

TERUG NAAR HOME

MAKER MANUALS 01

VIBECODING

Apps en tools bouwen zonder technische achtergrond.

Formaat

Webeditie in neo brutalism-stijl

Licentie

CC BY-SA 4.0

Status

Alpha-editie 0.8



Maker Manual 01

Een praktische gids voor mensen die willen bouwen met AI zonder klassieke programmeur te zijn

Alpha-editie 0.8 - levend boek in ontwikkeling

Dit boek is een tussenstand. Het onderwerp beweegt nog, dus de tekst beweegt mee.

Gebaseerd op het praktijkwerk en de nieuwsbrieven van Erwin Blom, met hulp van AI uitgewerkt tot een versie die verder kan groeien.

Deze tekst is beschikbaar onder de Creative Commons-licentie  .

Je mag deze tekst delen, hergebruiken en bewerken, ook commercieel, zolang je Erwin Blom als bron noemt en afgeleide versies onder dezelfde licentie verspreidt.

INLEIDING

Vibecoding gaat voor mij niet over de hipste tools van dit moment. Over een paar maanden ziet dat speelveld er toch weer anders uit.

Het gaat ook niet over iedereen alsnog programmeur maken.

Het gaat over iets praktischere dingen. Veel meer mensen kunnen software dichterbij hun eigen werk zetten. Ze hoeven niet meer alleen te bedenken wat ze zouden willen. Ze kunnen een eerste versie laten ontstaan, testen waar het wringt en verder bouwen vanuit de praktijk.

Dat maakt het onderwerp interessant. Niet de wow van een demo, wel de vraag wat er verandert als ideeën minder vaak in de idee-fase blijven steken.

Ik schrijf dit voor mensen die niet uit de klassieke programmeerwereld komen, maar wel ideeën, smaak, richting of praktijkkennis hebben. Schrijvers, ontwerpers, ondernemers, docenten, redacteuren, muzikanten, onderzoekers, makers.

De hoofdgedachte van dit boek is eenvoudig. Vibecoding maakt software voor veel meer mensen maakbaar. Daardoor verschuift

de waarde. Minder hangt af van pure uitvoering. Meer van kijken, kiezen, vereenvoudigen, beoordelen en volhouden.

INHOUD

1. Waarom vibecoding ertoe doet
2. Wat vibecoding wel en niet is
3. Begin kleiner dan je denkt
4. Werk vanuit taken, niet vanuit tools
5. Van idee naar eerste bruikbare versie
6. Smaak, oordeel en productgevoel
7. Het speelveld verandert snel, dus jouw houding moet meebewegen
8. Wat blijft als de hype weg is

HOOFDSTUK 1 - WAAROM VIBECODING ERTOE DOET

Sommige veranderingen ogen eerst als een aardige truc en blijken later serieuzer dan gedacht. Vibecoding hoort daarbij.

Aan de buitenkant is het spectaculair. Je beschrijft iets in gewone taal en even later staat er een site, tool of app voor je neus. Als je geen programmeur bent, voelt dat al snel onwerkelijk.

De echte betekenis zit ergens anders. Software komt binnen bereik van veel meer mensen. Schrijvers, ontwerpers, docenten, muzikanten, redacteuren, zelfstandigen, onderzoekers en ondernemers die vroeger wel ideeën hadden, maar voor de uitvoering bijna altijd afhankelijk waren van anderen.

Die afstand is kleiner geworden. Je hoeft niet eerst een team, budget of technische partner op te tuigen om te zien of een idee ergens op slaat. Je kunt sneller testen, sneller bijsturen en sneller merken wat je eigenlijk bedoelt.

Dat verandert ook de rol van niet-technische makers. Je bent niet langer alleen degene met het idee of de briefing. Je kunt veel dichter op het maakproces zitten. Niet als engineer, wel als maker-regisseur.

Ik zag daar de afgelopen tijd allerlei voorbeelden van. Mensen die geen klassieke programmeur zijn, maar wel gewoon iets gingen bouwen zodra ze iets nodig hadden. Een browserextensie om spoilers te vermijden. Een communitysite die beter paste dan bestaande software. Een trainingssite voor een kind in het examenjaar. Een eigen versie van een vertrouwde werktuul omdat de bestaande omgeving niet goed genoeg voelde.

Daar zit voor mij de echte verschuiving. Alles wordt niet opeens makkelijk. Veel meer mensen kunnen wel iets proberen zodra ze een probleem dichtbij huis zien.

