

رهنگاشتی برای مدیریت
راهبردی عرضه API



راهبری پروژه‌های عرضه API



شناسنامه سند

عنوان سند:

راهبری پروژه‌های عرضه‌ی API:
ره‌نگاشتی برای مدیریت راهبردی عرضه‌ی API

تهیه‌شده در: واحد توسعه کسب‌وکار پادپوم
تاریخ تولید سند: اردیبهشت‌ماه ۱۴۰۱

نویسنده: مهدی ملکی‌فر
ویراستار: شادی سجادی
تصویرساز: سحر جمالی

کلیه حقوق این اثر متعلق به پادپوم (گروه فناپ) است.
نقل مطالب با ذکر منبع آزاد است.



راهاندازی، توسعه و هدایت یک کسب‌وکار نیازمند ارتباط با زنجیره معمولاً بزرگی از کسب‌وکارها به منظور تأمین خدمات پایه‌ای، زیرساختی و پشتیبانی است. بسیاری از شرکت‌ها و نهادها امروزه خدمات نرم‌افزاری آنلاین خود را در قالب API به کسب‌وکارهای دیگر ارائه می‌دهند: روشی چابک، امن، آسان و مقیاس‌پذیر.

«پادیوم» یک پلتفرم دوسویه است. بدین ترتیب که متقاضی می‌تواند علاوه بر دریافت خدمات از پادیوم، APIها و خدمات خود را در آن عرضه کند. محیط پادیوم، محیطی شفاف و رقابتی است و کاربران می‌توانند خدمات تأمین‌کنندگان مختلف را از نظر قیمت، عملکرد و... مقایسه کرده و از میان آن‌ها انتخاب کنند.

تلاش ما در پادیوم این است که توسعه‌دهندگان برای ایجاد یک بستر نرم‌افزاری جدید، مجبور به توسعه تمام اجزای آن نباشند و برای دریافت خدمات دیجیتال، نیازمند مذاکره با ده‌ها نهاد و سازمان نشوند. ما تکه‌های پازل را در محیطی چابک و شفاف گرد هم آورده‌ایم تا افراد بتوانند با چیدن این تکه‌ها، نرم‌افزار مطلوب خود را در کوتاه‌ترین زمان توسعه دهند.

پادیوم در سال ۱۳۹۸ در گروه فناپ و با حمایت بانک پاسارگاد متولد شد و آغاز به کار کرد. امروز، خدمات متنوعی از تأمین‌کنندگان مختلف در پادیوم در دسترس هستند و به مرور، با همکاری سایر بازیگران زیست‌بوم فناوری اطلاعات کشور، این خدمات گسترده‌تر و کامل‌تر خواهند شد. هر کاربر می‌تواند با مشاهده قیمت‌ها، عملکرد و سطح خدمات، تأمین‌کننده مناسب خود را انتخاب کند و به سرعت از خدمات پادیوم بهره‌برداری کند.

پادیوم به عنوان بخشی از مأموریت خود در حوزه توسعه اکوسیستم و گفتمان‌سازی، می‌کوشد تا تجربیات خود را در قالبی عملیاتی به مدیران و سیاست‌گذاران این حوزه منتقل کند.

powered by

pod



قناب



بانک پاسارگاد

درآمد

پیوند تحول دیجیتال در سازمان‌ها با مفهوم API، پیوندی ناگسستنی است و تمام سازمان‌هایی که در این مسیر گام بر می‌دارند، کم یا بیش، دیر یا زود، باید با API مواجه شوند.

سازمان‌ها بسته به ابعاد خود، ممکن است یک بهره‌بردار حداقلی از API باشند، یک عرضه‌کننده‌ی API باشند، یا در گامی فراتر، مالک پلتفرم شوند. این مسیر، بسته به آنچه سازمان در نظر دارد، می‌تواند در یک برنامه‌ی چند هفته‌ای خلاصه شود، یا ستونی برای تحول دیجیتال سازمان باشد.

گروه مالی پاسارگاد و به ویژه گروه شرکت‌های فناپ، سال‌هاست با API و تحول دیجیتال مأنوس هستند و پروژه‌های متعددی را برای خود و سایر نهادها به انجام رسانده‌اند و یکی از نقاط درخشان این تجربه، پلتفرم «پادیوم» است که نقش پنجره‌ی واحد عرضه‌ی API را در سرزمین هوشمند پاد و بانک پاسارگاد عهده‌دار شده است.

ما در پاد و پادیوم، همه چیز را از صفر آغاز کردیم. بارها نوشتیم و خط زدیم، بارها ساختیم و فرو ریختیم تا به آنچه در تناسب با رویای سرزمین هوشمند پاد بود، دست یابیم.

در این مسیر، تجربیات متعددی شکل گرفت که می‌تواند برای کسب‌وکارهای مشابه، کارگشا باشد. از این رو کوشیده‌ایم تا این مسیر را در قالب یک «هندبوک» تدوین کنیم تا مدیران بتوانند با تکیه بر این روش‌شناسی، مسیر راهبردی خود را در حوزه‌ی API ترسیم کنند. امید است که مقبول افتد.

مهدی ملکی‌فر

مدیر توسعه‌ی کسب‌وکار پادیوم

فهرست

API: چیستی / ۶
API: چسب کسب و کارها در زنجیره‌ی ارزش دیجیتال / ۸
API: کلید رویکرد «باز» / ۱۰
جایگاه API در مدل کسب و کار / ۱۱
سفر API سازمان شما / ۱۲
ارزیابی درونی / ۱۴
ارزیابی بازار / ۱۶
گزینش راهبرد / ۱۹
طراحی سید API ها / ۲۲
راهبرد قیمت گذاری سرویس / ۲۴
ساختار و منطق قیمت API / ۲۶
از کجا شروع کنیم؟ / ۲۷
راه بی بازگشت / ۲۸
به دنیای پلتفرمی خوش آمدید / ۲۹
رویکرد چابک در توسعه API / ۳۰
شن بازی با API ها / ۳۳
کلام آخر / ۳۴

API: چیستی

مراحل «اجرای» یک API به این شرح هستند:

۱. فراخوانی: ارسال درخواست با قالب مشخص از سوی سرویس گیرنده به سرویس دهنده

۲. عملیات (اجرای درخواست): اجرای درخواست توسط سرویس دهنده (هر نوع خدمتی، از استعلام و اعتبارسنجی گرفته تا صدور بیمه و کارت)

۳. پاسخ: بازگشت پاسخ از سرویس دهنده به سرویس گیرنده (گزارش وضعیت عملیات یا پاسخ استعلام یا...)

همان‌طور که در شکل زیر می‌بینید، استاندارد حاکم بر API، مربوط به ذات API است. به عبارت دیگر، مهم نیست چه سامانه‌ای آن را فراخوانی کند و چه سامانه‌ای آن را پاسخ دهد. هر کدام از این سامانه‌ها می‌توانند با زبان‌های برنامه‌نویسی مختلفی کار کنند، روی پلتفرم‌های مجزا ایجاد شوند و به شیوه‌های مختلفی کار کنند؛ اما برای کارکردن API، کافی است فرآیند فراخوانی API مطابق با پروتکل انجام شود. بدین ترتیب، هر سامانه‌ای می‌تواند به هر سامانه‌ی دیگری از طریق API متصل شود.

API مخفف عبارت Application Programming Interface است که به صورت لفظی می‌توان آن را «رابط برنامه‌نویسی کاربردی» ترجمه کرد.

برای درک بهتر API، یک رابط کاربری عادی (UI) را در نظر بگیرید. در UI، ارتباط بین یک فرد و یک سامانه کامپیوتری برقرار می‌شود. اما در API، در دوسوی ارتباط، دو سامانه کامپیوتری قرار دارند.

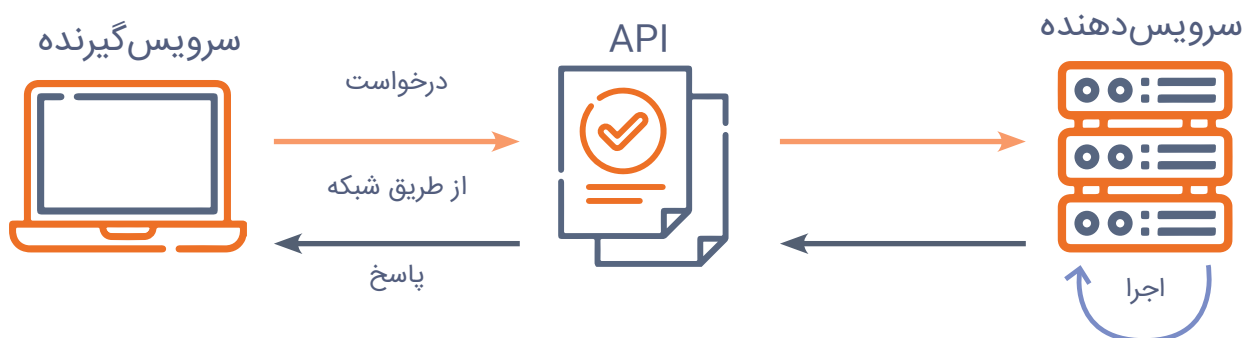
بنابراین می‌توان گفت API یک پل ارتباطی میان نرم‌افزارهای مختلف است. وجود این پل، بدین معناست که می‌توان بخش‌های مختلف نرم‌افزارها را به صورت مجزا از یکدیگر تعریف/ایجاد کرد و سپس یک پل ارتباطی میان آن‌ها برقرار کرد.

این پل، می‌تواند میان دو بخش از یک نرم‌افزار، دو نرم‌افزار داخلی یک سازمان، یا دو نرم‌افزار کاملاً مجزا، متعلق به اشخاص مجزا و در مکان‌های جغرافیایی مجزا باشد.

با این اوصاف، می‌توان به درستی گفت که API یک پل ارتباطی میان دو سامانه است؛ یک زبان ارتباطی با پروتکلی مشخص و استاندارد.

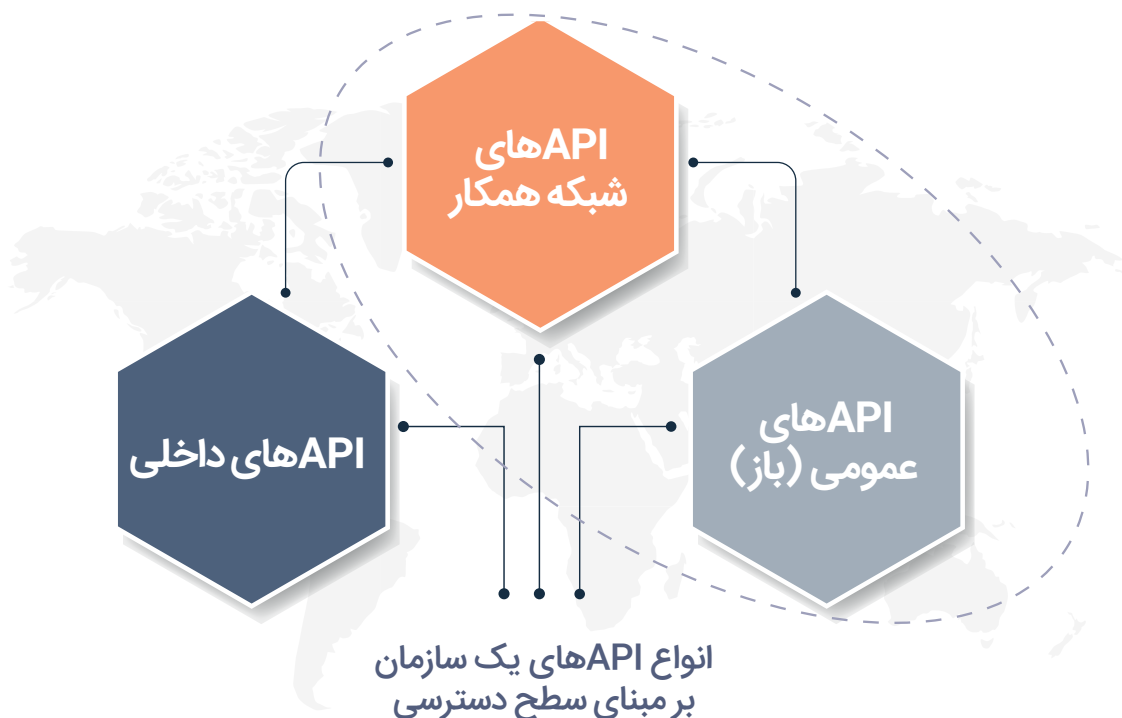
شکل ۱

سازوکار API: چگونه API می‌تواند دو سامانه را به یکدیگر متصل کند؟



شکل ۲

سطوح دسترسی API ها بر مبنای ماهیت آن‌ها تفاوت می‌کند.



سرور ارائه‌دهنده‌ی آن، یک رایانه‌ی شخصی عادی در سازمان باشد که درخواست‌ها را با تأخیری قابل‌توجه پاسخ می‌دهد. چنین زیرساختی، برای نخستین روزها شاید کافی باشد، اما با گسترش مقیاس بهره‌برداری، باید زیرساخت نیز به تناسب آن توسعه یابد.

بسته به ابعاد سازمان و ذی‌نفعان API، ممکن است نیاز به توسعه‌ی زیرساخت سخت‌افزاری و ارتباطی (شبکه) باشد. همچنین با اهمیت یافتن API در کسب‌وکارها، امنیت آن‌ها نیز اهمیت بیشتری می‌یابد. این روزها بسیاری سازمان‌ها تحت حملات بی‌وقفه‌ی سایبری قرار دارند و زیرساخت امنیت API، تبدیل به یک امر حیاتی برای عرضه‌ی API شده است.

از این رو است که سازمان‌ها معمولاً می‌کوشند ابتدا API های خود را در اختیار نهادهایی قرار دهند که اولاً، به آن‌ها اعتماد کامل دارند و ثانیاً، در تعامل با آن‌ها، می‌توانند نواقص سرویس‌های خود را به تدریج برطرف کنند.

همچنین برخی API ها بالذات نباید در اختیار همگان قرار گیرند. برای مثال، امروز سرویس‌های سازمان ثبت احوال ایران، از جمله سرویس‌های حساسی هستند که تنها در اختیار سازمان‌های خاصی قرار گرفته‌اند و عموم کسب‌وکارها نمی‌توانند از آن‌ها مستقیماً استفاده کنند.

به طور کلی API ها بر حسب نوع «استفاده‌کننده» بر سه نوع هستند:

۱. API های داخلی: این API ها میان سامانه‌های داخلی سازمان تعریف و استفاده می‌شوند.
۲. API های همکار: این API ها توسط شبکه همکاران معتمد سازمان (سازمان‌های همکار، مشتری و...) استفاده می‌شوند.
۳. API های آزاد: این API ها توسط تمام توسعه‌دهندگان قابل استفاده هستند.

سازمان‌ها در طول دوره‌ی بلوغ خود در استفاده از API ها معمولاً به ترتیب، روند فوق را طی می‌کنند؛ بدین معنا که در ابتدا می‌کوشند سامانه‌های خود را مبتنی بر API طراحی کنند، سپس API هایی را در اختیار شبکه‌ی همکاران خود قرار می‌دهند و سپس API های خود را به صورت عمومی منتشر می‌کنند.

دلیل این امر، این است که توسعه‌ی زیرساخت API، نیازمند تخصص، تجربه و زیرساخت فنی است. این نیازمندی‌ها، یک‌شبه توسعه نمی‌یابند. بلکه سازمان به مرور زمان با آن‌ها آشنا می‌شود.

برای مثال، در نخستین روزهای ایجاد یک API، ممکن است

API: چسب کسب‌وکارها در زنجیره‌ی ارزش دیجیتال

وقتی یک «پل» در اختیار داشته باشید، می‌توانید دو ساختمان را دو طرف رودخانه بنا کنید. در واقع، ابزار «پل ارتباطی» به شما اجازه می‌دهد تا به جدا کردن سامانه‌ها از یک‌دیگر به شیوه‌ای متفاوت فکر کنید.

برای بسیاری از کارکردهای نرم‌افزاری می‌توانیم از API استفاده کنیم. در نتیجه، نرم‌افزار تبدیل به یک «پازل» مرکب از «بلوک‌های نرم‌افزاری» می‌شود، مثلاً:

۱. احراز هویت کاربر
۲. نقشه و مسیریابی
۳. پرداخت و کیف پول
۴. داده‌های هواشناسی
۵. صدور بیمه
۶. اعلام جرایم
۷. وضعیت راه‌ها و...

حال که با چستی API آشناییم، به این می‌پردازیم که API چه نقشی در سامانه‌های نرم‌افزاری و کسب‌وکاری دارد؟

بدون API [یا چیزی مشابه آن] تمام سامانه‌ها باید با یک‌دیگر به صورت یک‌جا و یک‌پارچه توسعه پیدا کنند. چرا که برای جدا کردن سامانه‌ها، باید یک «پل ارتباطی» میان آن‌ها وجود داشته باشد. API دقیقاً همان پل ارتباطی است که به ما کمک می‌کند تا سامانه‌های مختلف را به صورت جزیره‌های مستقل توسعه دهیم.

برخی از این سامانه‌های مستقل ممکن است به لحاظ تئوری امکان تجمع داشته باشند (مثلاً دو زیرسامانه متعلق به یک سازمان)؛ اما برخی از سامانه‌ها، اساساً امکان تجمع ندارند. برای مثال، هیچ کس نمی‌تواند برای اپلیکیشن خود یک «بانک» تأسیس کند.

API در اولین قدم، به ما کمک می‌کند تا نرم‌افزار خود را به بخش‌های مختلفی تقسیم کنیم. قرار نیست تمام بخش‌های یک نرم‌افزار را خودمان از نو ایجاد کنیم. به نوعی می‌توان گفت API یک بستر «برون‌سپاری» نرم‌افزار است.



