Engineer

Software

programmerne og dataene, en computer kører

Hvad det er

Software er alt det i et computersystem, som man ændrer ved at skrive frem for ved at lodde: programmerne, de data, de har med sig, og de indstillinger, der styrer dem. Danske fagfolk bruger det engelske ord, i bestemt form softwaren, mens det ældre danske programmel mest ses i gamle standarder og kontrakter. Et enkelt stykke software kaldes et program, og noget, en person åbner for at løse en opgave, hedder som regel en applikation eller en app. Når nogen siger »koden«, mener de typisk den kildetekst, en udvikler redigerer.

Modstykket er hardware, den fysiske maskine, og firmware ligger imellem de to. NIST's ordliste trækker grænsen ved ændring: software er programmer og tilhørende data, som kan skrives eller ændres, mens de kører, og firmware er den slags, der ikke kan ændres under kørslen. Styresystemet på en bærbar er software, koden i tastaturets controller er firmware, og selve tastaturet er hardware.

Inden i softwaren findes der lag. Systemsoftware som et styresystem eller en driver styrer maskinen. Applikationssoftware gør noget for et menneske: et regneark, et bookingsystem, en beskedklient. Imellem dem ligger biblioteker og pakker, færdige dele, som begge slags bygges af, og i et moderne produkt er langt det meste af koden kommet den vej i stedet for at være skrevet til lejligheden.

En sidste skelnen får betydning senere. Kildekode er det, mennesker skriver og læser, og en compiler eller en fortolker gør den til instruktioner, som en processor udfører. Begge dele er software, men de kan ikke bytte plads, når man køber. Får man kun de kompilerede filer udleveret, er det kun leverandøren, der kan ændre dem.

Brug begrebet i en beslutning

Brug »Hvad er software?« 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

Den, der bestiller, køber eller lejer software, vælger samtidig, hvem der skal holde den ved lige. Det, der gør software tillokkende, nemlig at en kopi intet koster, og at en ændring efter levering er billig sammenlignet med at bygge en maskine om, er også grunden til, at den aldrig bliver færdig. Komponenterne udsender rettelser. Styresystemer og browsere udfaser grænseflader. En dag offentliggøres en sårbarhed i et bibliotek, som ingen på holdet husker at have valgt. Alt det lander som arbejde hos den, der ejer koden.

De gængse veje fordeler arbejdet forskelligt:

- **Lejet som tjeneste.** Leverandøren står for opdateringer og servere. Til gengæld arver man deres udgivelsesplan, deres priser og besværet ved at skifte, og ens data ligger hos dem.
- **Licenseret og drevet selv.** Man bestemmer selv, hvornår der opgraderes, og står så også selv for opgraderingen, backuppen og sikkerhedsrettelserne.
- **Bygget til formålet.** Den passer til arbejdsgangen i stedet for omvendt, og vedligeholdelsen hører til ejeren, så længe systemet kører. Om valget var klogt, afgøres af udgifterne efter lanceringen.
- **Open source.** Under en licens, der opfylder Open Source Definition, må man bruge, undersøge, ændre og videregive koden. Nogen skal stadig lægge opdateringerne ind, og gratis at hente er en anden egenskab end gratis at drive.

Uanset vej er der to spørgsmål, som er værd at stille, før man skriver under. Hvad får man udleveret til sidst: kildekoden, vejledningen til at bygge den og en liste over, hvad den består af? Og hvordan vil leverandøren holde den sikker? NIST's Secure Software Development Framework er blandt andet skrevet til den anden samtale, fordi det giver køber og leverandør et fælles sprog om udviklingspraksis. Til det første spørgsmål beskriver en rapport fra NTIA en

software bill of materials, forkortet SBOM, som en formel optegnelse over de komponenter, der indgår i softwaren, og over deres plads i forsyningskæden. At bede om sådan en stykliste er en billig måde at se, hvor meget af det, man betaler for, der er skrevet af andre.

