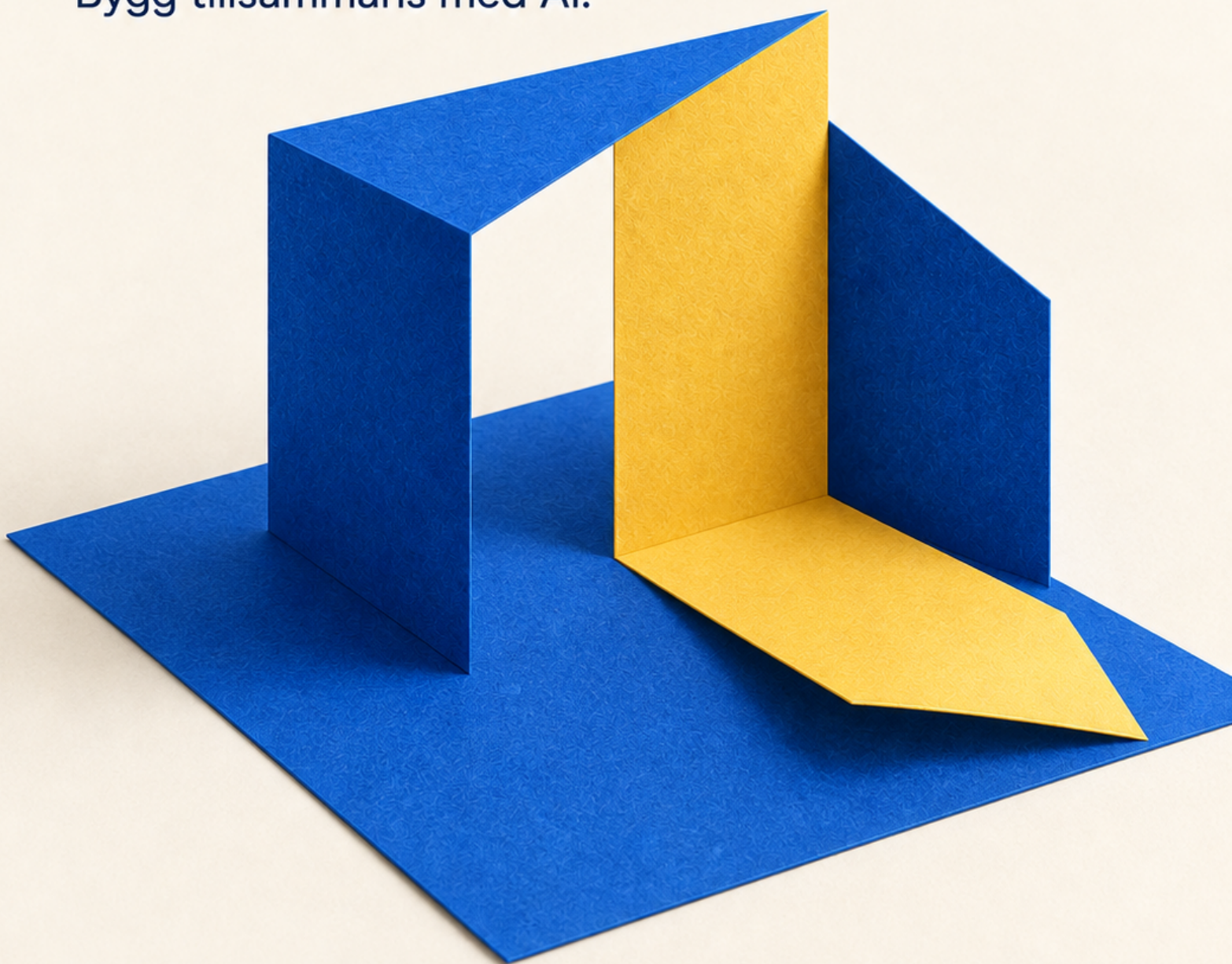


Vad är vibecoding?

Från idé till eget verktyg.
Bygg tillsammans med AI.



På svenska. För företag.

DIN VÄG GENOM MANUALEN

Börja här

För dig som kan din verksamhet och undrar om du kan skapa egen mjukvara med hjälp av AI.

Du förstår arbetssättet, kan beskriva ett rimligt första bygge och vet vad du behöver ta ansvar för när idén växer.

Du beskriver. AI hjälper till att bygga.	3
Börja med en vanlig arbetsdag	4
Den första versionen är ett samtal	5
Vad är värt att bygga?	6
När fler ska använda verktyget	7
Ta med en idé till AI	8

Källor och vidare läsning finns på sidan 9.

Läs från början eller slå upp det som hjälper dig just nu. Exempelen är påhittade för att visa hur du kan tänka. Alla uppdrag och förklaringar finns här i manualen.

BÖRJA HÄR

Du beskriver. AI hjälper till att bygga.

Du har kanske tänkt: "Det borde finnas ett litet program för just det här." Nu kan du utforska den idén tillsammans med AI.