- حق‌الزحمه‌ی پیک و راننده و... از طریق API انجام می‌شود.
۲. احراز هویت: به جای ایجاد یک سرور امن توسط خود مجموعه، تمام کاربران می‌توانند از طریق یک زیرساخت API محور به یک سرور احراز هویت مرکزی متصل شوند.
۳. بیمه‌ی بار و افراد: هر بسته‌ی ارسالی تنها با فراخوانی یک API و ارسال اطلاعات و ارزش بسته، بیمه می‌شود. افراد و پیک‌های مجموعه نیز در صورت نیاز می‌توانند بیمه‌ی حوادث روزانه شوند.
۴. نقشه: به منظور ذخیره‌سازی و ثبت موقعیت‌های مکانی مبدأ و مقصد، سرویس‌های نقشه به‌صورت API قابل استفاده هستند.
۵. دیتابیس: شرکت به جای ایجاد دیتابیس، می‌تواند از زیرساخت دیتابیس شرکت‌های دیگر با API استفاده کند.
۶. IoT: شرکت با اتصال به یک پلتفرم IoT، می‌تواند در هر لحظه با فراخوانی یک API از وضعیت اشیاء متصل (خودروها، بسته‌ها، انبارها، و...) مطلع شود.
۷. سفارش: شرکت می‌تواند دریافت و ثبت سفارش خود را به‌صورت API در اختیار مشتریان قرار دهد. توجه داشته باشید که این API یک تفاوت اساسی با سایر API ها دارد: چرا که اینجا، شرکت حمل‌ونقل در نقش تأمین‌کننده ظاهر می‌شود.

بدین ترتیب، می‌توان یک فضای اقتصادی را در نظر گرفت که در آن، ابزار انتقال ارزش، API باشد. بدین معنا که خدمات و داده‌ها میان کسب‌وکارها از طریق API ها تبادل می‌شوند و به نوعی، API ها ابزاری برای ارتباط میان کسب‌وکارها هستند. هر کسب‌وکار می‌تواند از طریق API خدمات خود را به دیگران عرضه کرده و از دیگران خدمت بگیرد.

این اشتراک داده‌ها و دادوستد خدمات، کسب‌وکارها را در اکوسیستمی یک‌پارچه گرد هم آورده و منافع آن‌ها را به یک‌دیگر پیوند می‌زند.

بدین ترتیب، API مفهومی متفاوت به زنجیره‌ی ارزش می‌بخشد. در این مفهوم، کسب‌وکارها به جای آن که کالا را دست به دست کنند، اطلاعات را از یک‌دیگر دریافت کرده و پس از پردازش در اختیار کسب‌وکارهای دیگر و مصرف‌کنندگان قرار می‌دهند. بستر «جابه‌جایی ارزش» در این زنجیره، همان API است.

برای مثال، در مورد یک سامانه‌ی مدیریت حمل‌ونقل کالا، مانند الوپیک یا پادرو، می‌توان چنین فضایی را در نظر گرفت.

یک سامانه‌ی مدیریت حمل‌ونقل را در نظر بگیرید. این API ها در این مجموعه قابل استفاده هستند:

۱. پرداخت: سامان‌دهی تمام پرداخت‌ها، شامل درگاه پرداخت مشتری، کیف پول مشتری، پرداخت حق‌الزحمه پرسنل،

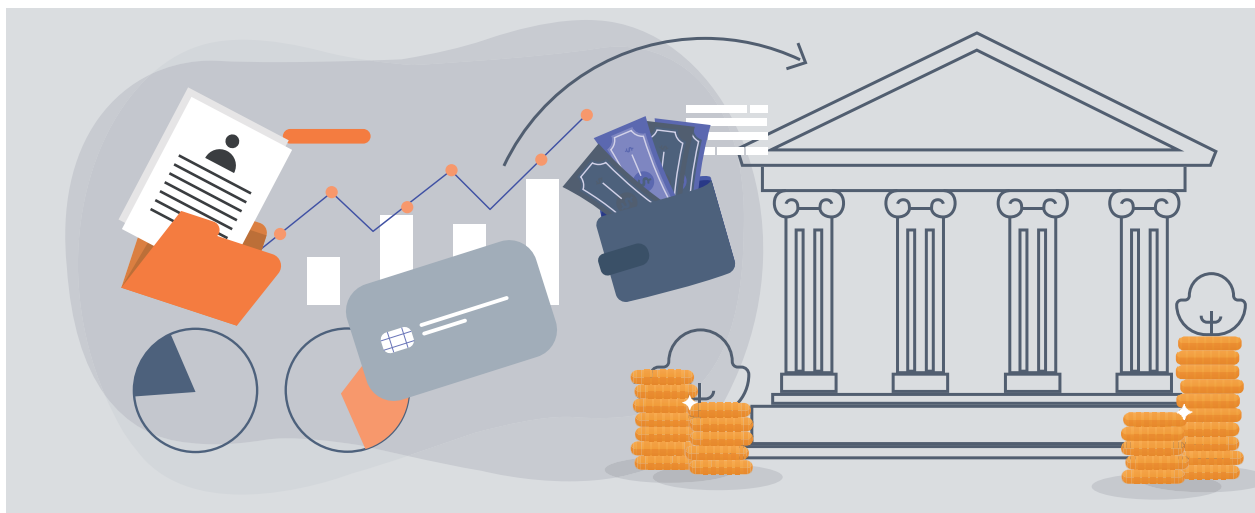
شکل ۳

نقشه‌ی ارتباطی یک سامانه‌ی مدیریت حمل‌ونقل با ذی‌نفعان خود از طریق API





API: کلید رویکرد «باز»



است که API در گفتمان «داده‌های باز» یا Open Data خلق می‌کند.

بدین ترتیب، می‌توان گفت API ها نقشی کلیدی در گفتمان «باز» در دولت، بانک‌داری و نوآوری دارند. برای مثال، تا چندی پیش، پلتفرم‌های تاکسی آنلاین برای تسویه‌های روزانه‌ی خود، حساب‌های تجمیع‌شده‌ی رانندگان را روی یک فایل اکسل به شعبه‌ی بانک تحویل می‌دادند. اما امروز این سرویس از طریق API تنها با یک کلیک و به‌صورت «آنی» صورت می‌گیرد.

تا چندی پیش، برای ثبت یک «بیمه» حتما باید کارشناس بیمه هویت فرد بیمه‌شونده را تایید می‌کرد و مشخصات بیمه‌ی مورد نیاز را در سامانه‌ی مرکزی بیمه ثبت می‌کرد. امروز، بسیاری از بیمه‌ها این قابلیت را پیدا کرده‌اند که از طریق API انجام شوند و تمام واسطه‌های «انسانی» را کنار بگذارند. مثلاً وقتی یک تکنیسین لوله‌کشی از طریق یک پلتفرم نظیر «آچاره» به محل مشتری اعزام می‌شود، با فشردن یک تیک ساده می‌تواند خود را به‌صورت روزانه، بیمه‌ی حوادث کند. (البته این امکان الزاماً وجود ندارد و صرفاً امکان‌پذیری آن مد نظر است).

بدین ترتیب، شاهد این هستیم که زیرساخت بانک، بیمه، ثبت احوال و... که زمانی زیرساخت‌های اختصاصی سازمانی و شرکتی بوده‌اند، به شیوه‌ای جدید در اختیار کسب و کارها قرار گرفته‌اند و هر کسب و کار می‌تواند خدمات آن‌ها را به گونه‌ای در سامانه‌های خود ادغام کند که گویی خود، مالک آن خدمت است!

نقش مهم دیگر API ها، بازشدن داده‌های سازمان‌ها، شرکت‌ها، بانک‌ها، نهادهای مالی، شهرداری‌ها و... است.

فرض کنید می‌خواهید بدانید EPS سهام یک شرکت بورسی چقدر است. برای این کار، دو سناریو پیش روی کسب و کار شماست:

سناریوی اول:

وارد پنل کاربری کارگزار خود شوید. نام نماد شرکت را جست‌وجو کنید، اطلاعات را مشاهده کنید و مقدار EPS را بخوانید. در این سناریو برای خواندن یک داده‌ی ساده، باید چند دقیقه زمان صرف کنید. در نتیجه، هرچند این داده‌ها ذاتاً «باز» و در اختیار شما هستند، اما به دلیل کندی دسترسی به این داده، شما نمی‌توانید استفاده درستی از آن داشته باشید.

سناریوی دوم:

وارد پنل کسب و کار خود شوید و با فراخوانی یک API که پیش‌تر در سامانه‌ی خود پیاده‌سازی کرده‌اید، تمام داده‌های نمادهای یک گروه، مثلاً کشاورزی را به پنل خود منتقل کرده و به سادگی جست‌وجو و تحلیل کنید. مثلاً یک ربات طراحی کنید که هر زمان EPS این سهم به مقدار مشخصی رسید، به شما هشدار دهد.

در سناریوی دوم، تمام فرآیندها به‌صورت «خودکار» انجام می‌شوند. در نتیجه شما می‌توانید از تمام ظرفیت داده‌ها استفاده کرده و بدون اتلاف وقت، از آن‌ها بهره‌مند شوید. این، تفاوتی

جایگاه API در مدل کسب‌وکار

۵. پیش‌گذار ارزش غنی‌تری ارائه می‌دهید، API خود می‌تواند یک ارزش جدید باشد، یا پیش‌گذار ارزش پیشین شما را تقویت کند؛

۶. تولید و مدیریت API یکی از فعالیت‌های کلیدی شما خواهد بود؛

۷. API تبدیل به بخشی از منابع کلیدی شما خواهد شد؛

۸. کسب‌وکارهای جدیدی به شبکه‌ی همکاران خود می‌افزایید؛

۹. از فروش API، مشتریان جدید و ارزش غنی‌تر خود، جریان‌های درآمدی جدید ایجاد می‌کنید؛

۱۰. هزینه‌های خرید و مدیریت API را می‌پردازید.

می‌توان گفت که API‌ها بسته به نوع و ماهیتشان، جایگاه‌های متفاوتی در مدل کسب‌وکار شما خواهند داشت و نقش‌های مختلفی را بر عهده می‌گیرند.

حال اگر در سطح بنگاه به جایگاه API‌ها بنگریم، می‌توان جایگاه‌های مختلفی را برای API در نظر گرفت. مدل استروالد می‌تواند چارچوب مناسبی برای بررسی کارکرد API در ارکان مختلف یک کسب‌وکار باشد.

به شکل زیر بنگرید. نقش‌های API در هر یک از ارکان مدل کسب‌وکار آورده شده است. API‌ها به عنوان یک ابزار «تحويل ارزش» می‌توانند در تمام ابعاد بوم کسب‌وکار شما نقش‌آفرینی کنند. این نقش‌آفرینی خود را به شیوه‌های مختلفی نشان می‌دهد:

۱. با خلق ارزش جدید، مشتریان جدیدی را هدف می‌گیرد؛

۲. ارائه‌ی خدمات به مشتریان خود را به‌صورت خودکار انجام می‌دهید؛

۳. خدمات خود را از کانال جدید API ارائه می‌دهید؛

۴. خدمات جدیدی خلق می‌کنید؛

شکل ۴

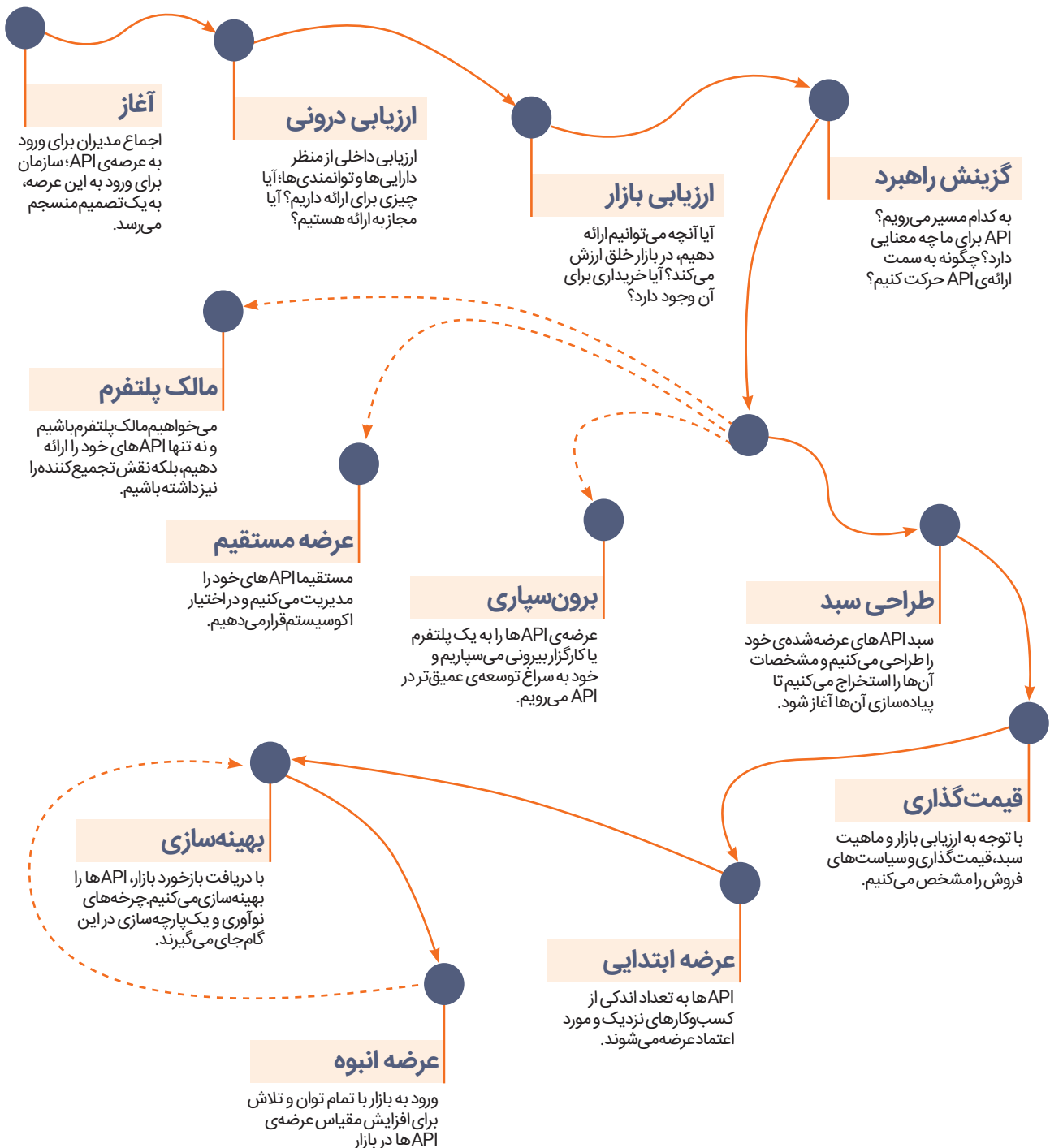
نقش‌هایی که API می‌تواند در مدل کسب‌وکار ایفا کند: نمایش روی بوم کسب‌وکار استروالد



سفر API سازمان شما

شکل ۵

سفر API سازمان شما، قدم به قدم: از صفر تا تجاری‌سازی. در امتداد این هندبوک، هر یک از این قدم‌ها را بررسی می‌کنیم.





موردکاوی: پادیوم

بی‌اغراق می‌توان گفت در ایران صنعت مالی و بانکی، از نظر بلوغ API، صنعت پیشرو محسوب می‌شود. این صنعت و ذی‌نفعان آن سال‌هاست که با مفهوم API آشنا هستند و عملیات خود را از این طریق به پیش می‌برند؛ به‌گونه‌ای که امروز، API یک جزء جدانشدنی از زنجیره‌ی ارزش مالی و بانکی ایران به شمار می‌رود. در جهان نیز کم و بیش چنین فضایی وجود دارد و عرضه‌ی API به عنوان بستر فنی بانک‌داری باز، در برخی کشورها به یک الزام تبدیل شده است. بانک پاسارگاد نیز سال‌هاست که سرویس‌های خود را در قالب API به مشتریان خود عرضه می‌کند. این سرویس‌ها توسط شرکت داتین (شرکت توسعه‌دهنده‌ی نرم‌افزار بانک‌داری بانک پاسارگاد) و خود مجموعه‌ی بانک پاسارگاد مدیریت و عرضه می‌شدند.

به موازات این امر، همگام با شکل‌گیری سرزمین هوشمند پاد، نیازمندی اساسی به یک «پنجره‌ی واحد عرضه‌ی API» در فناپ احساس شد؛ چرا که مهم‌ترین بستر اتصال کسب‌وکارها به زیرساخت سرزمین هوشمند پاد نیز API است و تقریباً تمام خدمات این پلتفرم، از احراز هویت گرفته تا حساب دیجیتال و حتی هوش مصنوعی، مبتنی بر API عرضه می‌شوند. بدین ترتیب پادیوم متولد شد. در نخستین روزها، پرسش‌های زیادی در زمینه‌ی ماهیت پادیوم مطرح بود:

آیا قرار است پادیوم سرویس‌هایی به جز سرویس‌های سرزمین هوشمند پاد را عرضه کند؟ آیا سرویس‌های بانک پاسارگاد، در قلمروی پادیوم جای می‌گیرند؟ نگاه پادیوم باید عرضه‌محور باشد یا تقاضا محور؟ هنگامی که یک گروه مالی بزرگ قصد ورود به چنین عرصه‌ای داشته باشد، ذی‌نفعان متعددی باید برای تصمیم‌گیری در مورد چنین خط کسب‌وکاری به اجماع برسند و تمام این پرسش‌ها باید با تکیه بر تحلیل و بینش کسب‌وکاری پاسخ داده شوند.