Sådan ser det ud i praksis

På én platform blev sikkerhedsarbejdet flyttet fra mavefornemmelse til byggeprocessen. I et løbende sikkerhedsprogram dannes der en SBOM ved hvert build med Anchore og Syft, Dependabot holder øje med 6 økosystemer, CodeQL analyserer koden, ZAP tester den kørende instans, og hver GitHub Action fra tredjepart er låst til en commit-hash frem for et tag. Ved siden af automatikken ligger 21 skriftlige sikkerhedsaudits, heriblandt en trusselsmodel. En sårbarhed, som de konfigurerede tjek genkender, kan få buildet til at fejle. Siden nævner også prisen: mængden af fund skal sorteres løbende, og det arbejde holder aldrig op.

Vejen fra ændring til drift er den anden halvdel. Hvor udgivelser havde hvilet på, at nogen huskede en række manuelle trin og udførte dem lidt forskelligt hver gang, sparede automatiserede bygge-, test- og udrulningspipelines anslået 4.000 timer, og udgivelserne blev både hurtigere og mere pålidelige. Gevinsten kom af, at forløbet var det samme hver gang, og ikke af, at folk passede ekstra godt på.

Grænsen mellem software og hardware er helt konkret på et reparationsbord. Reparationer af bærbare computere omfattede skærme, hængsler, tastaturer og bundkort og desuden softwareproblemer på Linux, Windows og Mac. En bærbar, der ikke vil starte, kan have fejlen i laderen, bundkortet eller batteriet, så arbejdet begyndte med at finde den egentlige fejl, før noget blev rørt.

Hvor det går galt

En sikkerhedsadvarsel bliver til en uges udgravning

Der offentliggøres en sårbarhed i en udbredt komponent, og ingen kan sige, hvilke af organisationens systemer der indeholder den. Overblikket fandtes kun i folks hoveder og i låsefiler, som ingen læser, så det skal stykkes sammen under tidspres, system for system. Dan en SBOM som en del af hvert build, og gem den sammen med den udgivelse, den beskriver. Så bliver spørgsmålet »er vi ramt?« til en søgning.

Udgivelser afhænger af, hvem der laver dem

Udrulningen virker, når én kører den, og går i stykker, når en anden gør det, og forskellen viser sig at være et trin, der kun findes i nogens hukommelse. Hver udgivelse bliver en lille improvisation. Læg hele vejen ind i en udrulningspipeline, der bygger, tester og udruller på samme måde hver gang, og behandl hvert manuelt trin, der er tilbage, som en fejl, der skal fjernes.

Budgettet slutter ved lanceringen

Efter to eller tre år føles hver opdatering risikabel, frameworket er flere hovedversioner bagud, og det eneste forslag på bordet er at skrive det hele om. Softwaren blev betalt som en bygning og derefter ladt i fred, mens alt omkring den flyttede sig. Sæt en fast post af til vedligeholdelse, opgradér afhængigheder i små, regelmæssige skridt, mens hvert skridt stadig er lille, og regn med, at platformene nedenunder ændrer sig i deres eget tempo.

Navnet står fast, mens koden bag det skifter

Det samme build giver forskellige resultater to dage i træk uden et eneste commit imellem. Et sted peger en reference på et mål, der flytter sig: et versionsinterval, et `latest` -tag eller en CI-action, der hentes via et tag, som ejeren kan flytte. Lås tredjepartskode til en præcis version eller, hvor tags kan flyttes, til en commit-hash, læg låsefilen i versionsstyring, og lad en bot foreslå opdateringerne, så hver ændring kommer som en diff, man kan gennemgå.

Læs mere

- NIST CSRC Glossary: software — de definitioner, NIST's publikationer bruger, herunder den, der skiller software fra firmware.
- NIST SP 800-218, Secure Software Development Framework — grundlæggende praksis for at udvikle software med færre sårbarheder, skrevet så købere kan bruge den over for leverandører.
- NTIA: The Minimum Elements for a Software Bill of Materials — hvad en liste over komponenterne i et stykke software skal indeholde for at være til nytte.
- The Open Source Definition — de ti kriterier, en licens skal opfylde, før software under den regnes for open source.

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 software?« 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/software/