

کلا توی طراحی محصول یک پروسه ی همیشگی که طی میشه به این صورته که یک ایده ( به هر طریقی ) مطرح میشه ، روی اون ایده فکر میشه که ایا قابل اجرا هست یا نه ، بعد محصول طراحی میشه و بعد میره برای توسعه ، بعد از اون تست ازش گرفته میشه و در نهایت محصول ارائه میشه .

اما در واقعیت اکثر مواقع روندی که طی میشه چرخه ای هست نه خطی . خیلی از مراحل هم ارتباط دو طرفه دارن و روی هم تاثیر میزارن . برای مثال توی مرحله ی اول یک ایده مطرح میشه و بعد از اینکه اعتبار سنجی شد ، دیزاین میشه ، اما در حین دیزاین متوجه یک سوال جواب داده نشده میشن و همون باعث آپدیت شدن ایده میشه . یا ممکنه رد مرحله ی توسعه به ی روند کاربری بر بخوریم که باعث بشه دیزاین رو تغییر بدیم .

گاهی هم تمام این مراحل رد میشه و زمانی که داریم دیتا های سیستم و رفتار کاربر ها رو بررسی میکنیم باعث بشه دوباره توی هر کدوم از مراحل قبلی یک سری آپدیت ایجاد کنم . توی اکثر شرکت ها هم خیلی از مراحل تقریبا موازی پیش میرن . برای مثال در حالی که تیم مارکتینگ داره برای محصولی که تا الان توسعه داده شده ، تبلیغ میکنه ، توسعه دهنده ها در حال توسعه بخش بعدی باشن ، تیم دیزاین سراغ دیزاین فیچر بعدی بره و پروداکت لید هم روی داده های که تا الان جمع اوری شده و چشم انداز مرحله ی بعدی پروژه تمرکز کنه .

هر کدوم مرحله خودشون جزییات بیشتری دران و همیشه یک روند خطی طی نمیشه ی اکثر شرکت ها علاوه بر این روند چرخه ای برای هر بخش از محصول ، خیلی از مراحل برای بخش های جدا تقریبا موازی پیش میرن . برای مثال در حالی که تیم مارکتینگ داره برای محصولی که تا الان توسعه داده شده ، تبلیغ میکنه ، توسعه دهنده ها در حال توسعه بخش بعدی باشن ، تیم دیزاین سراغ دیزاین فیچر بعدی بره و پروداکت لید هم روی داده های که تا الان جمع اوری شده و چشم انداز مرحله ی بعدی پروژه تمرکز کنه .

اما بریم سراغ جزییات هر بخش :

بخش اول - از یک ایده تا رسیدن به یک طرح پیشنهادی

ایده میتونه به صورت های مختلفی مطرح بشه :

گاهی یک نفر به صورت کلی یک ایده رو بیان میکنه . برای مثال میگه " من دوست دارم اپی داشته باشم که شب ها قبل اینکه بخوابم از من بخواد تا کار هایی که در طول روز انجام دادم رو جمع بندی کنم و برای فردا برنامه ریزی کنم ."

این یک ایده ی خیلی کلی هست . توی این حالت باید اول درباره ی این ایده تحقیق بشه : شاید همین الان اپی وجود داشته باشه ، در حال حاضر افراد از چه طریقی این مشکل رو حل میکنن ؟ و ...

بعد از اون باید اعتبار سنجی بشه : حتی اگر این اپ ساخته بشه ایا اونقدر به کاربران ارزش اضافه میکنه که حاضر باشن از این برنامه استفاده کنند ؟

یک ریسکی که این حالت ایده داره اینکه که مشکلی که در ابتدا بیان شد و براساس اون ایده شکل گرفت ، اصلا مشکل واقعی نباشه

به این حالت از ایده میگویند : business driven

حالت بعدی بر اساس کاربر ها هست (customer driven) و توی شرکت هایی که با تجربه تر هستن بیشتر استفاده میشه . توی این حالت ، ایده از یک نیازی که مشتری خودش بیان کرده شکل میگیره . برای مثال ممکنه شرکت یک محصول برای رزرو تور و بلیط داشته باشه و بعضی از کاربر ها توی بخش نظرات یا از طریق پشتیبانی این رو بگن که تور نیاز ندارن و اما دوست دارن که یک راهنمای محلی پیدا کنن تا مکان های مدنظرشون رو با توجه به سرعت و علاقه مندی خودشون با یک راهنما ببینن . همین میتونه ایده ی شروع یک محصول مستقل جدید باشه و یا به عنوان یک فیچر به محصولی که همین الان وجود داره اضافه بشه .