در نهایت، با بررسی نیاز بازار و توانمندی‌های گروه مالی پاسارگاد (نه فقط فناپ و نه فقط بانک پاسارگاد)، تصویر کنونی پلتفرم پادیوم شکل گرفت؛ چرا که می‌توانست نقش یک «فروشگاه همه‌کاره API» را برای مشتریان ایفا کند و همزمان، پیوندهای اکوسیستمی ارزنده‌ای برای گروه مالی پاسارگاد خلق کند.

هنگامی که صحبت از تحول دیجیتال سازمان باشد، باید سرفصلی برای مدیریت APIها در نظر گرفته شود. اینجاست که سازمان اساساً تصمیم می‌گیرد آیا می‌خواهد به دنیای APIها ورود کند؟ آیا قصد دارد تنها مصرف‌کننده‌ی API بماند یا برای ورود به دنیای API، گام جدی‌تری بر می‌دارد؟

پس از آن، سازمان باید یک ارزیابی از توانمندی‌های خود داشته باشد؛ اولاً، داده یا خدمتی برای عرضه در قالب API در سازمان وجود دارد؟ ثانیاً، توانمندی فنی تولید و عرضه‌ی API را در سازمان داریم؟ ثالثاً، به لحاظ قانونی و ملاحظات بالادستی، امکان عرضه‌ی این داده یا خدمت به‌صورت API وجود دارد؟

سپس باید به سراغ بازار رفت؛ آیا این داده یا خدمت، بازار ارزنده‌ای دارد؟ جواب این سوال ممکن است بسیار مبهم باشد، چرا که از یک دینامیک مرغ و تخم‌مرغی تبعیت می‌کند. تا وقتی یک API وجود ندارد، هیچ مدل کسب‌وکاری برای استفاده از آن شکل نگرفته و طبیعتاً مشتری بالفعلی ندارد. (مگر آنکه رقبا خدمات مشابهی را پیش‌تر در بازار عرضه کرده باشند). ارزیابی سازمان از بازار مشخص می‌کند که آیا اساساً باید APIها توسعه یابند؟

سپس سازمان مسیر خود برای ورود به عرصه‌ی API را تعیین می‌کند؛ آیا سازمان در نظر دارد که یک ورود حداکثری داشته و خود، مالک پلتفرم باشد؟ یا یک حضور حداقلی و عرضه‌ی مستقیم APIها برایش کفایت می‌کند؟ شاید گزینه‌ی بهتر، اعتماد به یک پلتفرم دیگر باشد.

سپس سازمان سبد API خود را طراحی می‌کند؛ چه APIهایی با چه مشخصاتی، با چه پروتکلی، با چه محدودیت‌ها و چه سیاست‌های قیمت‌گذاری قرار است به بازار عرضه شوند؟ چه کسانی مجاز به دریافت سرویس خواهند بود؟

در گام بعد، سازمان در فاز پایلوت، APIهای خود را عرضه می‌کند. در این قدم، بازخوردهای اولیه از بازار دریافت می‌شود و سبد API، طبق بازخورد بازار، بهینه‌سازی می‌شوند. هم‌چنین زیرساخت فنی در این گام افزایش مقیاس می‌یابد.

در گام نهایی، سازمان وارد سیکل عرضه‌ی انبوه می‌شود و در سیکل‌های نوآوری، می‌کوشد تا ضمن فروش انبوه سرویس، طیف جدیدی از خدمات را عرضه کند.

در ادامه، تمام این سفر را موشکافی می‌کنیم.



ارزیابی درونی

در ارزیابی درونی، سازمان سه موضوع را مورد بررسی قرار می‌دهد: داده‌ها و خدمات قابل عرضه، توانمندی فنی و الزامات فرادستی

توانمندی فنی و زیرساخت‌های سازمانی

پیش از هر چیز باید در نظر داشته باشید که برای عرضه‌ی API، سه بُعد از آمادگی فنی باید در سازمان شما وجود داشته باشد:

۱. **آمادگی منابع انسانی:** ارائه‌ی API نیازمند یک تیم خبره است که بتواند داده‌ها و خدمات موجود را به API تبدیل کرده و آن‌ها را مستندسازی کند. به طور معمول، حداقل یک تیم دونفره فنی بدین منظور نیاز است. همچنین در ادامه، این تیم باید پشتیبانی و بهینه‌سازی API‌ها را به عهده گیرند.

۲. **آمادگی نرم‌افزاری:** زیرساخت نرم‌افزاری سازمان شما باید آمادگی عرضه‌ی خدمت/داده به‌صورت API را داشته باشد. فرض کنید که تمام داده‌های استعلام‌پذیر شما در بستر یک نرم‌افزار پیش‌ساخته انبار می‌شوند که امکان اتصال به شبکه نداشته باشد. در این صورت، گام اول در مسیر ورود به API، فراهم‌سازی بستر نرم‌افزاری خواهد بود.

۳. **بستر سخت‌افزاری:** عرضه‌ی API نیازمند اتصال شبکه و سرور است. به هر ترتیب، یک توان رایانشی و زیرساخت شبکه‌ای استاندارد باید فراهم شود تا بتوانید API عرضه کنید. این زیرساخت بسته به سطح حساسیت، امنیت، تعداد مشتریان و تعداد فراخوانی سرویس‌ها، ممکن است پیچیدگی بیشتری پیدا کند.

توجه داشته باشید که این نیازمندی‌ها به‌صورت «حداقلی» مطرح می‌شوند. در سازمان‌های بزرگ‌تر، بخشنامه‌ها و فرایندها نیز بخش جدانشدنی از آمادگی سازمانی محسوب می‌شوند. در بسیاری موارد، این زیرساخت‌ها باید ذیل واحد خاصی در سازمان ایجاد شوند و امکان ایجاد آن‌ها به‌صورت اقتضایی وجود ندارد. از این رو، شما در اولین گام باید تمام توانمندی‌ها و زیرساخت‌های نرم و سخت مورد نیاز را در سازمان خود ایجاد کرده و همراهی مدیران ارشد سازمان را جلب کنید. در این مسیر، استفاده از محتوای تخصصی و آموزش مدیران، قطعاً اثربخش خواهد بود.

پیش از هر چیز، باید دانست که آیا سازمان بالذات می‌تواند API عرضه کند؟ بدین منظور، سه پرسش اساسی را باید پاسخ داد و جنبه‌های مختلف آن را بررسی کرد.

داده‌ها و خدمات قابل عرضه

بسیاری از خدمات و داده‌های موجود در سازمان، قابلیت عرضه یا ثبت درخواست به‌صورت آنلاین را دارند. به طور کلی، داده‌ها و خدمات قابل عرضه به‌صورت API را می‌توان به شیوه‌ی زیر دسته‌بندی کرد:

۱. **API‌های استعلامی:** این API‌ها عموماً کارکردی از جنس خوانش مقدار، استعلام وضعیت یا استعلام انطباق هستند. برای مثال، فرض کنید ورودی یک API استعلامی، کد ملی یک فرد باشد. در ازای ورود کد ملی، می‌توان مقادیری نظیر نام و نام خانوادگی فرد را دریافت کرد. همچنین می‌توان اعتبار فرد را استعلام کرد یا وضعیت حیات فرد را جویا شد. همچنین می‌توان انطباق کد ملی فرد را با یک کد ملی که پیش از این در محلی دیگر در سامانه ثبت شده، استعلام کرد.

۲. **خدمات مبتنی بر API:** در این دسته API‌ها، درخواست یک خدمت به‌صورت API ثبت می‌شود. مثلاً سرویس حمل بار را هم می‌توان از طریق رابط کاربری موبایل سفارش داد و هم می‌توان مستقیماً به‌صورت API فراخوانی کرد. در این شرایط، API ابزار خودکارسازی سفارش خدمت است. خدمات بیمه، حمل‌ونقل، ثبت درخواست‌های اداری و... از این دست هستند.

به طور کلی، می‌توان گفت در ازای فراخوانی یک API، یا خدمتی انجام می‌شود و یا داده‌ای در اختیار سرویس‌گیرنده قرار می‌گیرد. حال باید دید که کدام خدمات و داده‌ها در سازمان شما می‌توانند به‌صورت API ارائه شوند. یکی از کلیدهای مهم در این زمینه، مطالعه موارد مشابه است. شما می‌توانید به بسترهای فروش API مراجعه کنید و ببینید نهادهای مشابه شما، چه خدمات و داده‌هایی را به‌صورت API عرضه می‌کنند.

الزامات فرادستی

بسیاری از نهادها و سازمان‌های بزرگ، ذیل جهت‌گیری‌های راهبردی کلانی حرکت می‌کنند که تصمیم‌گیری در مورد آن‌ها الزاما در قلمروی سازمان نیست. این جهت‌گیری‌ها ممکن است سازمان را «ملزم» به ارائه‌ی API کنند یا بالعکس، سازمان را از عرضه‌ی برخی سرویس‌ها منع کنند.

برای مثال، داده‌های مربوط به وضعیت آلودگی هوا که توسط شهرداری‌های کلان‌شهرها جمع‌آوری می‌شود، به‌صورت ذاتی داده‌های «عمومی» محسوب می‌شوند و انتشار آن‌ها یک «الزام فرادستی» است. در مقابل، اطلاعات هویتی افراد که در پایگاه‌های داده‌ی ملی گردآوری شده‌اند، حساسیت شدیدی را طلب می‌کنند و انتشار آن‌ها در قالب API، تابع روال‌ها و رویه‌هایی است که به‌صورت سخت‌گیرانه توسط نهادهای بالادستی تنظیم می‌شوند.

در سطح سازمان‌های خصوصی، می‌توان به پروتکل‌هایی نظیر PSD2 اشاره کرد که بانک‌های اروپایی را از منظر رگولاتوری

ملزم به ارائه‌ی API در چارچوبی خاص می‌کند. اما در نگاهی کلان‌تر، سازمان‌های خصوصی عرضه‌کننده‌ی API، این الزام را از اسناد راهبردی خود دریافت می‌کنند، نه از نهادهای رگولاتوری.

سازمان‌های خصوصی می‌توانند در اسناد راهبردی خود، API را به عنوان یک کانال جدید، یک منبع درآمد جدید و یک ابزار انتقال ارزش جدید در نظر بگیرند و واحدهای عملیاتی را ملزم به عرضه‌ی API کنند. در چنین شرایطی، شما به عنوان راهبر یک واحد کسب‌وکاری، راهبرد عرضه‌ی API خود را با اسنادی نظیر نقشه‌ی راهبرد شرکت هم‌سو می‌کنید.

نتیجه‌گیری

در این گام، شما به سه سوال اساسی پاسخ می‌دهید:

۱. چه داده و خدمتی در سازمان وجود دارد که به‌صورت API قابلیت عرضه داشته باشد؟

۲. آیا بستر فنی و زیرساخت‌های سازمانی مورد نیاز برای عرضه‌ی API در سازمان وجود دارد؟

۳. چه خدماتی «باید» و چه خدماتی «نباید» عرضه شوند؟ آیا محدودیتی در بازار هدف وجود دارد؟ آیا عرضه‌ی API با راهبردهای سازمان هم‌سو است؟

با این مفروضات، با فهرستی از خدماتی که «امکان» عرضه‌ی آن‌ها را داشته باشیم، به گام بعد می‌رویم.





ارزیابی بازار

**ز چه اصلم ز چه فصلم به چه بازار خرندم؟
در گام قبل، به اصل و فصل پرداختیم و در این گام، بازار را ارزیابی می‌کنیم.**

به منظور ارزیابی بازار API شما نیازمند ارزیابی حداقل چند عامل هستید:

۱. تعداد و فرکانس فراخوانی

۲. تعداد مشتری

۳. تمایل مشتری به پرداخت

۴. قدرت چانه‌زنی

ارزیابی این عوامل یک ارزیابی حداقلی از آینده‌ی API‌های سازمان شما در بازار به دست می‌دهد. هر چند بدون شک عوامل اقتصادی دیگری در موفقیت فروش API‌ها دخیل هستند و این عوامل به تنهایی ضامن موفقیت نخواهند بود.

تعداد و فرکانس فراخوانی

API‌ها با خود معنای «اتوماسیون» را یدک می‌کشند و این اتوماسیون، هزینه دارد.

فرض کنید شما قصد دارید یک بار، داده‌های یک پزشک را در پنل سازمان نظام پزشکی استعلام کنید. انجام این کار به صورت دستی از شما ۳ دقیقه زمان می‌برد. در مقابل، فرض کنید قصد دارید داده‌های دو هزار پزشک را استعلام کنید. این کار به صورت دستی از شما ۶ هزار دقیقه (معادل ۱۰۰ ساعت) زمان می‌برد. پیاده‌سازی API‌های سازمان نظام پزشکی، به صورت فرضی حدود ۵ ساعت زمان نیاز دارد و پس از آن، استعلام به صورت خودکار انجام می‌شود.

مثال فوق، به خوبی به شما نشان می‌دهد که «فرکانس» دریافت خدمت، چطور می‌تواند به عرضه‌ی API معنا ببخشد. در واقع بسیاری از سرویس‌ها تمامی مشخصه‌های یک سرویس «خوش‌تعریف» را دارند، اما فرکانس فراخوانی آن‌ها به قدری پایین است که رویکرد API را از ارزش ساقط می‌کند.

تجربه نشان می‌دهد که داده‌های استاتیک معمولاً کم‌ترین مطلوبیت را در میان API‌ها دارند. برای مثال، سرویس

«بوستان‌های تهران بزرگ» را در نظر بگیرید. تهران بزرگ تعداد مشخصی بوستان دارد که کل دیتای آن‌ها در یک حرکت قابل دریافت است و حجم کل دیتابیس مربوط به آن‌ها با تمام مشخصات، شاید از یک مگابایت تجاوز نکند. از سوی دیگر، این داده‌ها کاملاً استاتیک هستند و حداکثر هر سال یک مرتبه به‌روزرسانی برای آن‌ها کفایت می‌کند. در این شرایط، سرویس‌گیرنده ترجیح می‌دهد در یک اقدام تمام دیتا را دریافت کند و آن را به صورت API پیاده‌سازی نکند.

تجربیات داخلی نشان‌گر آن است که سازمان‌های متعددی در ایران، در ارزیابی این عامل دچار چالش هستند. آن‌ها داده‌های استاتیکی که حجم محدودی دارند را در قالب API پیاده‌سازی و عرضه می‌کنند و سپس به دلیل عدم استقبال بازار از این API‌ها، نسبت به بازار API دلسرد می‌شوند. از سوی دیگر، هزینه‌های نگهداری و به‌روزرسانی این سرویس‌های بدون مشتری، موجب می‌شود تا سازمان این سرویس‌ها را رها کند و حتی مخاطب محدود این سرویس‌ها نیز نتواند آن‌ها را دریافت کند.

یک معیار بسیار مهم در قیمت‌گذاری

سرویس‌ها این است که هزینه‌ی

«خودکارسازی سرویس» از هزینه‌ی

استخدام نیروی انسانی تجاوز نکند. برای

مثال، یک سرویس انطباق تصویر اگر

ماهانه ۲۰ میلیون تومان هزینه در بر داشته

باشد، به سرعت با نیروی انسانی جایگزین

خواهد شد. شرکت می‌تواند همین کارکرد را

با یک کارشناس دورکار، با هزینه‌ی ماهانه ۶

میلیون تومان تأمین کند.



تعداد مشتری

عرضه‌ی عمومی API ها به طور ضمنی بدین معناست که مشتری کافی برای این سرویس در فضای بیرون وجود دارد. اما معمولاً این ارزیابی درست از آب در نمی‌آید. این موضوع، به این دلیل است که معمولاً سازمان‌ها «ارزش» خلق‌شده توسط API های خود را بیش از حد واقع تخمین می‌زنند و احساس می‌کنند که سرویس‌های آن‌ها «حتماً» مورد استقبال واقع می‌شود.

در ارزیابی تعداد مشتری، دو نگاه به طور موازی باید پیگیری شود:

۱. **نگاه معمول در ارزیابی بازار:** در این نگاه، تعداد کل مشتریان بالقوه با توجه به اعداد و ارقام، آمار، وضعیت بازار، نگاه خبرگی و... سنجیده می‌شود.

۲. **نگاه ارزش‌محور:** در این نگاه، مطلوبیت سرویس از جهت خلق ارزش برای مشتری سنجیده می‌شود. اینکه سرویس چگونه می‌تواند مدل کسب‌وکار یا عملیات مشتری را متحول کند و آیا مطلوبیت کافی برای پیاده‌سازی توسط مشتری را دارد یا خیر؟ به عبارت دیگر، آیا منافع سرویس برای مشتری بر هزینه‌های دریافت و پیاده‌سازی سرویس غلبه می‌کنند؟ این موضوع را در شکل ۶ مورد بررسی قرار داده‌ایم.

در واقع در نگاه سنتی ارزیابی بازار، حجم بازار بالقوه شناسایی می‌شود و در نگاه ارزش‌محور، به این پرسش پاسخ می‌دهیم که چه تعداد از مشتریان بالقوه را می‌توانیم به مشتری بالفعل تبدیل کنیم؟

تمایل مشتری به پرداخت

با فرض عبور از تمام معیارهای پیشین، در بسیاری موارد، مشتریان تمایل کافی به پرداخت هزینه برای سرویس ندارند. دلایل مشتریان برای عدم پرداخت هزینه برای سرویس عبارتند از:

۱. **عادت بازار:** هنوز در بسیاری از بازارها، فروش API تبدیل به یک مدل کسب‌وکار رایج نشده است. در برخی بازارها نیز به دلیل رقابت بسیار زیاد سرویس‌دهندگان، عرضه‌ی سرویس رایگان تبدیل به روتین شده است. بازار پرداخت ایران، یکی از بازارهایی است که از کارمزدهای بسیار پایین رنج می‌برد و کسب‌وکارهای پرداخت مجبورند از مدل‌های جایگزین (نظیر بهره‌ی رسوب) برای درآمدسازی استفاده کنند.

