Hur Eneo utvecklas med hjälp av AI

Teknikspåret på Eneos användarträff, 17 november 2026

Çagri Çimen

Systemutvecklare och tekniskt ansvarig för AI-plattformen Eneo

Max Eriksson

Systemutvecklare i AI-teamet och Eneo

Så här ser dagen ut

09.30	**Resan**	Från handskriven kod till AI-native
09.45	**Arbetssättet**	Team, kontext, verifiering och förvaltning
10.15	**Frågor**	
10.30	Paus	
10.45	**Eneo**	Frontendbytet, och tal till text live
11.30	**Frågor**	
11.45	Lunch	
13.00	**Ni bygger själva**	Praktiskt pass, med fika 14.30
15.30	**Alla tillsammans**	Vart Eneo är på väg

Agenterna skriver koden. Vad gör vi?

Förmiddagen handlar om vår del: behovet, arkitekturen, kontrollen och förvaltningen.

Från handskriven kod till AI-native

	Förstå behovet	Forma arkitekturen	Skriva koden	Köra kontroller	Bedöma resultatet
För hand Där Eneo började	Vi	Vi	Vi	Vi	Vi
AI som hjälp Förslag och chatt	Vi	Vi	Vi + AI	Vi	Vi
AI-native Där vi arbetar i dag	Vi	Vi + AI	Agenterna	Agenterna	Vi

Vår del har inte försvunnit. Den ligger före och efter koden.

Vi arbetar mer som lösningsarkitekter

Mindre av vår uppmärksamhet		Mer av vår uppmärksamhet
Syntaxen	→	**Verksamheten och problemet**
Hur koden ser ut	→	**Arkitektur och designmönster**
Var kommentarerna står	→	**En lösning som håller över tid**

Koden går fort att skriva. Det dyra misstaget är att bygga fel sak.

Därför behöver vi förstå verksamheten djupare än förr.
Detaljerna vi fortfarande fastnar i gäller design och UX.

Det som var för dyrt går nu att göra

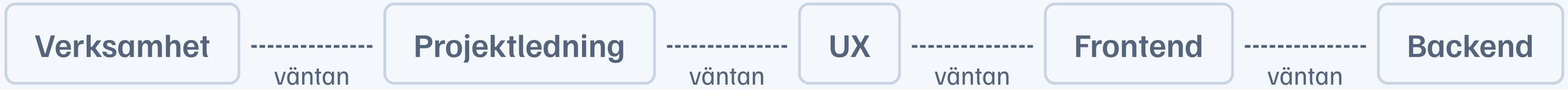
Förr	För dyrt och för långsamt. Vi lät det vara.
Nu	**Vi tar tag i det, och bygger det så att andra kan ta del av det.**

Vårt exempel är Eneos frontendbyte. Det visar vi om en stund.

Att skriva koden är inte längre det som tar tid.

Det som tar tid nu	Så möter vi det
Förstå vad som behövs	**Ett team kring behovet**
Ge agenten rätt sammanhang	**Kontext för människor och AI**
Avgöra om resultatet håller	**Verifiering vid kontrollpunkter**
Äga det som byggs	**Förvaltning som tål takten**

Från beställa och leverera till ett team

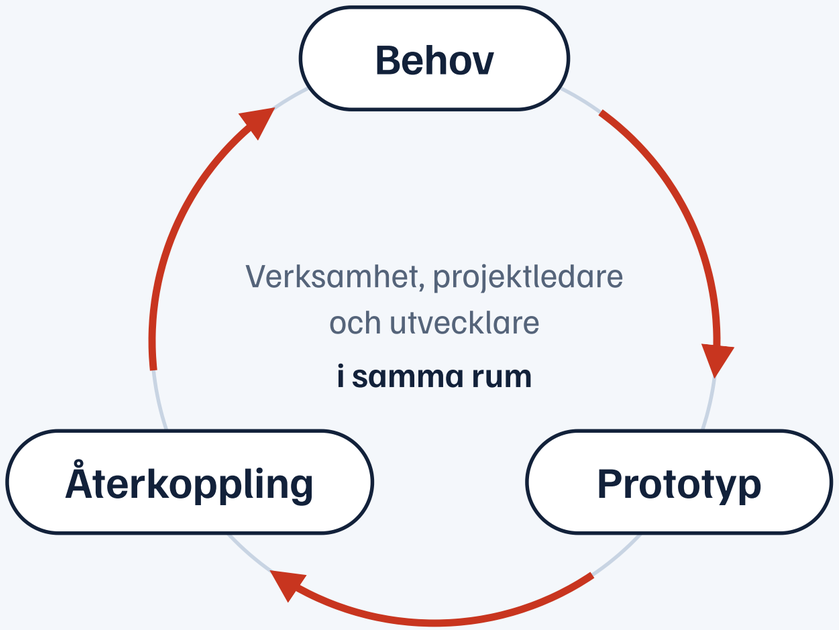


Varje överlämning ger ledtid, och ett ”vi” och ett ”ni”.

