل پروژه دچار مشکل شد. برای همین بعد آن مدیران، تیم‌ها را ملزم کردند همه چیز را خودشان از ابتدا پیاده سازی کنند.

۱۴) در برخی پروژه‌ها به دلیل کمبود وقت ارزیابی کیفیت نداریم. مثلاً در یکی از پروژه‌ها مدیر فنی به معماری اطمینان کامل داشت برای همین معماری ارزیابی نشد اما زمانی که سیستم زیربار رفت مشخص شد معماری دارای ایراد است اگر فرایندی داشتیم که این موارد را ابتدای کار می‌سنجیدیم این مشکلات پیش نمی‌آمد و هزینه به شرکت تحمیل نمی‌شد.

مدیر شرکت: برای حل این مشکلات، از مشاوران مختلفی کمک گرفته‌ایم. با این حال، راه‌حلهایی که تاکنون ارائه شده‌اند، یا عملیاتی نبوده‌اند و یا کافی به نظر نرسیده‌اند. از شما درخواست داریم راه‌حل پیشنهادی خود را ارائه کنید. لطفاً برای هر یک از پیشنهادات خود، دلیل انتخاب آن را توضیح دهید و مشخص کنید که این راه‌حل برای رفع کدام مشکل پیشنهاد شده است. همچنین توضیح دهید که چگونه می‌توان اطمینان حاصل کرد که راه‌حل شما موثر، کافی و عملی است.