Vibecoding används här för att skapa mjukvara genom att beskriva vad du vill ha, låta AI hjälpa till med koden och sedan förbättra resultatet tillsammans. Ordet används lite olika. Replit beskriver arbetssättet som att formulera en idé, bedöma det som kommer tillbaka och arbeta vidare i flera omgångar. [1]

Alex driver en cykelverkstad. Hon lånar ut cyklar medan kundernas egna repareras, men uppgifterna hamnar på lappar och i kollegornas huvuden. Alex och verkstaden är ett påhittat exempel. Hennes önskan är enkel: se vilka lånecyklar som är lediga och vilka som ska lämnas tillbaka.

Hon kan beskriva vardagen för en AI som hjälper till att bygga program. Då kan de ta fram en första version att titta på och prova. Verktöget behöver inte ha en AI-chatt inbyggd bara för att AI hjälpte till att skapa det. En tydlig lista med bra funktioner kan vara hela poängen.

Din verksamhetskunskap behövs

Du vet vilka problem som är irriterande på riktigt. AI kan hjälpa dig att undersöka hur ett verktyg skulle kunna lösa dem.

BESKRIV BEHOVET

Börja med en vanlig arbetsdag

Du behöver kunna förklara vad som ska fungera. Det går utmärkt att börja på vanlig svenska.

Alex beskriver först vad som händer vid disken. En kund behöver en lånecykel. Kollegan letar efter en ledig cykel, antecknar utlåningen och vill senare kunna registrera återlämningen. Ibland är en cykel tillbaka men trasig. Då får den förstås inte dyka upp som ledig igen.

När detta händer	Ska verktyget hjälpa så här
En cykel lånas ut	Visa att den är utlånad och när den väntas tillbaka.
En cykel lämnas tillbaka	Låt personalen markera om den är redo att lånas ut igen.
En cykel behöver lagas	Håll den borta från listan över lediga cyklar.

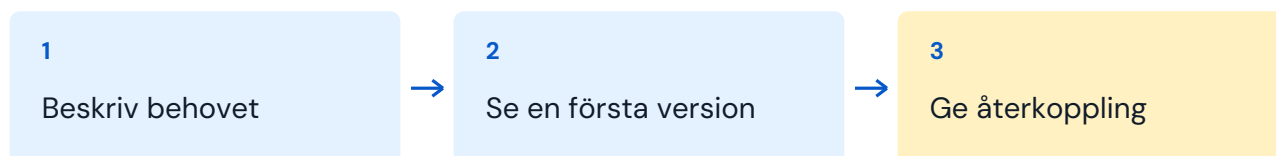
Det här är början på hennes **krav**: vad programmet behöver klara. Hon kan lägga till vilka som ska använda det och vilka uppgifter de behöver se. Replits guide om att avgränsa ett projekt betonar just användare, huvudsakliga funktioner och vad som får vänta. [2]

Alex kan också visa en egen skiss eller beskriva ett verktyg hon tycker är lätt att använda. Hon behöver inte välja färgen på varje knapp än. Frågan om en trasig cykel räknas som ledig är betydligt mer brådsande.

TITTA, PROVA OCH FÖRBÄTTRA

Den första versionen är ett samtal

En prototyp är en tidig version som hjälper dig upptäcka vad som behöver ändras.



Sedan går arbetet ett varv till. Replits beskrivning av vibecoding bygger på just denna växling. [\[1\]](#)

Alex börjar med tre påhittade cyklar och några låtsasutlåningar. När hon ser prototypen upptäcker hon att datumet finns, men att den viktigaste frågan fortfarande är svår att besvara: vad kan vi låna ut just nu? Hon ber därför om en tydlig vy för de cyklar som är klara att användas.

Sedan låtsas hon att en kund kommer tillbaka med ett punkterat däck. Om cykeln ändå hamnar bland de lediga har hon hittat ett fel i beteendet. Hon kan beskriva händelsen och vad hon väntade sig, så att AI kan hjälpa henne att undersöka och ändra lösningen.

Återkopplingen blir konkret: "Efter att jag markerat cykeln för reparation ska den försvinna från lediga cyklar." Det ger mer att arbeta med än "gör appen bättre". På samma sätt kan en kollega visa att en text är otydlig eller att viktig information ligger för långt ned.

VÄLJ OMFATTNING

Vad är värt att bygga?

En egen lösning ska förtjäna sin plats i företaget. Det gör den genom att hjälpa någon med ett verkligt behov.

Alex börjar med att titta i verkstadens befintliga system. Kanske finns redan en funktion för lånecyklar. Om den är begriplig och passar arbetet kan hon använda den. Om det saknas något litet kan en justering eller en enkel koppling vara mer intressant än att ersätta hela systemet.

I vårt exempel upptäcker hon att hon fortfarande vill utforska en egen översikt. Då begränsar hon första bygget till lånecyklarna. Kundbokning, reservdelslager och personalens semester får bo kvar där de redan fungerar. De behöver inte flytta bara för att hon har fått en ny idé.