Verksamhet	Projektledning	UX	Frontend	Backend

Samma kompetenser, sida vid sida, med gemensamt ansvar för hela lösningen.

Prototypen byggs under samtalet

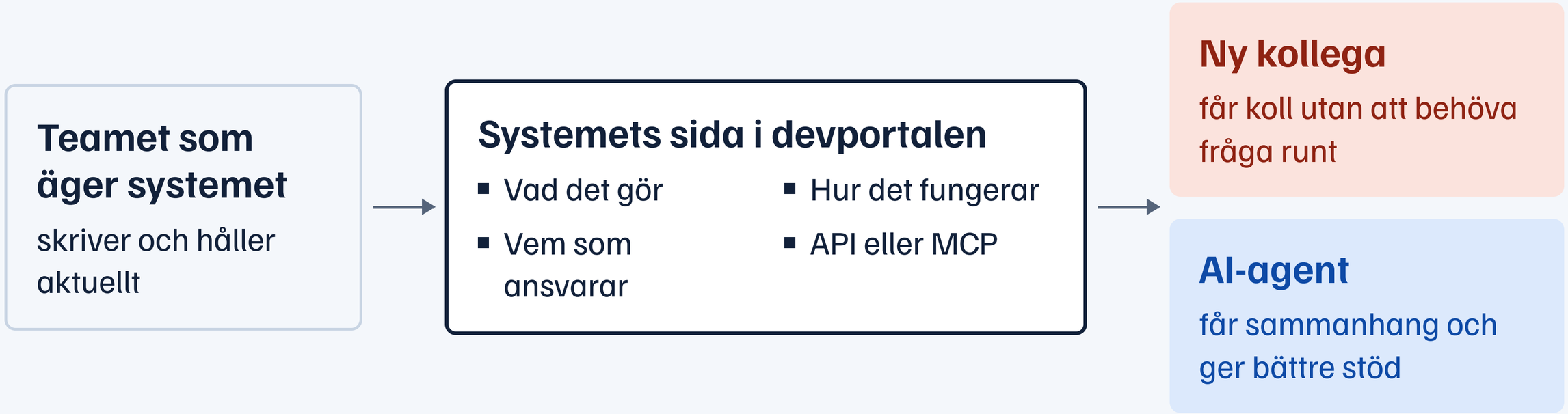


Att översätta behov till en teknisk lösning är vårt ansvar.

Den som ska använda lösningen vet sällan i detalj vad den vill ha, och inte alltid vad som har blivit möjligt.

Därför visar vi tidigt vad som går att göra. Det öppnar för andra lösningar än den man först tänkte sig.

Samma källa för människor och AI



Metakatalogen, vår IDM, byggs så: kunskapen och datat går att nå för båda.

devportal.sundsvall.dev

Från personberoende till organisationens förmåga

	När kunskapen sitter hos en person	När kunskapen går att nå
Ny i teamet	Väntar på att någon har tid att förklara	**Börjar med sammanhanget på plats**
Nytt projekt	Börjar från noll och bygger eget	**Bygger på tjänster och data som redan finns**
Nya idéer	”Det tar flera år”, och blir inte av	**Går att genomföra, och går att dela**

Vi tänker före och bedömer efter

Före: vi + AI

Vi bollar arkitekturen

Med varandra och med AI, som får ifrågasätta hur vi tänker. Vi bestämmer också vad klart betyder.

Under: agenterna

Skriver, kontrollerar och rättar

Agenterna kör tester och bygge, och går tillbaka tills kontrollerna går igenom.