نوع بعدی data driven هست . این حالت بیشتر برای محصولاتی به کار میره که همین الان هم وجود دارن و میشه ازشون دیتا استخراج کرد . مثلاً دیتا میگویند 40000 کاربر وارد مرحله ی نهایی پرداخت شدند ، اما 1000 نفر پرداختشون رو نهایی نکردن . اینجا این دیتا ها در واقع دارن یک سیگنال واضح از مشکل موجود میدن ، که مرحله ی بعدی باید مراحلی که کاربر طی میکنه بررسی بشه ، بر اساس اون یک فرضیه برای حل این مشکل بیان بشه و بعد اعتبار سنجی بشه .

حالت نهایی هم هم بر اساس نیاز بازار هست . گاهی شرکت رقیب محصولی ارائه میده . برای خارج نشدن از چرخ رقابت ، شما هم باید محصولی مشابه اون و با کیفیت بالاتر ارائه بدی . نمونه ی خیلی مطرح این موضوع رقابت بین شرکت های اپل و سامسونگ برای ارائه گوشی جدید هست .

در نهایت بعد از اینکه یک ایده شکل گرفت وارد مرحله ی بعدی میشیم : آیا اصلاً این محصول ارزش کافی برای ساخت داره ؟

اینجا میایم چشم اندازی که برای محصول تا 5 سال آینده میبینیم ، کاری که الان باید انجام بشه و اینکه قراره به چه هدفی برسیم رو مشخص میکنیم .

بعد از اینکه این موارد مشخص باید اولویت ها مشخص بشه . ایده ها خیلی قابلیت بسط و گسترش دارن اما بیشتر از ظرفیت تیم در اون لحظه باشه . توی این مواقع یک product lead با یکسری از فریم ورک ها تصمیم میگیره چه فیچر هایی الان توی اولویت هستن .

مثلا ممکنه از فریمورک RICE استفاده کنه :

Reach × Impact × Confidence

-----  
Effort

یا از مدل MoScOW :

What features we :

Must have

Should have

Could have

Won't have

.  
بعد از اون با دسته بندی موارد میام PRD محصول رو مینویسیم . ( شامل مشکل ، هدف ، یوزر استوری ، نیازمندی ها و معیار های پذیرش هر نیازمندی )

بعد از اینکه ایده صرفا از ایده ی خام بودن تبدیل شد به موجودیتی که میشه اون رو دقیق بیان کمرد ف برای اون حوضه ها و فضای رشدش رو مشخص کرد و معیار های اندازه گیری دقیق اون معلوم شد ، میریم سراغ مرحله ی بعدی که دیزاین هست .

مرحله ی دیزاین توی این مرحله از PRD استفاده میشه تا یوزرفلو ها مشخص بشه ، بعد از ان وایر فریم و بعد هم دیزاین . اکثر از فیگما برای مراحل دیزاین استفاده میشد اما این یکی دو سال اخیر ، ابزار های مثل figma make , claude design و ... واقعا دارن کارایی خودشون رو نشون میدن و یکی از رقیب هایی جدی figma در حال حاضر Framer هست که اون هم بسیار قدرتمنده .  
بعد از اون هم وارد فاز توسعه محصول میشیم .

بعد از اینکه دیزاین نهایی شد ، این خروجی به همراه PRD به تیم توسعه تحویل داده میشه . اینجا اولین کاری که انجام میشه اینه که تیم توسعه میاد کار رو تیکه تیکه میکنه ، یعنی هر نیازمندی رو تبدیل به یک سری تسک کوچیک تر میکنه که قابل تخمین زدن باشه . معمولا از یک سیستم مثل Jira یا Linear استفاده میشه تا این تسک ها track بشن و مشخص بشه هر کدوم دست کیه و توی چه مرحله ای هست .

