Engineer

Kodebase

al kildekoden til én app, samlet i ét repository

Hvad det er

En kodebase, på engelsk codebase eller code base, er hele den mængde kildekode, et team redigerer, gennemgår og udgiver som én ting. Når udviklere siger »kodebasen«, mener de stedet, hvor hver ændring af en applikation ender: selve koden, testene, de scripts, der bygger og tjekker den, databaseændringerne og de dokumenter, der forklarer, hvordan delene hænger sammen.

The Twelve-Factor App, en metode til at bygge webtjenester, giver den mest brugbare definition, fordi den gør grænsen til noget, man kan efterprøve. I et centraliseret versionsstyringssystem som Subversion er en kodebase ét repository. I et distribueret system som Git er det ethvert sæt af repositories, der deler en rod-commit: den første commit i historikken, den eneste uden forælder. To repositories, der er vokset fra den samme første commit, er én kodebase, uanset hvor langt de er gledet fra hinanden. To, der begyndte hver for sig, er to kodebaser, også selv om filerne er næsten ens.

Samme dokument parrer definitionen med en anden regel: én kodebase, mange deploys. Et deploy er en kørende instans af applikationen, og der er som regel flere ad gangen. Produktion er ét, staging et andet, og hver udviklers lokale kopi tæller med. De kører alle den samme kodebase, ofte ved forskellige commits: en udvikler har arbejde, som staging mangler, og staging har arbejde, som produktion mangler.

Hvad der hører hjemme i den, følger af det. Kildekoden og alt, hvad der skal til for at gøre den til et kørende program, hører til, også et manifest, der navngiver hver afhængighed og dens version. Selve afhængighederne ligger som regel udenfor og hentes af en pakkehåndtering. Værdier, der skifter fra det ene

deploy til det næste, ligger også udenfor, og frem for alt adgangsoplysninger: en adgangskode, der er committet én gang, rejser med hver kopi af historikken bagefter.

En kodebase kan rumme flere programmer. Et monorepo holder en frontend, et API og databasekoden side om side i ét repository, så én commit kan ændre alle tre på en gang. Efter rod-commit-reglen er det stadig én kodebase. Det modsatte valg spreder en applikation ud over repositories med hver sin historik, og så er hvert repository sin egen kodebase. Twelve-factor siger det ligeud: er der flere kodebaser, er det et distribueret system, og hver komponent er en app for sig.

Brug begrebet i en beslutning

Brug »Hvad er en kodebase?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Systemets form lagt fast på papir først, så delene passer sammen, og prisen er kendt, før nogen skriver kode.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Én infrastrukturplan til 14 afdelinger — 14 forskningsafdelinger på én infrastrukturplan

Én infrastrukturplan til 14 afdelinger Kontakt os →

Hvorfor det betyder noget

Grænsen afgør, hvad der kan ændres samlet. Inden for én kodebase kan en ændring af et API og den skærm, der kalder det, skrives, gennemgås og merges som én enhed. På tværs af to kodebaser bliver den samme ændring til to pull requests, to udgivelser og en periode, hvor den ene side er foran den anden. Begge dele kan fungere; prisen bestemmes af, hvor stregen blev trukket.

Kodebasen er også den enhed, standarder kører over. En testsuite, en linter, en formatter, en regel for commitbeskeder og en task runner tager alle »kodebasen« som deres område, og det, der ligger uden for den, falder uden for dem alle.

Tidlige beslutninger stivner hurtigere i en kodebase end noget andet sted. Hver linje, der skrives efter et valg, forudsætter valget, så en regel om lagdeling eller en kvalitetsstandard fra de første uger bliver formen på alt, hvad der følger. At vende den senere betyder at skrive den kode om, der er vokset op omkring den.

For mennesker er kodebasen det, en nyansat får udleveret. Rummer den alt, der skal til for at bygge, teste og køre applikationen, kommer den nye i gang med én clone og én kommando. Ligger viden dels i koden og dels i en kollegas hukommelse, går den første uge med at finde ud af, hvad der er hvad.

Sådan ser det ud i praksis

En kodebase i mange sprog har brug for én hoveddør. På én platform er kodebasen et polyglot-monorepo i Go, TypeScript, SQL, Python og shell, og hvert sprog kommer med sin egen måde at bygge, teste og linte på. Svaret var én indgang: et go-task-lag, der voksede til 578 navngivne mål, 50 i rodfilen og 528 i namespacede filer under den, med udviklingstilstande til native, Docker og HTTPS, kodekvalitet, databasehåndtering og deploy. Enhver kan køre `task --list` og se hele værktøjskæden. Siden læser også tallet som en advarsel: 578 mål er mere, end nogen kan holde i hovedet, så det er navngivningen og opdelingen i namespaces, der holder det brugbart.

