

Codebase & Engineering Quality Pack

ساختار کد، کیفیت مهندسی، قابلیت تحویل پایدار و مدیریت ریسک دانش فنی
مقدمه: چرا صرفاً داشتن ریپوی کد کافی نیست؟

در بسیاری از پروژه‌های استارتاپی، ارائه لینک GitHub یا GitLab به عنوان «مدرک فنی» تلقی می‌شود. این نگاه سطحی فرض می‌کند که وجود کد مساوی با آمادگی مهندسی است، در حالی که تجربه عملی نشان می‌دهد بیشتر شکست‌های فنی نه از نبود کد، بلکه از ناپایداری، غیرقابل‌نگهداری بودن و وابستگی بیش‌از حد به افراد ناشی می‌شود.

هدف از این سند، نمایش «کد تمیز در حد کتاب» نیست. این سند به صورت شفاف نشان می‌دهد که تیم چگونه:

کد را سازماندهی کرده
کیفیت را کنترل می‌کند

تغییرات را بدون فروپاشی سیستم تحویل می‌دهد
و ریسک خروج افراد کلیدی را مدیریت می‌کند

تمرکز اصلی این سند بر قابلیت تحویل پایدار (Sustainable Delivery) و نگهداشت‌پذیری (Maintainability) است، نه زیبایی ظاهری کد.

ساختار کلی پروژه و منطق سازماندهی کد

کدبیس پروژه بر اساس این اصل طراحی شده که هر توسعه‌دهنده جدید بتواند بدون وابستگی به دانش شفاهی، ساختار سیستم را درک کند. ساختار پوشه‌ها و مازول‌ها منعکس‌کننده منطق دامنه (Domain Logic) است، نه سلیقه شخصی یا تاریخچه رشد تصادفی پروژه.

تفکیک مسئولیت‌ها در سطح معماری رعایت شده است؛ به این معنا که منطق کسبوکار، لایه دسترسی به داده، لایه ارائه و لایه یکپارچه‌سازی با سرویس‌های بیرونی به صورت شفاف از هم جدا شده‌اند. این تفکیک باعث می‌شود تغییر در یک بخش، اثر زنجیره‌ای غیرقابل‌کنترل در کل سیستم ایجاد نکند.

ساختار پروژه به‌گونه‌ای انتخاب شده که با رشد تیم و اضافه شدن فیچرهای جدید، نیاز به بازنویسی گسترده ایجاد نشود. این تصمیم آگاهانه برای جلوگیری از بدھی فنی پنهان گرفته شده است.

استانداردهای کدنویسی و تصمیم‌های مهندسی

در این پژوهه، استانداردهای کدنویسی نه به عنوان «قانون زیبایی‌شناسی»، بلکه به عنوان ابزار کاهش ریسک تعریف شده‌اند. هدف از این استانداردها این است که کد:

قابل خواندن باشد

قابل تغییر باشد

و وابسته به ذهن نویسنده اولیه نباشد

نامگذاری متغیرها، توابع و مازول‌ها به‌گونه‌ای انجام شده که نیت (Intent) را منتقل کند، نه صرفاً عملکرد فنی را. این موضوع به‌ویژه در تیم‌هایی با تغییر اعضا اهمیت حیاتی دارد، زیرا بخش بزرگی از انتقال دانش از طریق خود کد انجام می‌شود.

تصمیم‌های مهندسی مستندسازی شده‌اند؛ به این معنا که اگر یک الگوی خاص یا رامحل غیر بدیهی انتخاب شده، دلیل آن مشخص است. این کار از بازگشت دوباره به بحث‌های حل شده جلوگیری می‌کند.

رویکرد تیم به تست و اطمینان از کیفیت

در این پژوهه، تست به عنوان ابزار «اطمینان روانی تیم برای تغییر» تعریف شده است، نه صرفاً برای افزایش عدد Coverage. تست‌ها به گونه‌ای طراحی شده‌اند که تغییرات آینده را ایمن‌تر کنند، نه اینکه مانع حرکت شوند.