۲. **فرهنگ بازار:** فرهنگ غالب در بسیاری از کسب‌وکارها به سمت «تولید داخلی» می‌رود. راهبران استارت‌آپ‌های دیجیتال

معمولاً توانمندی فنی قابل‌توجهی دارند و دریافت سرویس را به منزله‌ی «ناتوانی در توسعه‌ی داخلی» می‌بینند. از این رو، می‌کوشند تا تمام سرویس‌های مورد نیاز خود را به‌صورت داخلی توسعه دهند و به سراغ تأمین‌کنندگان بیرونی نروند.

۳. **هزینه‌ی پایین منابع انسانی:** یک معیار بسیار مهم در قیمت‌گذاری سرویس‌ها این است که هزینه‌ی «خودکارسازی سرویس» از هزینه‌ی استخدام نیروی انسانی تجاوز نکند. برای مثال، یک سرویس انطباق تصویر اگر ماهیانه ۲۰ میلیون تومان هزینه در بر داشته باشد، به سرعت با نیروی انسانی جایگزین خواهد شد. شرکت می‌تواند همین کارکرد را با یک کارشناس دورکار، با هزینه‌ی ماهیانه ۶ میلیون تومان تأمین کند. در بازارهای توسعه‌یافته به دلیل هزینه‌ی بالای نیروی انسانی، اتوماسیون جذابیت بالاتری خواهد داشت؛ اتفاقی که هنوز در ایران نیافتاده است.

۴. **راهکارهای جایگزین:** امروز بسیاری از سرویس‌های زیرساختی با هزینه‌های پایین در دسترس هستند. در نتیجه، قیمت‌گذاری باید با نگاه رقابتی صورت گیرد تا راهکار سازمان شما، به سادگی با یک راهکار دیگر جایگزین نشود.

قدرت چانه‌زنی

برخی سرویس‌ها به‌صورت انحصاری در اختیار یک یا چند سازمان قرار دارند و مشتریان «مجبورند» که این سرویس را از همان سازمان(های) خاص دریافت کنند. هرچند در درازمدت ممکن است شکست انحصار رخ دهد، اما در بسیاری از موارد، انحصار به‌صورت قانونی یا بازاری وجود دارد. برای مثال، حوزه‌ی خدمات پرداخت تا چندی پیش به‌صورت قانونی در انحصار شرکت‌های PSP بود و از ظهور پدیده‌هایی نظیر پرداخت‌یاری و پرداخت‌سازی و پرداخت‌بانی و پذیرش آن‌ها توسط بانک مرکزی، مدت زیادی نمی‌گذرد. در مقابل، در بازارهایی نظیر حمل‌ونقل، تعداد عرضه‌کنندگان به‌صورت طبیعی محدود است و سرویس‌گیرندگان به هر صورت مجبور به دریافت سرویس از یکی از این شرکت‌ها هستند.

باید در نظر داشت که انحصار در سمت تأمین‌کننده، تنها یکی از عوامل قدرت چانه‌زنی است. در سوی دیگر، سرویس‌گیرنده نیز ممکن است قدرت چانه‌زنی بالایی داشته باشد و سرویس‌دهنده را مجبور به عرضه‌ی سرویس در قالب و شرایط خاصی کند. به طور کلی، بازار API از نظر قدرت چانه‌زنی بازار مقارنی نیست و این امر باید در ارزیابی بازار لحاظ شود.



شکل ۶

مطلوبیت دریافت سرویس؛ عوامل موثر بر تصمیم سرویس‌گیرنده

وابستگی عملیاتی

پیاده‌سازی API بدین معناست که سرویس‌گیرنده، بخشی از عملیات خود را مبتنی بر سرویس بیرونی بنا می‌کند. این وابستگی عملیاتی، سطحی از ریسک را به کسب‌وکار در پی دارد که در بسیاری از مواقع، موجب می‌شود تا کسب‌وکارها توسعه‌ی داخلی سرویس را ترجیح دهند. برند و سابقه‌ی عملکرد سرویس‌دهنده، حساسیت سرویس، قوانین و OLA/SLAها در این زمینه تعیین‌کننده هستند.

روال دریافت سرویس

فروش سرویس API در بسیاری از سازمان‌ها یک اتفاق جدید است که فرآیندها و زیرساخت‌های آن از پیش وجود ندارد. برای مثال، بسیاری از سازمان‌ها «قراردادهای استاندارد» برای فروش API تدارک ندیده‌اند یا روال خاصی برای مستندسازی و پشتیبانی از سرویس‌ها ندارند. این امر موجب عدم مطلوبیت سرویس می‌شود.

قابلیت یک‌پارچه‌سازی

هرچند که APIها ذاتا قرار است استقلال از زبان و پلتفرم و زیرساخت را ترویج دهند، اما برخی سازمان‌ها سرویس‌های خود را به‌صورتی ارائه می‌دهند که قابلیت یک‌پارچه‌سازی با خدمات سازمان‌های دیگر را نداشته باشد. به عبارت دیگر، یک راهکار «یک‌تکه» متشکل از ده‌ها API به مشتری عرضه می‌شود و مشتری باید یک انتخاب صفر و یکی داشته باشد؛ یا ده‌ها سرویس را با تمام مزایا و پیچیدگی از یک تأمین‌کننده دریافت کند، یا به سراغ تأمین‌کننده‌ی دیگر برود.

حریم خصوصی

بسیاری از APIها ممکن است حاوی داده‌های خصوصی و حفاظت‌شده‌ی کاربران باشند. حفظ امنیت داده‌ها یکی از دغدغه‌های مهم کسب‌وکارها است. همچنین ممکن است اسرار تجاری کسب‌وکار از این طریق افشا شود.

سناریوی کسب‌وکار

سرویس دریافتی چه تغییری در سناریوی کسب‌وکار ایجاد می‌کند؟ برای مثال، زیرساخت‌های مالی نوآورانه معمولاً می‌توانند تحول قابل‌توجهی در سناریوی کسب‌وکار ایجاد کنند. حساب‌های دیجیتال به کسب‌وکارها اجازه می‌دهند تا درآمدها را در خردترین سطح، تسهیم کرده و پیچیدگی‌های مالیاتی را در انتقال وجوه کاهش دهند. در مقابل، برخی سرویس‌ها موجب افزایش پیچیدگی می‌شوند.

تجربه‌ی مشتری

آیا دریافت سرویس موجب تحول تجربه‌ی مشتری (به شیوه‌ای خوشایند) می‌شود؟ برای مثال، تجربه‌ی کاربری سرویس‌های حمل‌ونقل، با امکان تسویه آنی بهبود یافته است. سرویس‌های احراز هویت متمرکز نیز سهم شایان توجهی در روان‌سازی تجربه مشتری در ورود به کسب‌وکارهای جدید داشته‌اند.

کاهش هزینه

APIها در سطح حداقلی، باید ارزش‌هایی از جنس کاهش هزینه‌ی ناشی از برون‌سپاری و اتوماسیون را خلق کنند. این ارزش‌ها به خصوص در مواردی که حجم عملیات بسیار بالا باشد، می‌تواند جذابیت قابل‌توجهی برای مشتری خلق کند. برای مثال، پلتفرم‌های مبتنی بر محتوای تولیدشده توسط کاربران (UGC) می‌توانند از هوش مصنوعی برای پایش محتوای خود استفاده کرده و محتوای غیراخلاقی را نشان‌گذاری کنند. سپس، ناظر پلتفرم می‌تواند محتوای نشان‌گذاری‌شده را بررسی کرده و اقدامات لازم را صورت دهد.

در ذهن سرویس‌گیرنده چه می‌گذرد؟

خود را جای سرویس‌گیرنده بگذارید. ببینید چه ارزشی برای وی خلق می‌کنید؟ چه هزینه‌ای به او تحمیل می‌کنید؟ از او چه می‌خواهید و در مقابل، به او چه می‌دهید؟



گزینش راهبرد

چگونه می‌توانیم وارد فضای API شویم؟ سازمان ما چه ظرفیتی برای ورود به این بازار دارد؟

عرفی، از ده مورد بیشتر نیستند و شبکه‌ی همکاران سازمان را تشکیل می‌دهند. طبیعتاً پشتیبانی فنی و روال‌های قراردادی مستقیماً توسط سازمان پیگیری می‌شود.

راهبرد سوم: مالکیت پلتفرم API

در راهبرد سوم، سازمان خود به سراغ ایجاد پلتفرم API می‌رود. سازمان‌هایی که طیف گسترده‌ای از سرویس‌ها را در اختیار دارند و قصد دارند شیوه‌ی تعامل جدیدی با زیست‌بوم کسب‌وکاری پیرامون خود داشته باشند، به سراغ این رویکرد می‌روند. این رویکرد بسیار هزینه‌بر و زمان‌بر است و برای عده‌ی انگشت‌شماری از سازمان‌ها مقرون‌به‌صرفه است.

راهبرد ترکیبی

در بسیاری موارد سازمان تصمیم می‌گیرد که ترکیبی از راهبرد اول و دوم را برگزیند. مشتری‌های مهم (شبکه‌ی همکاران سازمان) مستقیماً از سازمان سرویس می‌گیرند و کسب‌وکارهای خرد به یک پلتفرم API برون‌سپاری می‌شوند. در این شرایط، سازمان می‌تواند به‌صورت انتخابی، شیوه‌ی تعامل کسب‌وکاری خود با هر مشتری را تعیین کند.

برای گزینش راهبرد ورود به بازار API، عوامل مختلفی وجود دارند. در ابتدا به بررسی سه راهبرد اساسی ورود به بازار می‌پردازیم، این راهبردها به ترتیب، درگیری عملیاتی بیشتری را متوجه سازمان می‌کنند. چگونگی تقسیم کارکردها در هر راهبرد را در جدول ۱ آورده‌ایم.

راهبرد اول: برون‌سپاری بازاریابی، فروش و پشتیبانی

در این راهبرد، سازمان تمام سرویس‌ها را پس از پیاده‌سازی مستقیماً در اختیار یک پلتفرم فروش API به عنوان یک مشتری واحد قرار می‌دهد. سازمان هیچ گونه درگیری عملیاتی با سمت بازاریابی، فروش و پشتیبانی از سرویس‌گیرندگان ندارد و کنترل امنیت و دسترسی نیز تنها برای یک مشتری صورت می‌گیرد. این امر به سازمان اجازه می‌دهد که با فراغ بال، سرویس‌ها را توسعه داده و تنها بر کیفیت سرویس متمرکز شود.

راهبرد دوم: عرضه‌ی مستقیم

در این راهبرد، سازمان مستقیماً اقدام به عرضه‌ی سرویس می‌کند، اما نگاه پلتفرمی ندارد. در این شرایط، سازمان معمولاً به تعداد مشخصی سرویس‌گیرنده سرویس می‌دهد که به‌صورت

جدول ۱. شیوه‌ی توزیع کارکردها در راهبردهای مختلف عرضه‌ی API

کارکرد / راهبرد	برون‌سپاری	عرضه‌ی مستقیم	مالکیت پلتفرم
توسعه و پشتیبانی از سرویس	داخلی	داخلی	داخلی
بازاریابی سرویس / برندینگ	توسط پلتفرم	به‌صورت B2B توسط سازمان	به‌صورت چندکانالی توسط سازمان
فروش سرویس و پشتیبانی از مشتریان	توسط پلتفرم	توسط سازمان	توسط سازمان
روال مالی و قراردادی با مشتریان	توسط پلتفرم / روال استاندارد	توسط سازمان / روال اختصاصی	توسط سازمان / هم روال اختصاصی و هم روال استاندارد
کنترل دسترسی و نرخ تراکنش	توسط پلتفرم	توسط سازمان / معمولاً هزینه‌ی توسعه‌ی بالا، موجب ضعف سازمان در این کارکرد می‌شود	زیرساخت لازم توسط سازمان به‌صورت کامل توسعه داده می‌شود



راهبرد مناسب ما چیست؟ چگونه یکی از راهبردها را انتخاب کنیم؟

آماده باشیم. پلتفرم‌ها می‌توانند در این مسیر برای سازمان خلق ارزش کرده و بار عملیاتی فروش و پشتیبانی را به عهده بگیرند تا سازمان بتواند بر توسعه‌ی API‌ها متمرکز شود.

سوال دوم: روال و سناریوی استاندارد

به فرآیند فروش سرویس فکر کنید. برای مثال، سرویس‌های مالی معمولاً با «توافقات کسب‌وکاری» ارائه می‌شوند و قیمت‌گذاری استاندارد ندارند. در مقابل، سرویسی مانند استعلام وضعیت هوا می‌تواند با قیمت ثابت به ازای هر فراخوانی عرضه شود.

هنگامی که سرویس استاندارد داریم که می‌توانیم آن را به صورت شفاف و عمومی قیمت‌گذاری کنیم، برون‌سپاری فروش به پلتفرم‌ها مقرون‌به‌صرفه و توجیه‌پذیر است. اما در مواقعی که قیمت و روال عرضه‌ی سرویس به سناریوهای کسب‌وکاری پیوند می‌خورد، الزاماً برون‌سپاری سرویس به پلتفرم‌ها بهترین گزینه نخواهد بود.

یکی از مهم‌ترین دلایلی که پیش‌گامان توسعه‌ی پلتفرم‌های API در ایران، بانک‌ها و موسسات مالی هستند، همین است. این موسسات، هم تعداد مشتریان بالایی دارند و هم پیچیدگی و حساسیت بالایی را در روال عرضه‌ی سرویس تجربه می‌کنند. از این رو، معمولاً «مالکیت پلتفرم» برای آن‌ها بهترین گزینه است. آن‌ها نیاز دارند که تمام کارکردها و امکانات عرضه‌ی سرویس را به صورت داخلی ایجاد کنند تا کنترل و شفافیت کامل داشته باشند.

سوال سوم: توان فنی و بلوغ سازمان

همان‌طور که گفته شد، عرضه‌ی API نیازمند سطح مشخصی از توان فنی و زیرساخت‌های نرم و سخت در سازمان است. در نخستین روزهای ورود به این عرصه، سازمان با پیچیدگی‌های جدیدی مواجه می‌شود و به تدریج این پیچیدگی‌ها را هضم می‌کند.

تجربه نشان‌گر آن است که سازمان‌هایی که بدون تجربه‌ی پیشین در حوزه‌ی API و پلتفرم‌های نرم‌افزاری، به سراغ توسعه‌ی پلتفرم داخلی رفته‌اند، با شکست مواجه شده‌اند. از

هر یک از راهبردها برای یک سازمان و در یک مقطع خاص مناسب هستند. به عبارت دیگر، ممکن است امروز، راهبرد اول را انتخاب کنیم و سپس به سراغ راهبرد دیگر برویم. برای این موضوع باید چند پرسش را پاسخ دهیم:

۱. چند مشتری خواهیم داشت؟ توان عملیاتی ما چقدر است؟

۲. آیا روال و سناریوی استاندارد کسب‌وکاری برای عرضه‌ی سرویس به مشتریان داریم؟

۳. توان فنی و بلوغ ما در عرضه‌ی API چقدر است؟

۴. جایگاه API در مأموریت و چشم‌انداز سازمان چیست؟

با پاسخ به این پرسش‌ها می‌توانیم راهبرد ورود سازمان به حوزه‌ی API را مشخص کنیم. در ادامه به معنای هر یک از این پرسش‌ها و پیامدهای آن‌ها در راهبرد می‌پردازیم.

سوال اول: تعداد مشتریان

تا زمانی که تعداد مشتریان ما کم باشد، عرضه‌ی مستقیم می‌تواند برای سازمان جذاب‌تر باشد. مشتریان اندک معمولاً شبکه‌ی همکاران سازمان هستند و قابلیت اطمینان بالایی به آن‌ها وجود دارد. همچنین پشتیبانی فنی یا عقد قرارداد با آن‌ها معمولاً توان زیادی از سازمان طلب نمی‌کند. در این شرایط یک تیم سه‌نفره مرکب از یک پشتیبان فنی، یک پشتیبان کسب‌وکاری و یک سرپرست می‌تواند پاسخگوی نیازهای سازمان باشد.

اما باید پرسید که آیا تعداد مشتریان ما در طول زمان نیز ثابت است؟ به عبارت دیگر، آیا تعداد مشخصی کسب‌وکار در شبکه‌ی همکاران خود داریم که می‌دانیم حداقل تا افق یک سال ثابت هستند؟ یا صرفاً در فاز پایلوت هستیم و قرار است در ادامه، مشتریان بیشتری داشته باشیم؟

در حالتی که تعداد مشتریان ما تا افق میان‌مدت ثابت باشند، می‌توانیم مستقیماً به سراغ عرضه‌ی API برویم. اما در مواردی که می‌دانیم در آینده تعداد مشتریان بیشتر می‌شود، گزینه‌ی پلتفرم احتمالاً گریزناپذیر خواهد بود و در غیر این صورت، باید برای بار عملیاتی پیش‌بینی‌نشده‌ی قابل‌توجهی

مسئله‌ی مرغ و تخم مرغ

در ارزیابی راهبردها باید در نظر داشته باشید که به هر صورت، بازار API در حال حاضر در حوزه‌هایی به جز نقشه، احراز هویت، خدمات بانکی، مالی و پراخت، تقاضای قابل توجهی ندارد. بدین ترتیب، تمام این رویکردها در بدو امر با واکنش سرد از سوی بازار مواجه خواهند شد.