Grundreglerne kan også komme først. Et projekt begyndte den 25. november 2025 med et repository uden indhold, og tre beslutninger fra de første uger holdt stadig ni måneder og flere tusinde commits senere: en lagdelt stak, dataadgang bag stored functions og en zero-warnings-standard, der kom ind, før der var meget kode at holde til den. Begrundelsen på siden er kernen i denne artikel: en standard indført ved commit 5.000 er et oprydningsprojekt, mens den samme standard ved commit 50 bare er, hvordan repositoriet fungerer.

Væggene inde i en kodebase kan være selve designet. I grundlæggerens arkitekturbidrag til NextMariner, en maritim platform bygget i tre lag, står Next.js-frontenden for præsentation og intet andet, et bevidst tyndt Go-API bygget med Huma oven på Fiber router, validerer og orkestrerer, og forretningslogikken bor i PostgreSQL-funktioner. Når en regel ændres, ændres den i ét lag, i SQL. Grænserne blev skrevet ned og håndhævet i review, fordi en konvention, som ingen holder øje med, holder op med at være en.

Dette sites egen kodebase trækker en anden slags streg. Projektsiderne skrives i et separat repository, eksporteres hertil af et script og committes, så sitet kan bygges for sig selv. Inden for denne kodebase er de genererede filer, og instruktionerne for arbejdet her siger, at de aldrig rettes i hånden.

Hvor det går galt

Kopieret kode glider fra hinanden

To applikationer har brug for den samme hjælpefunktion, og det hurtigste er at kopiere den over. Et år senere findes der to versioner med forskellige rettelser, og en fejl, der er lukket i den ene, lever videre i den anden. Twelve-factor kalder flere apps, der deler den samme kode, et brud på metoden og peger på løsningen: flyt den fælles del ud i et bibliotek, hent den ind gennem pakkehåndteringen, og lad hver kodebase angive den version, den bruger.

En hemmelighed havner i historikken

En forbindelsesstreng eller en API-nøgle bliver committet i en konfigurationsfil, og fra da af ligger den i den historik, som hver clone bærer med sig, også efter at en senere commit har slettet linjen. Twelve-factors prøve er værd at tage før hver commit: kan denne kodebase udgives som open source i dag uden at afsløre en eneste adgangsoplysning? Alt, der dumper den prøve, hører hjemme i miljøet for hvert deploy.

En genereret fil bliver rettet i hånden

En fil skrevet af et script ligner enhver anden kildefil, så nogen retter en tastefejl direkte i den. Næste gang scriptet kører, skriver det den gamle tekst tilbage, og rettelsen forsvinder i stilhed. Mærk genererede stier tydeligt, lad generatoren være den eneste, der skriver dem, og send hver rettelse til generatorens input.

Væggene indeni bliver tynde

Under tidspres ryger en databaseforespørgsel direkte ind i en request handler, fordi det er den korteste vej, og den lagdeling, projektet startede med, holder op med at beskrive koden. Hver genvej er lille. Tilsammen gør de en ændring på to linjer til en, der rører hvert lag. En skreven regel hjælper, code review hjælper mere, og et tjek, der stopper buildet, når ét lag rækker ind i et andet, hjælper mest.

En standard dækker halvdelen af koden

En linter sat op til frontenden lader shellscripts og SQL være, og netop de filer samler de problemer, linteren skulle forhindre. En standard for hele kodebasen skal nævne hvert sprog i den, og det er den skjulte pris ved et polyglot-repository.

En fork bliver til et nyt produkt

Et team kopierer et repository for at starte et beslægtet produkt. Begge kopier beholder den samme rod-commit, så efter twelve-factor-reglen er de stadig én kodebase, der nu tjener to apps, og den udvikling i forskellige retninger, der er beskrevet ovenfor, begynder i et helt repositorys målestok. En sammenligning af rod-commits viser, hvilken situation man står i:

```
git -C app-a rev-list --max-parents=0 HEAD
git -C app-b rev-list --max-parents=0 HEAD
```

Ens output betyder en fælles rod og en fælles kodebase; forskelligt output betyder to adskilte historikker.

Læs mere

- I. Codebase — reglen om én kodebase og mange deploys og prøven med den fælles rod-commit bag den.
- II. Dependencies — hver afhængighed erklæret i et manifest og isoleret, så kodebasen siger præcis, hvad den trækker ind.
- III. Config — open source-prøven for, hvad der skal holdes ude af en kodebase.
- gitsubmodules — hvordan Git indlejrer ét repository i et andet, mens hvert beholder sin egen historik.
- git-rev-list — opslagsværket for at liste commits, også rod-commits i en historik.

Ordliste

Vælg ud fra jeres faktiske vilkår

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er en kodebase?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/codebase/