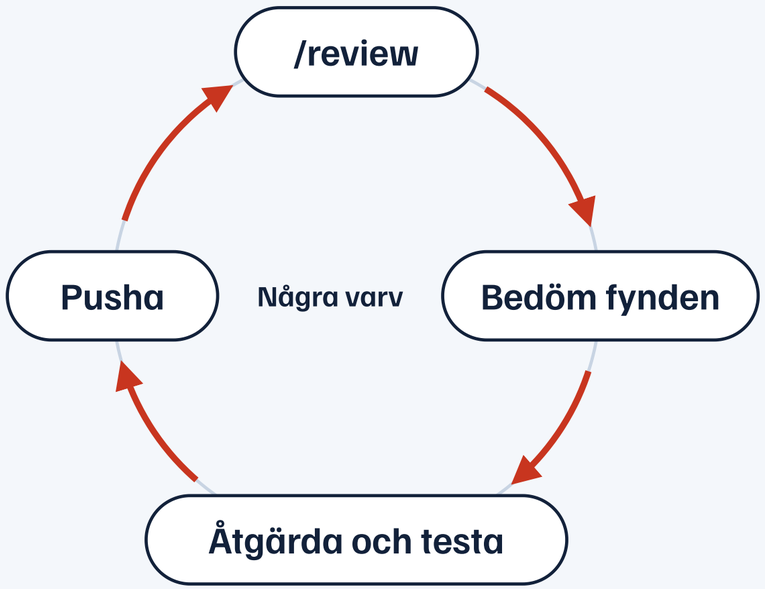
Efter: vi

Klickar igenom och läser logiken

Första bästa versionen går sällan vidare. Vi återkopplar tills det håller.

Högre tempo och bättre kvalitet kommer från samma sak: tydliga kontrollpunkter.

Ett extra steg för kvalitet: review-agenten



Hittar buggar och det som kan bli bättre,
med skills och regler för just det kodförrådet.

Skriven med Claude Code, granskad med
Codex. Vi litar inte blint på en modell.

Agenten ger råd. Vi avgör,
och det är vi som mergar.

Koden finns på github.com/CCimen/review-agent

Högre takt kräver bättre förvaltning

Bestämt innan takten ökar	
Struktur	Hur ett projekt ser ut, så att nästa person eller agent hittar
Stack	Vad vi bygger med, och vad vi låter bli att ta in
Beroenden	Hur och när bibliotek uppdateras
Bygga eller återanvända	Öppen källkod som redan håller går före eget bygge

Takten förstärker den förvaltning man redan har.
Brister märks hårdare och snabbare.

Ett bra första uppdrag för AI: gå igenom beroenden, tester och dokumentation.

Frågor så här långt?

Paus

Vi börjar igen 10.45, med Eneo som exempel.

Eneos frontendbyte

Från		Till
Svelte	→	**React**
SvelteKit	→	**Vite**
Melt UI	→	**Astryx från Meta** i beta

Ett praktiskt exempel: en större ändring som vi inte hade gett oss på utan AI.

Vi gör den för långsiktigheten: ett större ekosystem som fler underhåller.

Öppen källkod ger oss mycket gratis, till exempel komponentbiblioteket Astryx.

Har det varit problemfritt?

Absolut inte.

”One shot” räcker sällan. Det krävs några varv för att hitta rätt lösning.

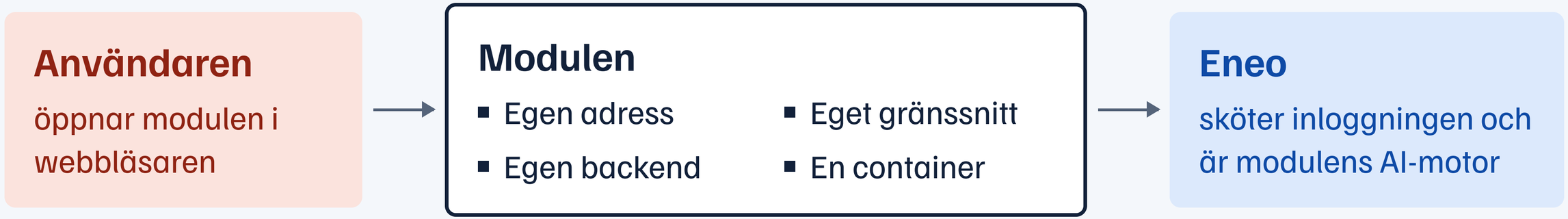
Det hör till utvecklingen. Vi väntar oss inte att modellen bygger allt rätt direkt.

Modellerna blir bättre, men vi behöver vara med i processen ändå.

Tal till text

Byggd som	Bygger på	Körs i
den första modulen	**flöden, ett nytt koncept**	**nya frontenden**

En modul är en liten webbapp bredvid Eneo



Modulen gör en sak bra. Inloggning, modeller och behörigheter finns redan i Eneo.