کلید حل این مشکل، ورود گام‌به‌گام و محتاطانه است. در واقع در ابتدا باید با منابعی کمتر به بازار ورود کنید و کم‌کم، با آگاهی‌بخشی درباره‌ی حضور خود، تقاضا را شکل بدهید. چرا که کسب‌وکارها باید با آگاهی از وجود این سرویس، سناریوهای کسب‌وکاری خود را طراحی کنند و این فرآیند مستلزم ماه‌ها زمان است. فرآیندی که در کسب‌وکارهای سنتی به آن «خاک‌خوردن» گفته می‌شود، در بازار نابالغ API به وفور و شدت مشاهده می‌شود و باید برای مواجهه با آن و تداوم در طول آن، آمادگی و منابع کافی داشته باشید.

جمع‌بندی

در این بخش، گزینه‌های راهبردی و سوال‌های بنیادین برای انتخاب یکی از گزینه‌ها را مرور کردیم. نقشه‌ی راه ما برای عرضه‌ی API اکیدا تابع این موارد است و حجم منابع مورد نیاز ما را تعیین می‌کند. طبیعی است که هرچه به سمت برون‌سپاری برویم، منابع کمتری مصرف می‌کنیم و در مقابل، کنترل و شفافیت پایین‌تری خواهیم داشت؛ هرچه به سمت توسعه‌ی داخلی زیرساخت حرکت کنیم، ضمن دستیابی به کنترل و شفافیت بیشتر، منابع بسیار بیشتری درگیر می‌شوند.

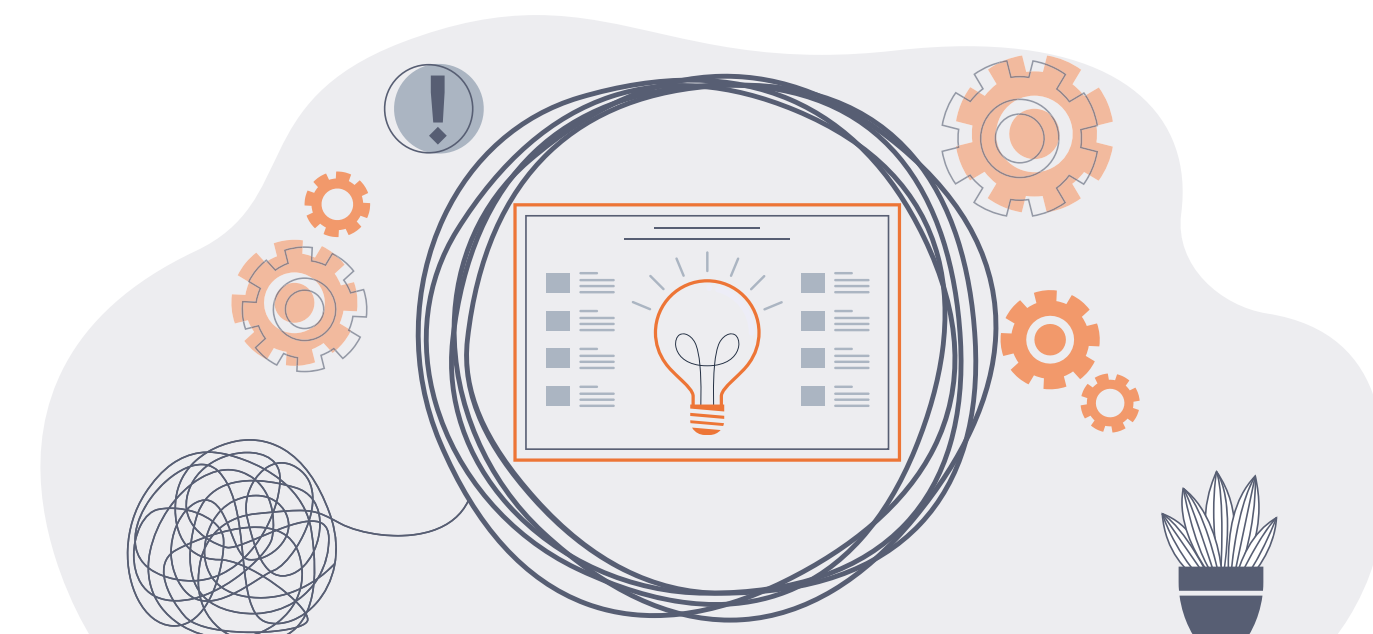
این رو توصیه می‌شود تا سازمان، ابتدا یک راهبرد «غایی و مطلوب» را تعریف کند و سپس به صورت گام‌به‌گام به سوی آن حرکت کند.

برای مثال، سازمان می‌تواند در هنگام ورود به بازار، با یک پلتفرم همکاری کرده و تا حد امکان، بر توسعه‌ی سرویس متمرکز شود. سپس می‌تواند ارائه‌ی سرویس مستقیم را تجربه کند و در کنار پلتفرم، تعدادی مشتری VIP را به صورت مستقیم هدف بگیرد. با عبور از این دو گام، سازمان می‌تواند با شناخت و توانمندی بالاتری به سمت توسعه‌ی پلتفرم حرکت کند. توصیه این است که سازمان پیش از ایجاد پلتفرم، حداقل دو سال تجربه‌ی عرضه‌ی API داشته باشد.

سوال چهارم: چشم‌انداز و راهبرد

اسناد فرادستی در بسیاری مواقع، تکلیف سازمان را با API‌ها مشخص می‌کنند. به عنوان یک موسسه مالی، احتمالا تحقق چشم‌انداز پنج‌ساله‌ی سازمان، بدون ایجاد یک پلتفرم API با سطح مشخصی از امکانات، در سایه‌ای از ابهام خواهد بود. نهادهای مالی حتی برای عرضه‌ی سرویس به چند پلتفرم API، خود نیازمند ایجاد یک پلتفرم اولیه با امکانات حداقلی هستند.

در مقابل، بسیاری از سازمان‌ها تنها باید کانالی برای عرضه‌ی API داشته باشند و این کار را می‌توانند به طور کامل از طریق یک پلتفرم بیرونی انجام دهند. در این سازمان‌ها، ورود به عرصه‌ی API چیزی جز حواس‌پرتی و پراکنده کردن منابع سازمان در بر ندارد.





طراحی سبد API ها

چه API هایی را عرضه کنیم؟ رهنگاشت سبد چیست؟

انجام می‌شود، از هم گسسته شود. در نتیجه، جذابیت آن برای مشتری کاهش یافته و مشتری ممکن است به سراغ شیوه‌های قدیمی‌تر پرداخت، مثل پرداخت با دستگاه خودپرداز برود.

مثالی که بررسی شد، ساده‌ترین شیوه‌ی وابستگی API های یک سبد به یک‌دیگر بود. هنگامی که تعداد API ها بیشتر باشد، وابستگی‌های دیگری نیز شکل می‌گیرد. برای مثال، یک سامانه‌ی حمل‌ونقل بین شهری را در نظر بگیرید. در این‌گونه API ها معمولاً پارامترهایی وجود دارد که باید به‌صورت استاندارد وارد شود. مثلاً پارامتر «نام شهر» معمولاً با یک «کد» مشخص می‌شود. در نتیجه، شما پیش از اینکه هر گونه API داشته باشید، باید یک نظام استاندارد برای کدهای شهرها ایجاد کنید. حال تصور کنید ده‌ها API قرار است از این نظام استاندارد کدگذاری استفاده کنند. بنابراین، تهیه‌ی نظام کدگذاری پیش از طراحی API ها یک الزام خواهد بود.

در نتیجه، باید گفت که API ها با اینکه ماژولار هستند، اما سطحی از پیوستگی و یکپارچگی را به‌صورت درونی خواهند داشت. حال باید چه کرد؟

در این مواقع می‌توانید یک سبد پیش‌نویس تهیه کنید و پارامترهای ورودی و خروجی هر سرویس را در نظر بگیرید. بدین ترتیب، می‌توانید وابستگی‌های پارامترهای هر سرویس را به گام‌های قبلی و بعدی مشخص کنید. همچنین پارامترهایی که به‌صورت استاندارد در سرویس‌های شما وجود دارند را فهرست کرده و در صورت نیاز، برای آن‌ها یک کدگذاری مشخص تعریف کنید.

رهنگاشت سبد

سوال مهم اینجاست که چه زمانی می‌توان سبد را به بازار عرضه کرد؟ در پاسخ به این پرسش، شما باید برای سبد API های خود، یک رهنگاشت تنظیم کنید. این رهنگاشت به شما می‌گوید چه زمانی وارد بازار شوید و چگونه سبد را توسعه دهید.

برای سبد نهایی خود یک برنامه زمانی نظیر «گانت‌چارت»

برای خواننده‌ی هوشمند، شاید اولین سوال این باشد: «مگر نه اینکه API ها ماهیتی ماژولار دارند؟ پس چه نیازی به طراحی سبد از پیش وجود دارد؟ هر زمان که لازم شد، یک سرویس جدید به سبد می‌افزاییم!»

پاسخ این است که متأسفانه در بسیاری موارد، API ها در سناریوهای کسب‌وکاری با یک‌دیگر ترکیب می‌شوند و امکان عرضه‌ی یکی از آن‌ها به تنهایی وجود ندارد. به عبارت دیگر، وقتی سبد به‌صورت نظام‌یافته طراحی نشود، ممکن است ارزش خلق‌شده توسط آن ابتر بماند. بیایید این موضوع را با مثالی شرح بدهیم:

امروز، سرویس پرداخت قبض همراه اول شامل سه بخش اساسی است: (۱) استعلام قبض (۲) پرداخت قبض و (۳) ثبت آنی پرداخت قبض.

در گام اول، مشتری باید بداند که چه میزان بدهی دارد، در گام دوم مشتری بدهی خود را پرداخت می‌کند و در گام سوم، کارگزار پرداخت به همراه اول اطلاع می‌دهد که قبض پرداخت شده است. تا چندی پیش، ثبت پرداخت قبض در سامانه‌ی همراه اول حداقل یک روز کاری طول می‌کشید و در تسویه‌های شبانه‌ی همراه اول با بانک‌ها انجام می‌شد. امروز، پیش از اینکه وجه پرداخت‌شده توسط مشتری به حساب همراه اول واریز شود، پرداخت آن در سامانه‌های همراه اول ثبت می‌شود. این امر به مشتری آرامش می‌دهد که پرداخت وی تکمیل شده و دیگر نیازی به اقدام و پیگیری بیشتر ندارد. همچنین وصل شدن خطوطی که به علت بدهی قطع شده‌اند، به‌صورت آنی صورت می‌گیرد.

در فرآیند فوق، ورودی اولین API تنها یک شماره تلفن همراه اول است. سپس خروجی اولین API به عنوان ورودی دومین قدم استفاده می‌شود و خروجی قدم دوم، در واقع ورودی قدم سوم است. حال تصور کنید هر یک از این سه سناریوی کسب‌وکاری در سبد API ها نادیده گرفته شوند. این امر، باعث می‌شود تا روال خودکاری که تنها با چند اشاره در یک اپلیکیشن کیف پول



جدول ۲. نمونه‌ی پیش‌نویس سبد API ها

ورودی‌ها	خروجی‌ها	ملاحظات
سرویس اعلام قبض	شناسه‌ی قبض و شناسه‌ی پرداخت	-
سرویس پرداخت قبض	اطلاعات تراکنش پرداخت	وابستگی به گام قبل
سرویس ثبت آتی پرداخت قبض	تاییدیه‌ی ثبت	وابستگی به گام قبل

در رویکرد «سفر مشتری» می‌توانید از مشتریان بالقوه‌ی خود کمک بگیرید و از آن‌ها بپرسید سناریوی کسب‌وکاری مطلوب آن‌ها چیست؟ آن‌ها یک سرویس «خوش‌تعریف» را چگونه می‌بینند و در هر قدم، چه ورودی‌ها و چه خروجی‌هایی در اختیار دارند؟ سرویس‌های مکمل برای آن‌ها چیست و برای دریافت سرویس از شما، حداقل چه پارامترهایی باید پوشش داده شود؟ اما ممکن است مشتری کسب‌وکاری بالقوه برای سرویس‌های شما وجود نداشته باشد. در این صورت چه کنیم؟ در این موقع به سراغ طراحی تجربه‌ی کاربر نهایی می‌رویم. برای مثال، فرض می‌کنیم مشتری هنگامی که قصد خرید یک لباس را دارد، چه قدم‌هایی را طی می‌کند و برای هر قدم، سرویس مناسبی طراحی می‌کنیم.

در نگاه «منبع‌محور»، سازمان API ها را بر مبنای منابع موجود طراحی می‌کند. برای مثال، شهرداری‌ها در ایران، معمولاً با نگاه منبع‌محور خدمات خود را عرضه می‌کنند. این نگاه به ویژه برای سازمان‌های عمومی می‌تواند مفید باشد، چرا که «انتفاع تجاری» برای نهادهای عمومی نباید معیار ارائه‌ی سرویس قرار گیرد. اما باید در نظر داشت که نگاه منبع‌محور به احتمال بسیار زیاد، موجب شکست تجاری سرویس‌ها خواهد شد. چرا که بسیاری از سرویس‌های قابل ارائه، طبق آنچه در بخش «ارزیابی بازار» دیدیم، جذابیت بازاری خاصی نخواهند داشت. برای مثال، شهرداری‌های تهران و مشهد در حال حاضر، داده‌های قبور متوفیان را از طریق API عرضه می‌کنند. سرویسی که با استقبال بسیار ضعیف مواجه شده است. اگر از منظر کاربر، چنین سرویسی را بررسی کنید، خواهید دید که یک پتل جست‌وجو برای چنین سرویسی کفایت می‌کند و ارائه‌ی آن در قالب API احتمالاً به لحاظ تجاری موفق نخواهد بود.

تعریف کنید. هر API ابتدا به صورت داخلی توسعه داده می‌شود، سپس وارد مرحله تست و ارزیابی می‌شود و پس از تکمیل مستندسازی، می‌تواند به بازار عرضه شود.

هنگامی که یک بسته شامل یک یا چند API داشته باشید که «وابستگی به غیر» نداشته باشند و در ستون ملاحظات، تمام وابستگی‌های آن‌ها فراهم باشد، می‌توانید به بازار وارد شوید. سپس می‌توانید طبق گانت‌چارت، سرویس‌های جدید را به بازار عرضه کرده و ارزش‌های جدیدی خلق کنید.

در نظر داشته باشید که بسیاری از وابستگی‌ها ممکن است در گام اول آشکار نباشند. در مثالی که برای سرویس‌های پرداخت قبض همراه اول آوردیم، با یک معمای «حل‌شده» مواجه هستیم. در حالی که الزاماً فرآیندها این‌طور مشخص و بدیهی نیستند. همان‌طور که گفتیم، تا چند سال پیش، الزام «ثبت آتی پرداخت قبض» بدیهی نبود و کاربران برای ثبت پرداخت قبض خود به دفاتر خدمات مشترکان مراجعه می‌کردند.

متأسفانه یا خوشبختانه، روال استاندارد برای این موضوعات وجود ندارد و هوشمندی کسب‌وکاری سازمان‌ها در طراحی سبد است که به یاری آن‌ها می‌آید. اما به طور کلی، سبد سرویس‌های عرضه‌شده توسط سازمان، به طور معمول بر پایه‌ی «سفر مشتری» یا «منابع سازمان» ساخته می‌شوند.

متأسفانه در بسیاری موارد، API ها در سناریوهای کسب‌وکاری با یک‌دیگر ترکیب می‌شوند و امکان عرضه‌ی یکی از آن‌ها به تنهایی وجود ندارد. به عبارت دیگر، وقتی سبد به صورت نظام‌یافته طراحی نشود، ممکن است ارزش خلق‌شده توسط آن ابتر بماند.

راهبرد قیمت‌گذاری سرویس

تعیین راهبرد قیمتی سرویس ملاحظات متعددی را در پی دارد.

می‌شود. سرویس‌های پرداخت قبوض، پرداخت عوارض، پرداخت اقساط، فروش بیمه، فروش شارژ سیم‌کارت و بسته‌ی اینترنت و... از این دست هستند. پس در صورتی که چنین سرویس‌هایی را در پورتفوی خود دارید، باید سناریوی مالی مجزایی را برای آن‌ها طراحی کنید. چرا که نمی‌توانید به سادگی «فروش خدمت» و «ثبت فاکتور» با این سرویس‌ها برخورد کنید.