Jämför hela arbetet

När hon jämför en färdig tjänst med ett eget bygge räknar hon även in sin tid, hjälp hon kan behöva och löpande kostnader. Hon frågar vem som tar hand om ändringar och fel. Hon vill också veta om hon kan få ut sina uppgifter om hon senare väljer en annan lösning.

En användbar avgränsning

För Alexs första version räcker det att personalen kan se cyklarnas läge, registrera en utlåning och hantera en återlämning. När det hjälper i vardagen har hon bättre underlag för nästa beslut.

FRÅN IDÉ TILL VARDAG

När fler ska använda verktyget

Nu tillkommer frågor som inte syns på den första snygga skärmen.

För att kollegorna ska kunna öppna en webbapp behöver den köras på ett sätt de kommer åt. Det är här **hosting** kommer in: någonstans där appen kan finnas och fungera. Läs "Vad är hosting?" för att få hela bilden av appen, internetadressen och datorerna som gör jobbet.

Alex behöver också bestämma var utlåningarna sparas och vem som får se och ändra dem. En kollega ska inte råka förlora gårdagens uppgifter när sidan öppnas igen. Manualen "Data, inloggning och åtkomst" förklarar de delarna, utan att du behöver lära dig allt mitt i ditt första byggsamtal.

Kod som AI har skapat kan innehålla fel och säkerhetsbrister. GitHub framhåller därför att genererad kod behöver granskas och testas. [3] Att Alex kan klicka sig igenom en snygg prototyp visar en del av beteendet. Det visar inte i sig att uppgifterna är skyddade eller att lösningen klarar företagets vardag.

Alex ansvarar för att ordna den bedömning som bygget behöver, ibland med teknisk hjälp. Hon kan leda arbetet utan att själv kunna varje kodrad. Innan verktyget blir viktigt för verksamheten behöver hon även en plan för fel, ändringar och säkerhetskopior. Det tar vi vidare i "Att äga det man bygger".

DITT NÄSTA INITIATIV

Ta med en idé till AI

Börja med något du kan förklara för en kollega. Då har du också början på ett bra bygguppdrag.

Välj en återkommande irritation i ditt företag. Beskriv vad som händer i dag, vem det ställer till det för och vad personen behöver kunna göra i stället. Om du har en idé till lösning får den gärna följa med. Lämna också utrymme för att upptäcka en enklare väg.

Alexs första uppdrag

Jag driver en cykelverkstad och vill få bättre koll på våra lånecyklar. Personalen behöver se vilka som är lediga, låna ut en cykel och registrera om den kommer tillbaka hel eller behöver lagas. Hjälp mig beskriva en liten första version med påhittade uppgifter. Fråga om sådant som påverkar hur arbetet ska fungera. Visa ditt förslag och förklara vad som kan vänta. Därefter kan vi börja bygga en prototyp.

När förslaget kommer kan Alex bedöma om AI har förstått verkstaden. Om det plötsligt handlar om kundklubbar och bonuspoäng leder hon tillbaka samtalet till lånecyklarna. Hon behöver inte tacka ja till alla funktioner bara för att de går att bygga.

Hennes första framsteg är en tydligare idé som hon kan diskutera, se och förändra. Nästa beslut blir om hon vill fortsätta bygga, använda en färdig tjänst eller ta hjälp. Med ett konkret exempel blir det mycket lättare att förklara vad hon behöver.

ATT GÅ VIDARE

Källor och nästa läsning

Text, exempel och figurer är framtagna för AI Manualer. Källorna nedan stödjer begrepp och sakuppgifter; de är inte svenska original till den här texten. Länkarna är lästa i september 2026.

[1] Replit Learn / [What is vibe coding?](#)

Stöd för den breda förklaringen av vibecoding som att beskriva, bedöma och förbättra ett program med AI. Begreppet används på olika sätt; manualen beskriver sitt eget användningssätt. Läst 14 september 2026.

[2] Replit Learn / [Scope your next project](#)

Stöd för att avgränsa ett första projekt med tydliga användare, funktioner och sådant som får vänta. Cykelverkstaden och dess krav är AI Manualers egna påhittade exempel. Läst 14 september 2026.

[3] GitHub Docs / [Application card: GitHub Copilot Agents](#)

Stöd för att AI-genererad kod kan innehålla fel och säkerhetsbrister och behöver granskas och testas. Inga produktspecifika funktioner eller åtkomstlöften har överförts till andra tjänster. Läst 14 september 2026.

Nästa manual att plocka upp

Vad är hosting? Förstå vad som behövs när verktyget ska gå att använda via internet.

Data, inloggning och åtkomst. Förstå lagring och vem som ska få se eller ändra uppgifter.

Att äga det man bygger. Planera för kostnader, förändringar och ansvar när verktyget används på riktigt.

AI Manualer är en oberoende utgivare av digitala manualer. Företagsnamn i källorna visar var underlaget kommer ifrån, inte ett samarbete eller en rekommendation att köpa en viss tjänst.