Tal till text är den första: inspelning, transkribering och redigering av texten.

Module kit: slipp uppfinna hjulet igen

Det här importerar du	
Inloggning	Via Eneo. Slås på för de sidor som behöver den.
Väg till Eneos API	Modulen når bara de anrop den uttryckligen tillåter
Gränssnitt	Eneos tema och sidram i Astryx, testat för tillgänglighet
Mall	En fungerande modul att kopiera, med CI och agentinstruktioner

Paketen importeras. Mallen kopieras. Du bygger det modulen ska göra.

Version 0.1, hämtad ur tal till text.
Öppen källkod på github.com/eneo-ai/eneo-module-kit

13.00 bygger ni själva

1 Grunden: be din agent bygga i Eneo

Eneo-skillen lär din agent hur Eneo fungerar. Du beskriver assistenten du vill ha, i stället för att klicka fram den.

2 Sedan, för den som vill: en egen modul

Valfritt. Mallen i module kit har instruktioner för din agent och en låtsas-Eneo att utveckla mot.

Gör det här nu, före lunch: skapa ditt testkonto på eneo-anvandartraff.sundsvall.dev, och kontrollera att Claude Code eller Codex startar.

Frågor och samtal

Hur arbetar ni med AI i er utveckling i dag, och vad hindrar er?

Frågor om roadmap tar vi när alla ses 15.30. Koden finns på github.com/eneo-ai/eneo.

Bygg mot Eneo med din egen agent

Slå upp datorn och starta Claude Code eller Codex.

Så här gör vi i eftermiddag



13.00
Kom igång

13.15
Första assistenten

Skapa den
genom din agent

13.45
Bygg vidare

Förstå, lägg till kunskap,
bygg en chattsida. En egen
modul för den som vill.

14.30
Fika

15.00
Bygg klart, visa och summera

Fastnar du? Räck upp handen eller fråga grannen. Vi går runt hela tiden.

Kom igång

1. **Skapa ett testkonto**

   eneo-anvandartraff.sundsvall.dev, under Testkonto. Din mejladress och koden vi visar.

2. **Klistra in startuppmaningen i din kodagent**

   Den står på samma sida. Agenten hämtar Eneo-skillen och läser den.

3. **Skapa en API-nyckel när agenten ber om det**

   Mina API-nycklar i Eneo: hemlig nyckel, omfång Yta, behörighet Skriv. Klistra in den i .env, inte i chatten.

4. **Kontrollera att det fungerar**

   Agenten listar de ytor nyckeln ser. Då är du i gång.

Be om en assistent

```
> Skapa en upphandlingsassistent i Eneo som
hjälper handläggare att granska
förfrågningsunderlag mot våra riktlinjer.

Välj en lämplig modell, skriv en
systemprompt och testa assistenten.
```

Exemplet står på träffsidan. Byt gärna till något från din egen verksamhet.

Agenten frågar det den inte kan avgöra själv, till exempel era riktlinjer

Den väljer modell, skriver systemprompten och testar med några frågor

Du får namn och länk, läser prompten och provar själv

Klart när du har chattat med assistenten i Eneo och håller med om det agenten skrev.

Bygg vidare i din egen takt

Prova det som passar dig	
Förstå	”Förklara hur ytor, assistenter och kunskap hänger ihop i Eneo.”
Kunskap	Ge assistenten egna dokument att svara utifrån
Bygg	”Bygg en enkel chattsida mot min assistent.”
Dela	Publicera assistenten, eller skapa en delad yta och en nyckel för den

Assistenter och ytor är grunden. Moduler är nästa steg.

Skillen fungerar även mot er egen Eneo. Mallen för moduler finns i eneo-module-kit.

Arbeta som i förmiddags

Kontext	Låt agenten läsa skillen först. Fråga vad den har förstått innan den börjar bygga.
Kriterium	Bestäm vad klart betyder. En mening räcker, men skriv den före arbetet.
Kontroll	Prova själv. Att något ser färdigt ut betyder inte att det är det.

Sitt gärna två och två. Den ena styr agenten, den andra granskar.

Fika

Vi fortsätter 15.00. Då bygger ni klart, och 15.15 visar vi för varandra.

Vad byggde ni? Och vad saknade agenten?

Det agenten saknade är det vi ska dokumentera härnäst.
Lägg en issue eller en PR på github.com/eneo-ai/eneo.

15.30 ses vi alla: vart Eneo är på väg.