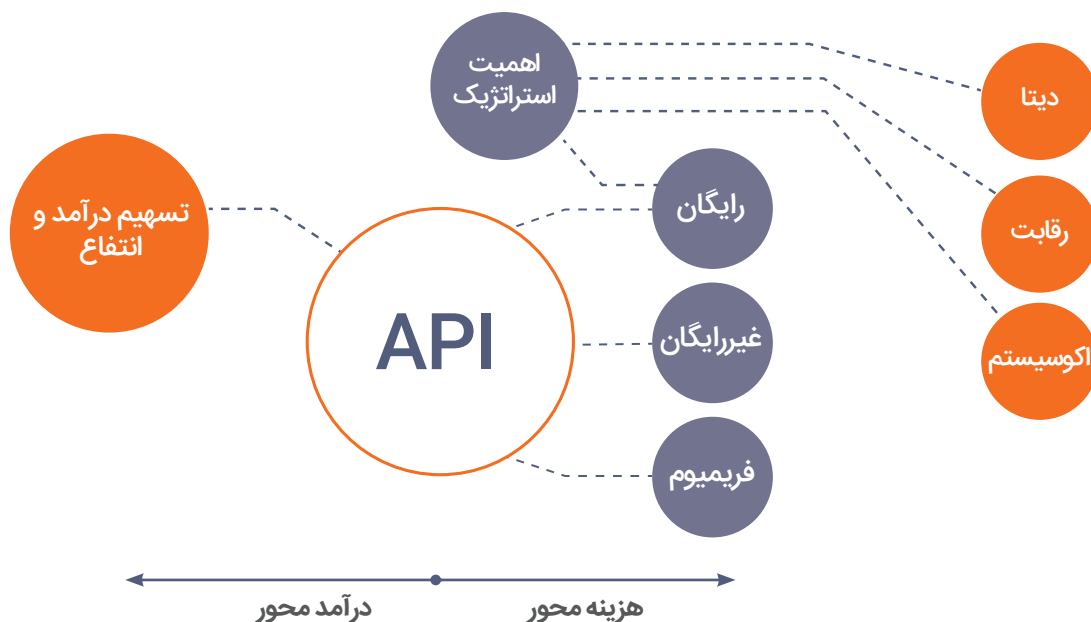
۲. سرویس‌های غیررایگان: این سرویس‌ها معمولاً با سناریوی هزینه‌ای مشخصی فروخته می‌شوند و مشتری به ازای هر فراخوانی (یا تعداد مشخصی فراخوانی)، هزینه‌ی مشخصی را پرداخت می‌کند. دقت داشته باشید که اگر سازمان شما تابع قوانین خاصی است، ممکن است در قیمت‌گذاری سرویس‌ها اختیاری نداشته باشید یا حداقل، نیازمند تایید قیمت‌گذاری توسط نهادهای دیگر باشید. مثلاً یک کمیته، یک کارگروه

اکنون می‌دانیم چه سرویسی را با چه راهبردی و برای چه بازار هدفی قصد داریم عرضه کنیم. بدین ترتیب، سوال‌های اساسی برای قیمت‌گذاری پاسخ داده شده است.

اکنون به مبانی کلی قیمت‌گذاری می‌پردازیم. به طور کلی می‌توان از منظر ساختار هزینه‌ی سرویس، API‌ها را به چهار دسته تقسیم کرد:

۱. API‌های درآمدزا: این API‌ها از جنس فروش سرویس هستند و خود، متضمن درآمد هستند. برای مثال، فرض کنید که شما مالک یک شرکت بیمه هستید و ثبت بیمه‌ی شخص ثالث خودرو را به صورت API در اختیار یک کارگزار قرار دهید. این کارگزار به ازای هر فراخوانی، می‌تواند ۴ درصد از هزینه‌ی ثبت بیمه را به عنوان کارمزد دریافت کرده و مابقی مبلغ را به حساب شما منتقل کند. در این گونه API‌ها، سناریوی مالی به صورت تسهیم درآمد یا انتفاع طراحی

شکل ۷
مدل‌های درآمدی API





ملاحظات دیگری در قیمت‌گذاری

در نظر داشته باشید که قیمت‌گذاری سرویس‌ها یک موضوع «وابسته به مسیر» است. بدین معنا که شما اگر با یک راهبرِ قیمتی وارد شوید، احتمالاً نمی‌توانید به سادگی راهبرِ خود را تغییر دهید. چرا که بازار، به دقت رفتار قیمتی شما را رصد می‌کند.

فرض کنید با قیمت‌گذاری بسیار بالا به بازار وارد می‌شوید و پس از شکست تجاری، تصمیم به کاهش قیمت می‌گیرید. شما با این کار، یک پیام واضح به بازار می‌دهید: «سرویس ما آن قدر نمی‌ارزید!»

در مقابل، اگر با قیمت‌گذاری پایین به بازار وارد شوید، مشتریان به سطحی از قیمت عادت می‌کنند و حتی ممکن است رقبا مجبور شوند قیمت خود را کاهش دهند، پس احتمالاً در یک بازه‌ی قیمتی مشخص (که پایین‌تر از مقدار بهینه است) قفل می‌شوید.

حال چه باید کرد؟ برای پرهیز از این اتفاقات، این گام‌ها را دنبال کنید:

- پیش از ورود به بازار، حتماً مطالعات عمیق بازار انجام دهید. با مشتریان بالقوه مصاحبه کنید و از آن‌ها بپرسید چقدر حاضرند برای سرویسی با این مشخصات هزینه کنند؟

- در بدو امر، قیمت‌ها را به صورت عمومی اعلام نکنید و با رویکرد توافقی حرکت کنید. ببینید که مشتریان در نهایت با چه قیمتی با شما معامله می‌کنند. قیمتی که در آن مشتری از شما سرویسی خریده باشد، احتمالاً معتبرتر از هر قیمتی است که با مطالعات بازار به دست می‌آید!

- متأسفانه یا خوشبختانه هنوز بسیاری از سرویس‌ها در ایران عرضه نشده‌اند و بسیاری از مواقع، شما اولین بازیگر هستید. اگر رقیبی در بازار وجود داشته باشد، شما به سادگی با تحلیل رقبا به قیمت‌گذاری صحیح می‌رسید، اما در شرایطی که رقیبی وجود ندارد، باید به سراغ موارد خارجی بروید. قیمت‌گذاری سرویس‌های خارجی را می‌توانید با عواملی نظیر برابری قدرت خرید، یا نسبت هزینه‌ی سرویس به حقوق ماهیانه‌ی یک کارمند، تعدیل کنید.

یا شورای مشخص، وظیفه‌ی قیمت‌گذاری سرویس‌های شما را به عهده دارد و شما باید تایید این نهاد را دریافت کنید. از طرف دیگر، ممکن است بنا به ملاحظات فردستی، موظف به ارائه‌ی سرویس رایگان باشید.

۳. سرویس‌های فریمیوم: این سرویس‌ها به صورت ترکیبی از سرویس رایگان و غیررایگان عرضه می‌شوند و افرادی که بخواهند به شیوه‌ی خاصی از سرویس استفاده کنند، باید هزینه‌ی آن را پرداخت کنند. مثلاً، بسیاری از API‌های عمومی موجود روی وب، بر مبنای دفعات فراخوانی محدود می‌شوند. کاربر رایگان می‌تواند تا سقف روزانه ۱۰۰ فراخوانی انجام دهد، اما اگر تعداد فراخوانی بیشتری نیاز باشد، باید هزینه‌ای پرداخت شود. این محدودیت ممکن است روی امکانات خاصی اعمال شود. مثلاً سرویس پیش‌بینی وضع هوا را تصور کنید. کاربر عادی ممکن است بتواند دمای هوا و وزش باد را دریافت کند، اما شدت رطوبت و بارندگی به صورت رایگان در اختیار وی قرار نمی‌گیرد. شما در طراحی سرویس خود می‌توانید محدودیت‌های نظام فریمیوم را روی تعداد فراخوانی یا امکانات یا داده‌های خاصی اعمال کنید.

۴. سرویس‌های رایگان: شما ممکن است تصمیم بگیرید سرویس خود را به صورت رایگان ارائه دهید. این تصمیم بنا به سه دسته مصلحت‌سازمانی صورت می‌گیرد:

- **توسعه‌ی اکوسیستم:** به منظور توسعه‌ی برند و گسترش اکوسیستم کسب‌کارهای متصل به خود، سرویس را رایگان ارائه می‌دهید. در این رویکرد، جذب حداکثری کسب‌وکارها در اولویت است و بعداً مدل‌های درآمدی دیگری تعریف می‌شوند.

- **رقابت:** اگر رقیب شما سرویسی را رایگان یا ارزان عرضه می‌کند، شما احتمالاً باید به سراغ ارائه‌ی سرویس رایگان بروید. در این نقطه، شما مزیت اولین بازیگر را از دست داده‌اید و تابع بازی بازار هستید.

- **دیتا:** بسیاری از سرویس‌ها با خود انبوهی از دیتا را به همراه می‌آورند. برای مثال، شرکت‌های خدمات پرداخت (PSP)، اشراف اطلاعاتی کاملی نسبت به تراکنش‌های تمام درگاه‌های پرداخت دارند. این اشراف در کسب‌وکار شما ممکن است آن قدر ارزش ایجاد کند که تصمیم به ارائه‌ی سرویس رایگان بگیرید.



ساختار و منطق قیمت API

قیمت‌گذاری صحیح سرویس، کلید فروش موفق شماست.

۲. **فرکانس محاسبه:** بسته به حجم فراخوانی‌ها ممکن است تصمیم بگیرید فرکانس‌های محاسبه‌ی هزینه را روزانه، ماهانه، فصلی و حتی سالانه در نظر بگیرید.

۳. **محصول:** در حالتی که یک سرویس عادی می‌فروشید، قیمت‌گذاری می‌تواند به ازای فراخوانی سرویس و دریافت پاسخ آن معنا پیدا کند. اما هنگامی که API همراه با یک خدمت بیرونی عرضه می‌شود و در واقع یک محصول بیرونی از طریق API خریداری می‌شود، مدل پرداخت هزینه ممکن است به نقاط مختلفی اعمال شود. مثلاً سناریوی درخواست پیک از طریق API را در نظر بگیرید. در این سناریو ممکن است برای درخواست سرویس و نهایی‌سازی آن، چند API فراخوانی شوند. اما تمام این API‌ها رایگان هستند و هزینه در زمانی دریافت می‌شود که محصول به مقصد رسیده است. در واقع ممکن است سناریوی پرداخت اساساً از مسیر API عبور نکند.

اگر قصد دارید سرویس «غیررایگان» تعریف کنید، منطق قیمت‌گذاری شما باید چندین عامل را لحاظ کند. پیش از هر چیز، باید به خلق ارزش برای مشتری و حداکثر تمایل مشتری برای پرداخت هزینه فکر کنید. پیش از این، درباره‌ی این موضوع صحبت کردیم و مجدداً بر آن تأکید می‌کنیم. قیمت‌گذاری سرویس شما با هر منطق و ساختاری که صورت گیرد، نباید از «ارزشی» که برای مشتری خلق می‌کند، قیمت بالاتری داشته باشد.

از سوی دیگر، سرویس برای شما هزینه‌ی تمام‌شده‌ای خواهد داشت و می‌توانید تخمین اولیه‌ای از این هزینه‌ها (در کنار منافع مالی یا راهبردی سرویس) داشته باشید. در این نقطه، می‌توانید یک «حداقل» هزینه برای سرویس در نظر بگیرید. با فرض اینکه چنین فاکتورها و کف و سقف‌هایی در قیمت‌گذاری لحاظ شده‌اند، شما می‌توانید وارد قیمت‌گذاری شوید.

ارکان قیمت‌گذاری عبارتند از:

۱. **مدل کلی:** به طور کلی می‌توانید از مدل «پرداخت به ازای استفاده» به عنوان مدل پایه‌ای استفاده کنید. این مدل، انعطاف‌پذیری بالایی به مشتری می‌دهد و مشتری می‌تواند به سادگی هزینه‌های خود را محاسبه کند. چالش این مدل که به آن Pay-as-you-go نیز گفته می‌شود، این است که سرویس‌دهنده نمی‌تواند تخمینی از دفعات فراخوانی توسط مشتری داشته باشد. مدل دیگر، تعریف بسته‌های ثابت (مثلاً ۱۰۰۰ فراخوانی) است. این مدل، به سرویس‌دهنده اجازه می‌دهد پیش‌بینی دقیق‌تری داشته باشد و روال مالی آن نیز به لحاظ صدور فاکتور، آسان‌تر است. در نهایت، بسیاری از سرویس‌دهندگان از مدل ترکیبی استفاده می‌کنند. مزیت مدل ترکیبی این است که سرویس پس از پایان یافتن یک بسته، قطع نمی‌شود و مشتری می‌تواند سرویس خود را (با قیمت بالاتر) دریافت کند. بسته‌های اینترنت همراه در ایران با این منطق ارائه می‌شوند.

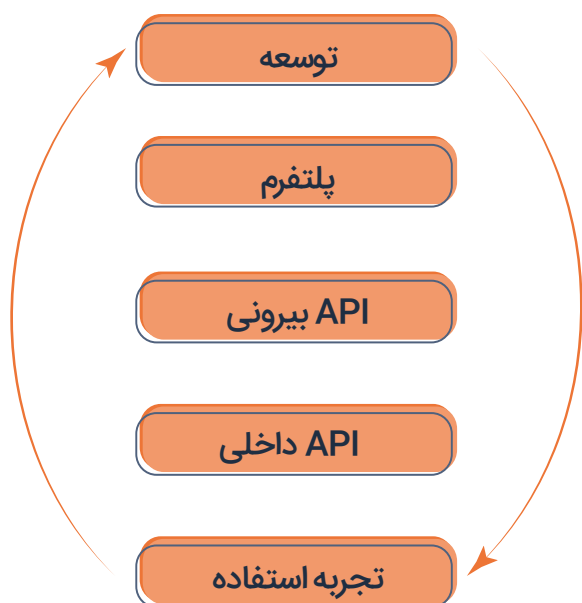
هشدار!!!!

پیش از تعیین مدل قیمت‌گذاری، زیرساخت‌های مالی شرکت خود را بررسی کنید. ببینید شرکت شما با چه شیوه‌هایی برای صدور فاکتور سازگارتر است؟

نظام‌های مدیریت مالی سنتی در ایران، آمادگی لازم برای پذیرش پیچیدگی‌های محاسبه‌ی درآمدها و هزینه‌های API را ندارند. این امر در آینده ممکن است دامن‌گیر کسب‌وکار شما بشود. پس در این مسیر، حتماً با واحد مالی سازمان هماهنگ باشید!

از کجا شروع کنیم؟

برای سازمان‌هایی که امروز به حوزه‌ی API وارد می‌شوند



کرده و محصول خود را بهینه‌سازی کنید. حتی پیش‌روترین سازمان‌ها در ایران، هنوز با سطوح بلوغ جهانی در این زمینه فاصله دارند و فضای بسیار زیادی برای توسعه و بهینه‌سازی در این بازار وجود دارد.

در نظر داشته باشید که مدیریت راهبردی API مانند هر امر دیگری در حوزه‌ی کسب‌وکار، تابع هزاران عامل اقتضایی است که ممکن است منحصر به سازمان شما باشد. آنچه در این سند مطرح شد، تنها پیش‌نویسی برای مسیر کلی شماسست و هر سازمان بنا به تجربه و اقتضائات خود، ممکن است فرآیند مجزایی را طی کند. در نتیجه، اعتماد به هیچ قالب از پیش آماده‌ای نمی‌تواند برای تمام سازمان‌ها کارگشا باشد و هیچ شاه‌کلیدی برای حل مسائل تمام سازمان‌ها وجود ندارد.

اگر API برای سازمان شما مفهوم نویی است، بهتر است پیش از توسعه‌ی API و عرضه‌ی آن، به سراغ استفاده از API بروید و تجربه‌ی استفاده از API را به عنوان یک مشتری مشاهده کنید. در گام بعد می‌توانید API‌های داخلی توسعه دهید و از API‌ها در لایه‌ی داخلی کسب‌وکار خود بهره ببرید.

اگر با مفاهیم API آشنایی کافی دارید و توان فنی لازم را در سازمان خود می‌بینید، به سراغ عرضه‌ی API بروید. در گام اول می‌توانید API‌هایی برای شبکه‌ی همکاران خود ارائه دهید و تست‌های اولیه را انجام دهید. در این فضا شما می‌توانید چالش‌های داخلی خود را در زمینه‌ی API برطرف کنید، ساختارها و فرآیندهای لازم را مستقر کرده و برای عملیات گسترده‌تر آماده شوید.

در اولین قدم در مواجهه با بازار، می‌توانید به سراغ پلتفرم‌ها بروید. پلتفرم‌ها این فرصت را در اختیار شما قرار می‌دهند در اولین قدم، با بازار در تماس باشید و به عنوان عرضه‌کننده‌ی API دیده شوید. آن‌ها همچنین عملیات فروش، کنترل دسترسی و پشتیبانی را مدیریت می‌کنند.

در گام بعد، می‌توانید به سراغ توسعه‌ی پلتفرم داخلی خود بروید. این پلتفرم می‌تواند شروع فصل جدیدی در کسب‌وکار شما باشد و جایگاه جدیدی در زنجیره‌ی ارزش به کسب‌وکار شما ببخشد.

در نظر داشته باشید که مالکیت پلتفرم از نظر هزینه‌های سرمایه‌گذاری و عملیاتی، فاصله‌ی بسیار زیادی با گزینه‌های راهبردی دیگر دارد. اما مزایای راهبردی قابل‌توجهی نیز در بر داشته و به شما اجازه می‌دهد پیش‌گذار ارزش جدیدی به بازار عرضه کنید. برای مثال، ممکن است یک پلتفرم ثالث، تمایلی به ارائه‌ی برخی سرویس‌های نوین شما نداشته باشد، اما مالکیت پلتفرم به شما اجازه‌ی آزمایش و اکتشاف می‌دهد. در هر صورت، مالکیت پلتفرم از بالاترین سطوح بلوغ در مدیریت API محسوب می‌شود و در کوتاه‌مدت دست‌یافتنی نیست.

در نهایت، پیشنهاد می‌کنیم که در تمام این مسیر، حلقه‌ی بازخورد را در نظر داشته باشید. پیوسته از بازار بازخورد دریافت



راه بی‌بازگشت

ورود به حوزه‌ی API چه پیامدهایی برای سازمان شما دارد؟

توجه داشته باشید که هزینه‌های ارائه‌ی سرویس بسیار بیش از این‌ها خواهد بود. توسعه، نگهداشت، مدیریت مالی، مدیریت فرآیندی، پشتیبانی و... همگی بارهایی بر دوش سازمان شما هستند و تا زمانی که سرویس‌ها «زنده» باشند، شما باید این هزینه‌ها را پرداخت کنید.

به ویژه که به طور معمول، سرویس‌ها را باید عطف به تغییرات دیگر در محیط داخلی و بیرونی، به‌روز نگاه دارید و این تغییرات نیز خود هزینه‌های جدیدی در پی دارند. مثلاً در به‌روزرسانی سرویس‌ها باید مراقب «سازگاری با نسخه‌های پیشین» باشید؛ بدین معنا که مشتریان نسخه‌های قبلی سرویس، نباید ناگهان دست خالی بمانند. هرگونه تغییر و مهاجرت در سرویس‌ها می‌تواند چالش‌های متعددی را در زمینه‌ی پشتیبانی و ارتباط با مشتریان ایجاد کند.

