Engineer

God praksis i softwareudvikling

afprøvede arbejdsmetoder til bestemte vilkår

Hvad det er

God praksis i softwareudvikling er en arbejdsform, der har vist sig nyttig nok til at være det normale valg under bestemte vilkår. Det kan være at gendanne en sikkerhedskopi i et separat miljø, når man skal vide, hvor hurtigt driften kan begynde igen. En afsluttet kopiering viser, at jobbet kørte. En gendannelse viser desuden, om de gemte data kan åbnes, hvor længe arbejdet varer, og hvor man endnu må improvisere.

En anbefaling kan komme fra et forskningsmiljø, et projekt eller en organisation, der udgiver standarder. Den kommer ofte med forudsætninger. Holdet har måske adgang til et uafhængigt miljø eller til en anden person, der kan gennemgå ændringen. Et råd kan gøres til en fælles regel, og en kontrakt kan stille yderligere krav. Det er værd at finde ud af, hvem der står bag en regel, og hvor den gælder, inden man beslutter, om en undtagelse er mulig.

NIST’s ramme for sikker softwareudvikling beskriver i version 1.1 praksisser, opgaver og eksempler på gennemførelse. Rammen kan føjes til forskellige udviklingsforløb. Den viser en nyttig forskel mellem det, man vil opnå, og én bestemt måde at opnå det på. At kunne dokumentere en gennemgang er et resultat. En indstilling på en hostingplatform er én mulig vej. Et hold kan undersøge, om dets egen måde at arbejde på giver tilsvarende beviser og opfylder de krav, der faktisk gælder.

Ordene kan også have en formel betydning. IETF udgiver serien Best Current Practice efter den proces, der beskrives i RFC 2026. RFC 8174 præciserer, hvornår ordene med store bogstaver i den tilknyttede konvention har den særlige normative betydning. Når et team kalder noget god praksis, kan det blot mene en velafprøvet anbefaling. Se derfor efter kilden, formålet og anvendelsesområdet.

Brug begrebet i en beslutning

Brug »Hvad er god praksis i softwareudvikling?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Testet, udgivet og til at rulle tilbage, så ændringer når kunderne en tirsdag eftermiddag i stedet for sent en fredag aften.

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. Har automatiseret GIS-SaaS-udrulning med CI/CD — udrulning, data og rapportering, alt sammen uden manuelt arbejde

Har automatiseret GIS-SaaS-udrulning med CI/CD Kontakt os →

Hvorfor det betyder noget

En tydelig arbejdsform gør, at man slipper for at genåbne den samme diskussion ved hver ændring. En ny udvikler ved, hvilke beviser der skal med, når en kollega bliver bedt om at gennemgå koden. Driften får oplysninger i en form, den kan sammenligne fra den ene udgivelse til den næste. »Gennemgå koden grundigt« giver hvert teammedlem sin egen målestok. At kontrollere, at kunden kun kan blive opkrævet én gang, giver folk noget observerbart at tale om.

En kendt regel hjælper også holdet med at opdage undtagelser. Det kan normalt være klogt at idriftsætte små ændringer ofte. Opdatering af firmwaren på en maskine kan kræve, at en tekniker isolerer den, installerer ændringen og afprøver udstyret på stedet. Små ændringer kan stadig gøre design, test og rækkefølge bedre. Tidspunktet for den fysiske opdatering afhænger af, hvornår den kan afprøves og eventuelt fortrydes.

DORA beskriver udvikling på en fælles hovedgren med hyppig integration af små ændringer. Det er en god ledetråd, når et hold bruger meget tid på sammenfletning af store grene. De lokale målinger er stadig nødvendige. Hvor lang tid går der fra en ændring foreslås, til den er integreret? Hvad bliver afvist? Hvilke problemer opstår bagefter? Færre grene i oversigten gør ingen forskel, hvis der blot ligger mere arbejde på hver enkelt.

Sådan ser det ud i praksis

Forestil dig et team, der udvikler et internt rapportværktøj. En anden person skal gennemgå hver ændring, og den fælles testpakke skal bestås. Månedsrapporten bliver forsinket på grund af en fejl i et filter. Det tager flere timer at køre hele pakken, blandt andet fordi den afprøver en anden form for eksport. Rettelsen gælder kun filteret og kan trækkes tilbage, inden rapporten sendes ud.

Holdet undersøger først de krav, der gælder. En påkrævet godkendelse af rapporten skal stadig indhentes. Så afgrænser de risikoen. Rapporten skal vælge de rigtige rækker, bevare sin gruppering og udelade data fra andre afdelinger. Udvikleren kører afgrænsede test med kendte data og sammenligner rapporten med de forventede rækker. En kollega gennemgår både ændringen og resultatet. Holdet beholder den tidligere rapport og aftaler, hvem der kan standse udsendelsen. Den fulde pakke køres bagefter for at få viden til en eventuel varig ændring.

Skriv den sædvanlige regel, grunden til undtagelsen, hvad der stadig kontrolleres, og hvornår aftalen udløber. Notér også, hvad der faktisk skete. Hvis det samme problem dukker op næste måned, kan holdet gøre en hurtig kontrol af rapporten til en del af det normale arbejde. Afprøv den ved bevidst at vælge en forkert måned og medtage en række fra en anden afdeling. Se, om kontrollen opdager begge fejl. Tidtagning alene kan ikke give det svar.

Sagen om 22 automatiske kontroller før en commit viser, hvordan projektets aftaler blev håndhævet. En anden sag om tilgængelighedsfejl, som de automatiske tjek overså viser, at en konsekvent gennemført kontrol stadig kan dække mindre end brugerens samlede oplevelse.

Hvor det går galt

Tjeklisten bliver målet. En udfyldt liste kan eksistere samtidig med det problem, den skulle forebygge. Vælg et tilfælde, hvor arbejdsformen burde gøre en forskel, og se efter virkningen. Åbn for eksempel et udtræk af de gendannede data.

Værktøjet kommer til at stå for metoden. En platform til gennemgang giver ikke i sig selv nogen tid til at gøre indsigelse mod en ændring. Skriv, hvilket resultat den skal understøtte, og undersøg, om en kollega kan få de relevante oplysninger i tide.

Undtagelsen bliver en vane. En hurtig løsning bliver let brugt igen. Aftal, hvem der ejer undtagelsen, hvilke garantier der stadig gælder, og hvornår den skal vurderes. En gentagelse giver anledning til at forbedre den almindelige metode.

En anbefaling kopieres uden dens vilkår. Hyppige udgivelser er langt lettere på nogle systemer end på andre. Sammenlign vilkårene, begynd i mindre skala, og beslut på forhånd, hvilke observationer der taler for at fortsætte eller stoppe.

Læs mere

- NIST's ramme for sikker softwareudvikling, version 1.1 — praksisser, opgaver og eksempler på gennemførelse.
- RFC 2026, afsnit 5 — betydningen af Best Current Practice i IETF's standardiseringsarbejde.
- RFC 8174 — betingelserne for, at udvalgte ord med store bogstaver har normativ betydning.
- DORA om udvikling på hovedgrenen — små ændringer, hyppig integration og måder at vurdere virkningen på.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er god praksis i softwareudvikling?« 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/engineering-best-practices/