توی این مرحله هم باز همون حالت چرخه ای که اول گفتم پیش میاد . خیلی وقت ها دیزاینر یک فلو رو طراحی کرده که از نظر بصری قشنگه اما از نظر فنی یا خیلی سخته پیاده سازیش یا اصلا با محدودیت های بک اند

همخونی نداره . اینجا یک بحث بین دیزاینر و دولوپر شکل میگیره که بهش میگن design-dev handoff و خیلی وقت ها همینجا یک سری چیزا توی دیزاین تغییر میکنه یا یک راه حل جایگزین پیدا میشه .

توسعه به دو بخش frontend و backend تقسیم میشه که به صورت موازی پیش میرن . توی تیم های بزرگ تر ممکنه حتی یک تیم جدا برای infrastructure یا DevOps هم باشه که مسئول اینه که سرور ها ، دیتابیس و کل زیرساختی که قراره این فیچر روش سوار بشه رو آماده کنه .

یک نکته ی مهم اینجا استفاده از version control ( معمولا Git ) هست . هر دولوپر روی یک branch جدا کار میکنه و بعد از تموم شدن کار ، کدش رو از طریق یک pull request برای بقیه ی تیم میفرسته تا review بشه . این review هم خودش یک لایه از کنترل کیفیته که قبل از اینکه کد وارد نسخه ی اصلی بشه ، یک نفر دیگه هم کد رو چک کنه .

بعد از اینکه توسعه یک فیچر تموم شد ، میره برای تست . تست هم خودش چند لایه داره : اول unit test دولوپر خودش مینویسه تا مطمئن بشه هر تیکه از کد درست کار میکنه . بعد integration test که چک میکنه این تیکه ها با هم درست کار میکنن یا نه . و در نهایت QA مییاد و از دید یک کاربر واقعی محصول رو تست میکنن ، دنبال باگ میگردد ، سناریو های عجیب و غریب رو امتحان میکنه که شاید دولوپر اصلا بهش فکر نکرده بوده . یک نمونه که برای خودم اتفاق افتاد ، ( در شرکت قبلی من ui/ux دیزاینر بودم اما اختصاصا تست یک محصول هم برعهده داشتم ) بر تست یکی فیچر search از کیبورد عربی به جای فارسی استفاده کردم و اینجا مشخص شد خیلی از شهر ها به خاطر کد ASCII متفاوتی توی یک سری از حرف هست ، سرچ نمیشد . مثلا شهر " کرج " شاید در ظاهر توی کیبورد فارسی و عربی یکسان نوشته بشه ، اما در فارسی " ک " ساده هست و توی عربی " ك " نوشته میشه و کد ASCII متفاوتی دارند .

توی خیلی از شرکت ها قبل از اینکه فیچر برای همه کاربرا release بشه ، اول برای یک گروه کوچیک از کاربر ها منتشر میشه که بهش میگن beta test یا soft launch . این کار باعث میشه اگه یک مشکل جدی وجود داشته باشه ، قبل از اینکه به همه کاربر ها آسیب بزنه ، شناسایی بشه . یک روش دیگه هم که خیلی رایجه A/B testing هست ، یعنی دو نسخه از یک فیچر رو همزمان برای دو گروه مختلف از کاربرا منتشر میکنن و بعد رفتارشون رو با هم مقایسه میکنن تا ببینن کدوم نسخه بهتر جواب داده .

بعد از این مراحل ، محصول آماده ی release نهایی میشه . اینجا هم یک هماهنگی بین چند تیم لازمه : تیم مارکتینگ باید محتوای تبلیغاتی رو آماده کرده باشه ، تیم پشتیبانی باید بدونن این فیچر جدید چیه تا بتونه به سوالاتی کاربرا جواب بده ، و تیم DevOps هم باید مطمئن بشه که زیرساخت آماده ی این حجم از ترافیک جدید

اما ماجرا اینجا تموم نمیشه . بعد از release ، پروداکت لید و تیم دیتا میان و شروع میکنن به بررسی رفتار کاربر ها با این فیچر جدید : چند درصد از کاربرا ازش استفاده کردن ؟ آیا همون مشکلی که قرار بود حل بشه ، واقعا حل شده ؟ آیا یک مشکل جدید به وجود اومده ؟ همینجاست که دوباره برمیگردیم به همون نقطه ی اول : این دیتا هایی که جمع شده ، خودشون میتونن منبع یک ایده ی جدید بشن ، یا باعث بشن یکی از تصمیمای قبلی توی دیزاین یا حتی خود ایده ی اولیه دوباره بازبینی بشه .