هزینه‌های رفتن

تصور کنید که امروز شما به شبکه‌ای از کسب‌وکارها متصل شده‌اید، اما به دلیل هزینه‌های بالای نگهداشت سرویس‌ها، قصد دارید سرویس‌های خود را خاموش کنید.

کسب‌وکارهای شبکه‌ی همکار شما، مدل کسب‌وکار خود را بر مبنای سرویس‌های شما طراحی کرده‌اند و در سطح حداقلی و در صورت امکان، مجبور به مهاجرت به رقبا هستند. حتی ممکن است مجبور به تغییر مدل‌های درآمدی یا سناریوهای تجربه‌ی مشتری شوند. این امر برای آن‌ها هزینه در پی خواهد داشت. در نتیجه، تصویر و اعتبار برند شما در حوزه‌ی API قطعاً تحت تأثیر قرار خواهد گرفت.

با تمام این اوصاف باید گفت که پیش از ورود به این حوزه، جوانب موضوع را به دقت بسنجید. آیا صبر و حوصله و بودجه‌ی لازم برای ورود به این عرصه را در اختیار دارید؟

مسیر عرضه‌ی API، یک راه بی‌بازگشت است. یعنی اگر به آن ورود کردید و جای پای خود یافتید - یعنی سازمان دیگری سرویس‌های شما را دریافت کرد - دیگر نمی‌توانید از آن خارج شوید. حداقل باید گفت که خروج شما هزینه‌های سنگینی خواهد داشت. ماندن هم البته بی‌هزینه نیست.

هزینه‌های ماندن

در ابتدا باید به هزینه‌های «حضور» در عرصه‌ی API بپردازیم. همان‌طور که گفته شد، برای عرضه‌ی API شما سه دسته منابع را درگیر می‌کنید:

۱. منابع انسانی
۲. زیرساخت سخت‌افزاری
۳. زیرساخت نرم‌افزاری

تمام این زیرساخت‌ها برای شما هزینه‌بر هستند و تا مدت‌ها پس از عرضه‌ی API به نقطه‌ی سر به سر نمی‌رسند. چرا که هزینه‌ی هر فراخوانی معمولاً بسیار پایین است و دریافت فراخوانی در تعداد بسیار بالا است که هزینه‌های فوق را پوشش می‌دهد. بیا یک محاسبه‌ی بسیار ساده انجام دهیم. فرض کنیم هزینه‌ی هربار فراخوانی سرویس به طور میانگین ۱۰۰ تومان باشد (که رقم مناسبی است) و سازمان تنها یک کارشناس با حقوق ماهیانه ۱۰ میلیون تومان به مدیریت API‌ها تخصیص داده باشد (که با سربار، ماهیانه جمعاً ۲۰ میلیون تومان هزینه دارد). برای پوشش این هزینه، شما نیازمند ۲۰۰ هزار فراخوانی ماهانه هستید. اگر این رقم را به روز بشکنید، به رقمی در حدود ۶۷۰۰ فراخوانی روزانه می‌رسید. آیا سرویس شما این قدر تقاضا دارد؟

در امتداد همین مثال، فرض کنیم که تنها موفق به جذب یک مشتری شده‌اید که با فراخوانی روزانه ۵۰۰ مرتبه، کسب‌وکار خود را به این سرویس گره می‌زند. این رقم ماهیانه ۳ میلیون تومان درآمد برای شما ایجاد می‌کند. شما به طور خالص، ماهیانه ۱۷ میلیون تومان زیان می‌کنید تا این سرویس به دست مشتری برسد.

به دنیای پلتفرمی خوش آمدید

API ها کسب‌وکار شما را در آینده به «پلتفرم» تبدیل می‌کند

بنابراین، در محاسبه‌ی هزینه‌های «ورود» به API، باید هزینه‌های «عدم ورود» به API را نیز برای سازمانتان محاسبه کنید. ببینید که در دنیای پساتحول، سازمان شما چه جایگاهی خواهد داشت؟

نگاه راهبردی به آینده‌ی سازمان

آنچه در این مقطع باید انجام دهید، یک تحلیل راهبردی کلان از آینده‌ی زنجیره‌ی ارزش سازمان است. برای شروع می‌توانید یک نقشه از تمام ذی‌نفعان در سطح ۱ و ۲ تهیه کنید و کانال‌های ارتباطی موجود با این ذی‌نفعان را تصور کنید.

در گام بعد، فرض کنید که یک دنیای تماماً خودکار و پلتفرمی، این زنجیره‌ی ارزش به چه شکلی متحول می‌شود؟ اگر تحول خاصی شناسایی نکردید، می‌توانید گزینه‌ی API را کنار بگذارید.

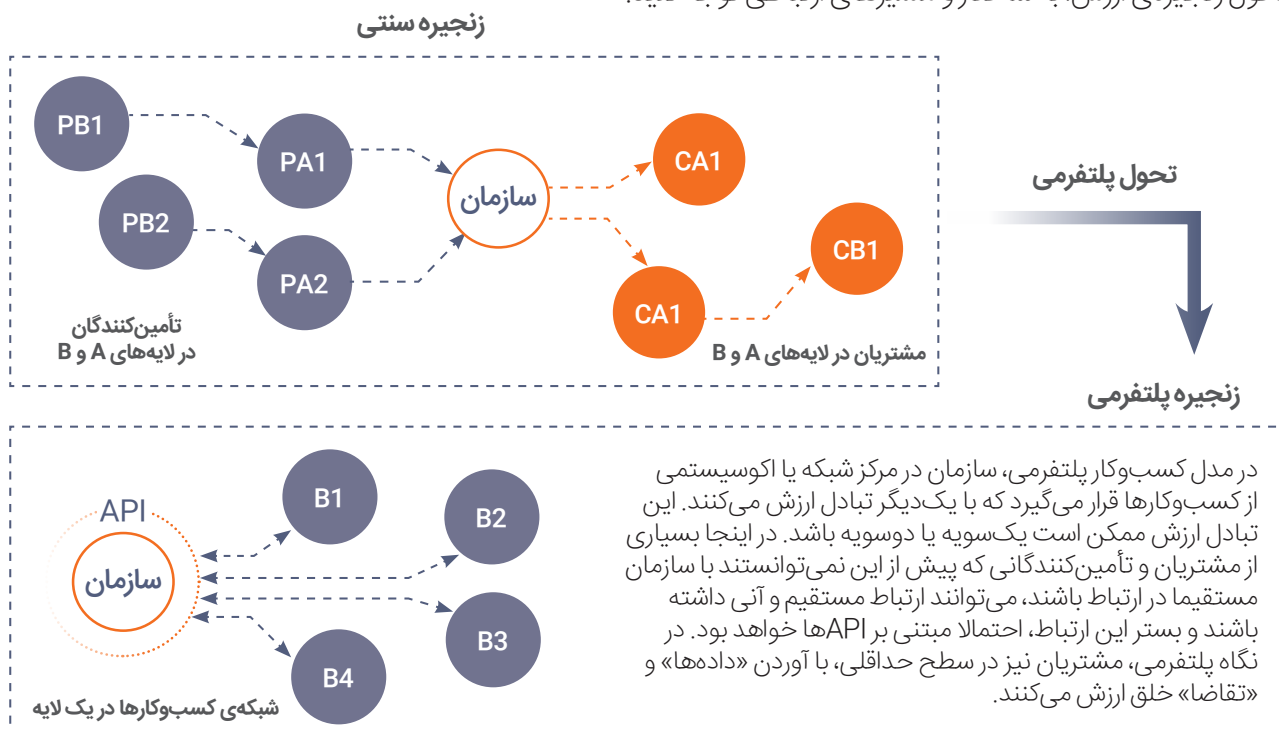
آن‌طور که دیدیم، این مسیر بسیار پر پیچ و تاب و دشوار است و شاید سازمان‌ها ترجیح بدهند که عطای آن را به لقایش ببخشند. اما این تمام ماجرا نیست.

واقعیت این است که خواه ناخواه، API ها ستون تحول دیجیتال سازمان شما هستند و در تحول شما از یک کسب‌وکار سنتی به یک پلتفرم، نقشی کلیدی ایفا می‌کنند.

همان‌طور که گفتیم، API ها ابزاری جدید برای «انتقال ارزش» محسوب می‌شوند. این ابزار، شبکه‌های تجاری دیجیتال آینده را شکل می‌دهد و اگر تخصص کافی در آن نداشته باشید، از قافله عقب می‌مانید. به عبارت دیگر، در دنیایی که شبکه‌های مختلف کسب‌وکاری از طریق API به یک‌دیگر متصل می‌شوند و با این زبان با یک‌دیگر صحبت می‌کنند، سازمان‌هایی که نتوانند از این طریق با یک‌دیگر ارتباط بگیرند، از شبکه‌ها بیرون می‌مانند.

شکل ۸

تحول زنجیره‌ی ارزش؛ به ساختار و مسیرهای ارتباطی توجه کنید.





رویکرد چابک در توسعه API

رویکرد چابک چیست و چطور در حوزه‌ی API از آن استفاده کنیم؟

آشنایی اولیه داشته باشند. مشکلی که در بسیاری سازمان‌ها مشاهده می‌شود، عدم آشنایی مدیران با این رویکردهاست و متأسفانه باید گفت که گاهی روح رویکرد چابک، در میان اصطلاحات تخصصی و نامفهوم گم می‌شود.

سازمان‌ها می‌کوشند تا یکی از چارچوب‌های چابک نظیر «اسکرام» را به ساختارهای سازمانی سنتی خود «بخوراند» و حاصل کار، چیزی نیست جز یک ترکیب ناهم‌خوان از نظام‌های بروکراتیک که روی سر هم سوار شده‌اند.

فقدان تجربه در مورد رویکرد چابک، یکی از مهم‌ترین دلایل شکست سازمان‌ها در اتخاذ این رویکرد است. سازمان‌هایی که به طور ناگهانی قصد می‌کنند به سراغ این رویکرد بروند و پروژه‌های بزرگی را با رویکرد چابک مدیریت کنند، معمولاً با شکست مواجه می‌شوند.

حال چه باید کرد؟

اگر شرکت شما یک مجموعه‌ی نرم‌افزاری است، قطعاً بسیار بیش از این و پیش از این با رویکردهای چابک آشنایی دارید. آنچه در ادامه می‌آید، راهکارهایی برای سازمان‌ها و مدیرانی است که با این رویکرد آشنایی ندارند. در ادامه، چند راهکار در این زمینه پیشنهاد می‌شود.

کوچک شروع کنید. در ابتدای امر می‌توانید یک تیم حداکثر پنج نفره را با این رویکرد مدیریت کنید. بعد از گذشت چند هفته و مشاهده‌ی برون‌دادها و چالش‌ها، می‌توانید برای بهبود شرایط کاری تیم و توسعه‌ی آن، قدم‌های بعد را بردارید. از سوی دیگر، با کوچک شروع کردن، می‌توانید اطمینان حاصل کنید که اگر در این مسیر شکست خوردید، پیامدهای گسترده‌ای در سازمان شما نخواهد داشت.

به سراغ مشاور بروید. صدها «نسخه» برای رویکرد چابک وجود دارد و نسخه‌ای واحد برای تمام سازمان‌ها وجود ندارد. مشاوران مدیریت چابک می‌توانند در این مسیر با شما همراه شوند، نیازمندی‌های سازمان شما را شناسایی کرده و با دانش تخصصی خود، چارچوبی مناسب برای شما توصیه کنند.

این پرسش که «چگونه باید فرآیند توسعه‌ی نرم‌افزار را مدیریت کرد؟» سال‌هاست که ذهن کسب‌وکارهای حوزه‌ی فناوری اطلاعات را به خود مشغول کرده است. این پرسش، پاسخ یکتایی ندارد، اما از دو دهه پیش، نگاهی جدید به نام نگاه «چابک» یا Agile به این حوزه راه یافته است.

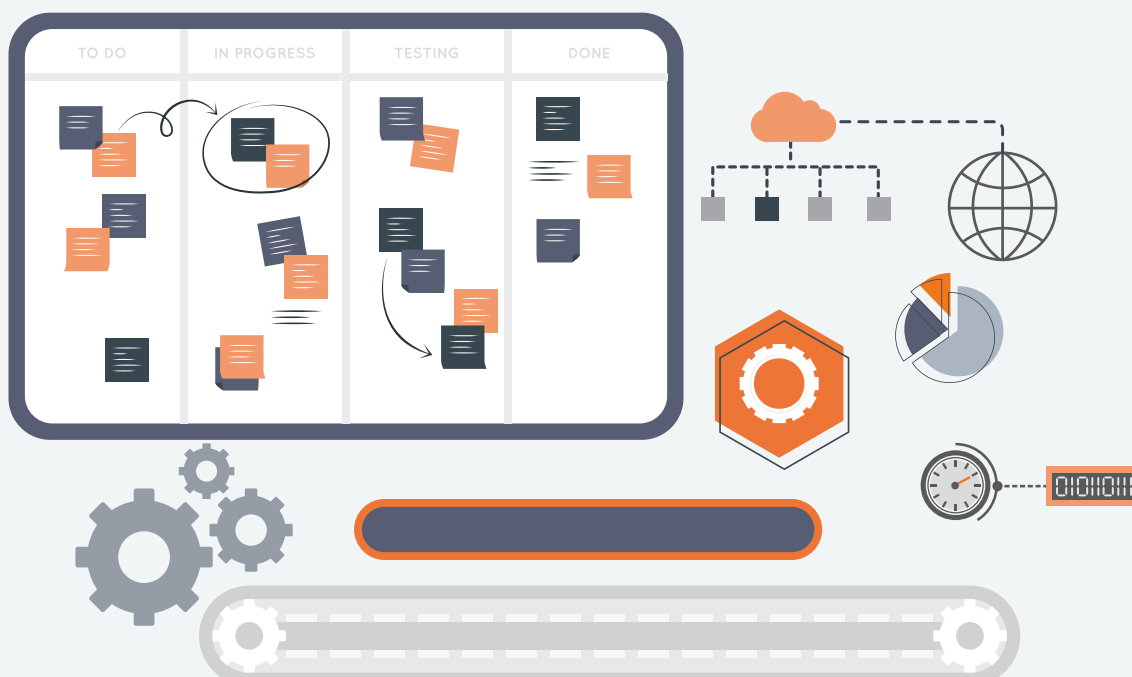
اوایل سال ۲۰۰۱ بود که عده‌ای از پیش‌تازان حوزه‌ی نرم‌افزار در یک آخر هفته گرد هم آمدند. آن‌ها از فرآیندهای کند و زمان‌بر خسته شده بودند و احساس می‌کردند که ارزش‌های عمیق نهفته در توسعه‌ی نرم‌افزار، در هیاهوی بروکراسی سازمانی گم شده است. آن‌ها در پایان این گردهمایی، این مانیفست را منتشر کردند که به نام «مانیفست چابک» شناخته می‌شود و تنها در ۶۸ کلمه، روح حاکم بر رویکرد چابک را بیان می‌کند:

«ما در حال کشف راه‌هایی بهتری برای توسعه‌ی نرم‌افزار هستیم. ما در این مسیر، در کنار توسعه‌ی نرم‌افزار، به دیگران کمک می‌کنیم تا چنین کنند. ما برای تحقق این کار، ارزش‌هایی را در اولویت قرار داده‌ایم:

۱. افراد و تعاملات مهم‌تر از فرآیندها و ابزارها هستند.
۲. نرم‌افزاری که کار کند از مستندات گسترده، مهم‌تر است.
۳. همکاری و هماهنگی با مشتری از مذاکرات قرارداد مهم‌تر هستند.
۴. پاسخ به تغییر، از دنبال کردن یک طرح مصوب مهم‌تر است. درست است که موارد سمت چپ مهم هستند، اما موارد سمت راست برای ما اهمیت بیشتری دارند.»

این مانیفست تنها روح حاکم بر رویکرد چابک است و پس از آن، چارچوب‌های متنوعی بر پایه‌ی این مانیفست ساخته شد. نام‌هایی نظیر «اسکرام» یا «کانبان» یا... در امتداد همین مسیر می‌آیند و چارچوب‌هایی هستند که بر این مبنا توسعه یافتند.

هدف ما از مطرح کردن این مانیفست در اینجا، این نیست که آموزش بنیادی رویکردهای چابک را در پیش بگیریم. بلکه می‌خواهیم نگاهی گذرا به آن‌ها داشته باشیم تا مدیران با آن‌ها



دهید. برنامه‌نویس‌ها علاقه‌ی وافری به کار در محیط آکواریومی دارند. بسیاری از نرم‌افزارها مادامی که توسط مشتری به بوته‌ی آزمایش گذاشته نمی‌شوند، بسیار زیبا، پایدار و کارآمد به نظر می‌رسند. اما آنچه واقعاً مسیر توسعه را مشخص می‌کند، بازخورد مشتری است. این مشتری آزمایشی که در ابتدای مسیر انتخاب می‌کنید، باید صبر و حوصله‌ی زیادی داشته و به سازمان شما اعتماد داشته باشد. در نتیجه به سراغ یک سازمان «آشنا» یا «متحد» بروید و از آن‌ها بخواهید نقش مشتری آزمایشی را برای محصول شما ایفا کنند. بی‌اغراق می‌توان گفت که بزرگ‌ترین اشتباه سازمان‌ها این است که پس از ماه‌ها و حتی سال‌ها توسعه، ناگهان و یک‌باره محصول خود را در معرض بازار قرار می‌دهند و انتظار دارند که بازار به اندازه‌ی تمام تلاش سازمان، برای محصول یا خدمت ارائه‌شده ارزش قائل باشد. اتفاقی که متأسفانه رخ نمی‌دهد؛ اگر نگوییم هرگز، باید بگوییم به ندرت.