تمرکز اصلی تست‌ها بر بخش‌هایی است که:

بیشترین ریسک شکست دارند

بیشترین تغییر را تجربه می‌کنند

یا بیشترین اثر را بر تجربه کاربر دارند

تест‌ها به تیم اجازه می‌دهند با اطمینان بیشتری Refactor انجام دهد، فیچر اضافه کند یا باگ‌ها را اصلاح کند بدون اینکه کل سیستم غیرقابل‌پیش‌بینی شود.

CI/CD و فرآیند تحویل پایدار

تحویل نرم افزار در این پروژه یک فعالیت دستی و پر ریسک نیست، بلکه یک فرآیند قابل تکرار و قابل پیش بینی است. فرآیند CI/CD به گونه‌ای طراحی شده که هر تغییر:

به صورت خودکار بررسی شود

تست شود

و فقط در صورت عبور از معیارهای حداقلی، وارد محیط‌های بعدی شود

این رویکرد باعث می‌شود وابستگی به افراد خاص برای Deploy کاهش یابد و ریسک خطای انسانی به حداقل برسد. تحویل پایدار به تیم اجازه می‌دهد به جای ترس از تغییر، آن را بخشی طبیعی از توسعه بداند.

مدیریت نسخه و تغییرات (Versioning)

مدیریت نسخه در این پروژه فراتر از افزایش عدد نسخه است. نسخه‌ها نمایانگر سطح پایداری، تغییرات ناسازگار و تعهد تیم به مصرف کنندگان داخلی و خارجی سیستم هستند.

تغییرات بزرگ به صورت کنترل شده و قابل ردیابی اعمال می‌شوند. این رویکرد باعث می‌شود در صورت بروز مشکل، امکان بازگشت یا تحلیل ریشه‌ای وجود داشته باشد.

بدهی فنی: شناسایی، ثبت و مدیریت

در این پروژه، بدهی فنی انکار یا پنهان نمی‌شود. بدهی فنی به عنوان واقعیتی اجتنابناپذیر پذیرفته شده، اما به صورت آگاهانه مدیریت می‌شود. بدهی‌های فنی شناسایی شده ثبت می‌شوند و درباره زمان و نحوه بازپرداخت آن‌ها تصمیم‌گیری می‌شود.

این رویکرد از انباسته شدن بدهی پنهان جلوگیری می‌کند و اجازه می‌دهد تیم بین سرعت و پایداری تعادل برقرار کند.

مستندسازی به عنوان ابزار انتقال دانش

مستندات در این پروره جایگزین فکر کردن نیستند، بلکه ابزار انتقال دانش‌اند. مستندات به گونه‌ای نوشته شده‌اند که:

برای فرد جدید قابل فهم باشند

به روزرسانی آن‌ها هزینه غیرمنطقی نداشته باشد

و با واقعیت کد همگام بمانند

تمرکز مستندات بر «چرا» است، نه صرفاً «چه.»

مدیریت ریسک خروج افراد کلیدی

یکی از ریسک‌های جدی هر تیم کوچک، وابستگی شدید به افراد کلیدی است. در این پروره، این ریسک به صورت صریح شناسایی شده و برای آن راهکار در نظر گرفته شده است.

دانش فنی در ذهن یک نفر متمرکز نشده و از طریق:

کد قابل خواندن

مستندات تصمیم‌ها

Review‌های مشترک

و تقسیم مسئولیت‌ها

در تیم توزیع شده است. هدف این است که خروج هر فرد، هر چند در دنک، باعث توقف یا فروپاشی پروره نشود.

قابلیت نگهداری در مقابل کد ایدهآل

این پروژه عمداً بهمنبال کد «بی‌نقص آکادمیک» نبوده است. اولویت با کدی است که:
کار کند

قابل تغییر باشد

و بتوان آن را تحويل داد

این انتخاب آگاهانه برای سازگاری با واقعیت‌های استارت‌آپی انجام شده است.

جمع‌بندی نهایی

این Codebase & Engineering Quality Pack نشان می‌دهد که تیم:

به کیفیت به عنوان ابزار بقا نگاه می‌کند

تحویل پایدار را بر قهرمان‌بازی فردی ترجیح می‌دهد

و مهندسی را یک سیستم می‌بیند، نه مجموعه‌ای از کدها

این سند تضمین نمی‌کند که سیستم بدون خطاست، اما نشان می‌دهد که تیم می‌داند چگونه با خط‌ازندگی
کند، آن را کنترل کند و ادامه دهد.