تحول را همه‌جانبه بنگرید. وقتی گفتمان جدیدی برای مدیریت تعریف می‌کنید، باید آن را «تمام و کمال» بپذیرید. ابزارها، فرآیندها، رویکردها و هرچه سازمان را به گذشته پیوند می‌زند، کنار بگذارید. وگرنه سرنوشت «شترگاوپلنگ» در انتظار سازمان شماست. در جایگاه پیش‌بینی نمی‌توان گفت که اتخاذ رویکرد چابک دقیقاً با کدام ساختار یا فرآیند یا فرهنگ در سازمان شما ناسازگار خواهد بود. این ناسازگاری‌ها در عمل آشکار می‌شوند و باید در مقابل آن‌ها قاطعانه تصمیم بگیرید. یا چابک، یا سنتی.

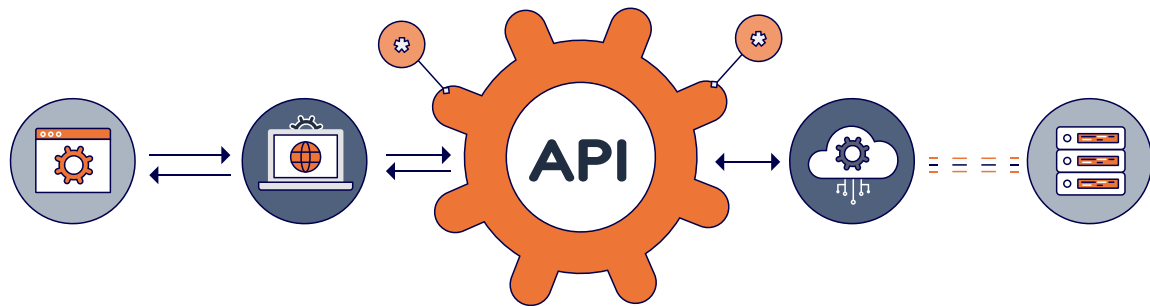
تله‌ی هزینه‌ی از دست‌رفته را در نظر داشته باشید. یکی از تله‌های مهم تصمیم‌گیری که نوآوری و چابکی را در نطفه خفه

البته دانش تخصصی مدیریت چابک ممکن است در سازمان شما وجود داشته باشد و تیم‌های شما با آن آشنا باشند. اما معمولاً این دانش در اختیار همکاران فنی است و مشاوران می‌توانند پل ارتباطی میان تیم فنی و بدنه‌ی مدیریتی سازمان باشند و نیازمندی‌های چارچوب چابک را به سطوح بالاتر انتقال دهند.

شاخص‌های جدید تعریف کنید. بسیاری از نظام‌های سنجش سنتی سازمان‌ها، نمی‌توانند پاسخگوی نیازهای توسعه‌ی نرم‌افزار باشند. معمولاً در نظام‌های سنجش کسب‌وکاری، شاخص‌های درازمدتی ترسیم می‌شوند که سپس با شکست آن‌ها به فصول یا ماه‌ها، می‌توان اهداف کوتاه‌مدت را ترسیم کرد. اما در توسعه‌ی نرم‌افزار ممکن است اهداف این دوره اساساً با اهداف دوره‌ی بعد تفاوت داشته و شاخص‌های متفاوتی را در بر بگیرند. این امر در ابتدا ممکن است برای همکاران کسب‌وکاری سازمان‌ها جدید و حتی نامعقول باشد، اما بخشی از طبیعت توسعه‌ی چابک نرم‌افزار است.

به رهبری توجه کنید. به اصول چابک بنگرید: «افراد و تعاملات» از فرآیندها و ابزارها مهم‌تر هستند. پیام این جمله برای سازمان‌ها چیست؟ یک تیم چابک بیش از فرآیندها با تکیه بر «رهبری» اداره می‌شود. در نتیجه، در انتخاب رهبر برای این تیم‌ها باید دقت دوچندانی داشته باشید. چرا که فرآیندها نمی‌توانند جور رهبری ضعیف را بکشند.

منظر مشتری را از یاد نبرید. سعی کنید به طور پیوسته، خروجی‌های تیم‌های چابک را در معرض تماس مشتریان قرار



در واقع، پس از صحبت‌های اولیه و درک زیبایی‌های رویکرد چابک، شما با انبوهی از کلمات و اصطلاحات تخصصی مواجه می‌شوید یا جایگاه‌های شغلی جدیدی را می‌بینید که نظام منابع انسانی سازمان شما با آن آشنایی ندارد. عباراتی مانند: اسپرینت، مالک محصول، بک‌لاگ، اسکرام‌مستر و... این‌ها مربوط به همان دانش تخصصی و فرآیندی هستند که به کمک یادگیری و به‌کارگیری متخصصان حوزه‌ی چابک می‌توانید به سازمان خود بیافزایید.

توسعه‌ی چابک API

در نهایت، API یک زیرساخت نرم‌افزاری است و روش‌های چابک، مانند هر پدیده‌ی نرم‌افزاری دیگری می‌توانند به آن اعمال شوند. آنچه در این مسیر اهمیت پیدا می‌کند، همگام‌سازی مسیر توسعه‌ی فنی API با آن چیزی است که در گام‌های قبل مطرح شد.

برای مثال، هنگامی که یک سناریوی کسب‌وکاری برای یک API طراحی شد، می‌توان به سرعت یک نمونه‌ی اولیه (پروتوتایپ) از آن API را تهیه کرد و برای یک آزمون اولیه‌ی کسب‌وکاری در اختیار مشتری آزمایشی قرار داد. بدین ترتیب می‌توان حلقه‌های بازخورد کوتاهی را شکل داد که می‌تواند بینش عمیقی را به راهبرد کلان سازمان تزریق کند.

به عبارت دیگر، چالش توسعه‌ی چابک API در این نیست که متد اسکرام را چگونه به توسعه‌ی API اعمال کنیم؟ سوال اصلی این است که چطور از ظرفیت‌های رویکرد چابک، در کسب‌وکار API خود بهره بگیریم و پاسخ به این سوال این است:

ایده‌های کسب‌وکاری خود را به سرعت پیاده‌سازی کرده و از بازار بازخورد بگیرید. اولین محصول حداقلی که «کار» می‌کند را به بازار عرضه کنید و ببینید چه اتفاقی می‌افتد؟ یک کمپین بسیار کوچک بازاریابی (حتی برای سرویسی که وجود ندارد) برگزار کنید و ببینید آیا تلفن‌ها به صدا در می‌آیند؟

می‌کند، تله‌ی هزینه‌های از دست رفته است. این تله اشاره به مواقعی دارد که هزینه‌ی قابل‌توجهی برای یک هدف اشتباه صورت گرفته است، اما سازمان (یا فرد) در برابر پذیرش بیهودگی این هزینه و تلاش، مقاومت می‌کند. در نتیجه، در مسیر قبلی به حرکت ادامه می‌دهد تا حداقل به یک خروجی قابل ارائه دست یابد. بیایید مثالی بزنیم: شما قرار است خط تولید رب گوجه تأسیس کنید، اما در میانه‌ی راه متوجه می‌شوید که باید مربا تولید کنید. بسیاری از سازمان‌ها، اگر ماشین‌آلات مورد نیاز رب گوجه را خریده باشند، دشوار می‌توانند اشتباه خود را بپذیرند و زبان را در لحظه متوقف کنند. آن‌ها خط تولید رب گوجه را تأسیس می‌کنند و تلاش می‌کنند تا سناریوی کسب‌وکاری را با تصمیم خود، یعنی کارخانه‌ی رب‌سازی هم‌خوان کنند. رویکرد چابک، دقیقاً در مقابل این مسیر قرار دارد. شما باید بتوانید به سرعت به تغییرات پاسخ دهید، حتی اگر به معنای دور ریختن بسیاری از تلاش‌های پیشین باشد.

با این کلمات خارجی چه کنیم؟

رویکرد چابک از دور بیشتر به یک «تحول فرهنگی» می‌ماند تا یک تحول فرآیندی و تخصصی. باید گفت که رویکرد چابک برای مدیران بالا دست جنبه‌ی «فرهنگی» و برای مدیرهای عملیاتی جنبه‌ی «فرآیندی» و «تخصصی» دارد. اما منظور از این جنبه‌ی تخصصی چیست؟

فرض کنید که می‌خواهید روش «اسکرام» را در سازمان خود پیاده‌سازی کنید. این روش، نه تنها بر روح مانیفست چابک تکیه می‌کند، بلکه ساختار سازمان، ساختار برنامه‌ریزی، ساختار تعامل، شرح شغل و حتی تکنیک‌های برگزاری جلسات را نیز در اختیار مدیران می‌گذارد. در نتیجه باید گفت که پیاده‌سازی یک رویکرد چابک در سطح عملیاتی، علاوه بر تحول فرهنگی، نیازمند دانش تخصصی و فرآیندی است. این دانش را همان‌طور که گفته شد، باید از متخصصان آن جویا شوید (مثلاً به سراغ مشاور بروید).

شن‌بازی با API ها

«سندباکس» چیست و چه کاربردی در توسعه API دارد؟



به عبارت دیگر، مشتری شما باید چند بیمه‌ی شخص ثالث (که احتمالاً هر یک قیمتی در حدود دو میلیون تومان دارد) از شما خریداری کند تا پیاده‌سازی خدمت تکمیل شود. در این شرایط، بهتر است یک محیط آزمایشی برای دریافت API طراحی کنید تا مشتریان بتوانند در آن محیط، تست و پیاده‌سازی سرویس را انجام دهند.

به همین طریق، برای ارائه‌ی نسخه‌های آتی API های خود نیز می‌توانید از محیط‌های «سندباکس» استفاده کنید. مثلاً نسخه‌ی دوم API ها را در محیط غیرعملیاتی در اختیار همکاران قرار دهید تا آن‌ها را تست کنند.

تأکید بر این مفهوم به این علت است که ممکن است همکاران فنی شما در معماری خود، این مهم را از یاد ببرند. اما به خاطر داشته باشید که این امر برای شما در کنار اهمیت فنی، تبعات کسب‌وکاری جدی خواهد داشت.

یکی از مفاهیم بسیار مهم در توسعه‌ی نرم‌افزار، مفهوم سندباکس است. این مفهوم، به دلیل اینکه «معنای کسب‌وکاری» دارد، با بسیاری از مفاهیم نرم‌افزاری دیگری که از زبان توسعه‌دهندگان می‌شنوید، مجزا است.

«سندباکس» به زبان فارسی یعنی همان فضای پر از ماسه که کودکان با اسباب‌بازی‌هایشان در آن مشغول می‌شوند و قلعه‌های شنی برای خود می‌سازند. در فضای نرم‌افزار، همواره یک «محیط آزمایشی» برای توسعه‌ی نرم‌افزارها در نظر گرفته می‌شود و تغییرات، پیش از اعمال در محیط‌های عملیاتی، در این فضا مورد آزمایش قرار می‌گیرند.

حال چرا در یک مستند کسب‌وکاری این قدر روی این مفهوم فنی تأکید می‌کنیم؟ فرض کنید شما API بیمه‌ی شخص ثالث را در اختیار کسب‌وکارها قرار داده‌اید. اکنون یک کسب‌وکار قصد دارد با دریافت این سرویس از شما، یک درگاه آنلاین فروش خدمات بیمه ایجاد کند. این شرکت ممکن است در طول فرآیند پیاده‌سازی سرویس‌ها نیازمند ۲۰ مرتبه فراخوانی سرویس باشد و حداقل نیاز دارد چند سفارش موفق ثبت کند.

کلام آخر

نگاهی به تمام آنچه گفته شد

به پایان این دفتر رسیدیم. در اینجا می‌کشیم تا با یک مرور کلی بر تمام آنچه در این مستند دیدیم، یک چک‌لیست راهبردی برای مسیر API سازمان شما فراهم کنیم:

جدول ۲. چک‌لیست راهبردی API ها

گام			پرسش‌ها، گزینه‌ها، ملاحظات و خروجی‌ها
ارزیابی درونی	انتخاب نوع API:	زیرساخت‌های موجود	الزامات فرادستی (بایدها و نبایدها):
	۱. استعلامی (دیتا) ۲. خدمات مبتنی بر API	۱. منابع انسانی ۲. سخت‌افزار ۳. نرم‌افزار	۱. رگولاتوری ۲. مأموریت سازمان ۳. راهبرد سازمان
ارزیابی بازار	تعداد و فرکانس فراخوانی:	تعداد مشتری:	قدرت چانه‌زنی:
	۱. API پذیر ۲. API ناپذیر	۱. کمتر از ۱۰ (شبکه‌ی همکار) ۲. بیشتر از ۱۰ (رویکرد باز)	۱. بازار رقابتی ۲. بازار انحصاری
گزینش راهبرد API	انتخاب راهبرد:	تأمیل مشتری به پرداخت:	بر مبنای:
	۱. برون‌سپاری به پلتفرم ۲. عرضه‌ی مستقیم ۳. مالکیت پلتفرم ۴. ترکیبی	۱. روال و سناریوی استاندارد سرویس ۲. بلوغ سازمان ۴. جایگاه API در چشم‌انداز و راهبرد	۱. تعداد مشتری در مقابل عملیاتی ۲. روال و سناریوی استاندارد سرویس ۳. بلوغ سازمان ۴. جایگاه API در چشم‌انداز و راهبرد
طراحی سبد API ها			۱. تدوین فهرست API ها ۲. تعیین پارامترهای ورودی و خروجی هر API ۳. رهنگاشت عرضه‌ی API (گانت چارت)
راهبرد قیمت‌گذاری سرویس	API های درآمدزا:	ارزش راهبردی API رایگان:	مبنای قیمت‌گذاری:
	۱. رایگان ۲. فریمیوم ۳. غیررایگان	۱. دیتا ۲. رقابت ۳. اکوسیستم	۱. رقابت ۲. مطالعات بازار ۳. آزمون بازار
ساختار قیمت‌گذاری	مدل کلی:	شیوه‌ی پرداخت:	محصول و خدمت:
	۱. پرداخت به ازای استفاده ۲. خرید بسته ۳. مدل ترکیبی	۱. پیش‌پرداخت ۲. پس‌پرداخت	۱. روزانه ۲. هفتگی ۳. ماهیانه
ارزیابی مالی و راهبردی			۱. نگارش برنامه مالی ورود (هزینه‌های ورود به API) ۲. نگارش برنامه‌ی عدم ورود (ارزیابی پیامدهای راهبردی) ۳. ارزیابی سناریوهای مرزی (ورود ناموفق)
کاربست رویکرد چابک			۱. کوچک شروع کنید ۲. به سراغ مشاور بروید ۳. شاخص‌های جدید تعریف کنید ۴. به رهبری توجه کنید ۵. منظر مشتری را از یاد نبرید ۶. تحول را همه‌جانبه بنگرید ۷. هزینه‌ی فرصت را از یاد نبرید ۸. روال‌های تست و آزمایش (سندباکس) داشته باشید



پادیوم به عنوان بخشی از مأموریت خود در حوزه‌ی توسعه اکوسیستم و گفتمان‌سازی، اسناد و گزارش‌های موسسات معتبر بین‌المللی را که به صورت آزاد منتشر شده‌اند، به صورت رایگان در اختیار فعالان اکوسیستم دیجیتال ایران قرار می‌دهد.

این گزارش‌ها منابع مناسبی برای آشنایی با مفاهیم بانکداری باز و نقش API در اکوسیستم مالی هستند. تا کنون دو گزارش از این دست توسط پادیوم منتشر شده که نسخه‌ی دیجیتال آن‌ها در کانال تلگرام پادیوم در دسترس است.

۱. گزارش «[بانک‌ها چگونه می‌توانند در اقتصاد API شکوفا شوند؟](#)» که توسط موسسه‌ی اکسچنچر منتشر شده است. در این گزارش، به چگونگی تحول درونی بانک‌ها در مسیر تحول به سمت بانکداری باز و API محور پرداخته می‌شود. در این گزارش می‌خوانیم که چطور ارکان تکنولوژی، حاکمیت سازمانی و... دست به دست هم می‌دهند تا بانک‌ها سفر دیجیتال خود را با موفقیت سپری کنند.

۲. گزارش «[گذار به مدل کسب‌وکار پلتفرمی در بانکداری](#)» که توسط موسسه‌ی PWC منتشر شده است. این گزارش به چپستی مدل کسب‌وکار پلتفرمی در بانکداری می‌پردازد و نشان می‌دهد مدل کسب‌وکار بانک‌ها، مدل درآمدی آن‌ها و همچنین تجربه‌ی مشتری چگونه در بانکداری پلتفرمی متحول می‌شوند.

برای مطالعه بیشتر، [لینک‌دین پادیوم](#) را دنبال کنید.



با ما در ارتباط باشید

مهدی ملکی‌فر
مدیر توسعه‌ی کسب‌وکار
۰۹۱۲۲۳۰۳۰۱۵



عاطفه موفقی
راهنمای کسب‌وکار
۰۹۳۶۶۵۸۹۳۵۴



شهروزه مقدم
مدیر امور مشتریان
۰۹۱۲۵۴۸۴۵۳۶



مهدی علیشیر
مدیر اجرایی
۰۹۱۷۰۶۵۳۲۲۰



پست الکترونیک: Info@podium.ir
تلفن تماس: ۰۲۱۸۹۵۱۸۹۸۹
نشانی: تهران، پردیس، پارک فناوری پردیس
بلوار نوآوری ۱۲، پلاک ۱۲۳