De bottleneck verschuift daardoor ook. Minder naar pure technische uitvoering. Meer naar probleemgevoel, richting, smaak, productgevoel en het vermogen om een eerste versie op waarde te schatten.

Vibecoding maakt software dus niet simpel. Vibecoding maakt software maakbaar.

HOOFDSTUK 2 - WAT VIBECODING WEL EN NIET IS

Hoe populairder een woord wordt, hoe groter de kans dat het uitrekt. Dat gebeurt nu ook met vibecoding.

Voor de een betekent het software maken met gewone taal. Voor de ander snel prototypes bouwen met AI. Weer iemand anders bedoelt er agents, automations of een beetje slim chatten mee. In al die beschrijvingen zit iets bruikbaars en ook veel ruis.

De versie die ik het nuttigst vind is simpel: vibecoding is iteratief software maken met hulp van AI, waarbij taal richting geeft aan iets dat gebouwd, getest en verbeterd wordt.

Daar zitten een paar dingen in opgesloten. Er moet iets ontstaan. Je stuurt met taal en feedback. Je kijkt naar wat terugkomt en past het aan. En je komt zelden in één keer op de juiste plek.

Vibecoding is dus niet hetzelfde als chatten. Je kunt prima een uur over een app praten zonder een stap dichterbij een bruikbare versie te komen. Bouwen begint waar taal vorm krijgt.

Het is ook niet hetzelfde als automatiseren. Een workflow laten lopen of een script iets laten afhandelen is iets anders dan een digitaal ding laten ontstaan waar iemand mee kan werken.

En het is geen magische knop. Je typt niet één briljante zin en krijgt vanzelf een goed product. Je moet nog steeds kiezen, begrenzen, bijsturen, testen en soms opnieuw beginnen.

Dat tussengebied maakt het juist interessant. Je bent geen zuivere opdrachtgever en ook niet per se klassieke programmeur. Je zit er middenin.

HOOFDSTUK 3 - BEGIN KLEINER DAN JE DENKT

Vibecoding maakt overmoedig. Dat hoort bij de charme en het risico.

Zodra je merkt dat een tool echt iets kan bouwen, schiet je hoofd vooruit. Voor je het weet denk je aan een platform, een startup of

een compleet systeem, terwijl je nog niet eens weet hoe je een eerste opdracht goed moet neerzetten.

De grootste beginnersfout is bijna altijd dezelfde: te veel tegelijk willen. Een mooi ontwerp, meerdere rollen, logins, opslag, AI in de app zelf, een nette huisstijl, mobiel, een businessmodel. Dan bouw je geen eerste project, dan bouw je frustratie.

Klein beginnen is geen gebrek aan ambitie. Het is de snelste manier om te leren. Je ziet sneller of het idee klopt, of de opdracht scherp genoeg was, of de tool hier goed in is en of jij nog steeds begrijpt wat de kern van het probleem is.

Een goed eerste project is beperkt in scope, direct bruikbaar, makkelijk te beoordelen en niet afhankelijk van vijf systemen tegelijk. Een kleine tool voor je eigen werk. Een simpele site met één duidelijke functie. Een workflowhulp die één irritatie oplost.

Je eerste succes moet tastbaar zijn. Iets waar je op kunt klikken. Iets dat echt een handeling ondersteunt. Dan verandert vibecoding van interessant verhaal naar werkvorm.

Frustratie hoort daar gewoon bij. Tools luisteren slecht, bouwen soms iets anders dan je vroeg of slopen bij een fix weer drie andere dingen. Dat is vervelend en ook normaal. Wie dat accepteert, leert sneller.

HOOFDSTUK 4 - WERK VANUIT TAKEN, NIET VANUIT TOOLS

Een van de snelste manieren om jezelf kwijt te raken in vibecoding is verliefd worden op tools.

Dat is begrijpelijk. De tools zijn de zichtbare ingang van deze hele beweging. Daar zit de snelheid, de verrassing en vaak ook de hype. Alleen zijn tools veel vluchtiger dan taken.

Veel mensen beginnen met de verkeerde vraag: welke tool moet ik gebruiken?

De betere vraag is: wat probeer ik precies gedaan te krijgen?

Wil je een eerste prototype maken? Een kleine workflow automatiseren? Een dataset bruikbaar maken? Een intern hulpmiddel bouwen? Een site publiceren? Zodra dat helder is, wordt de toolkeuze rustiger.

Taken blijven. Interfaces schuiven. Favorieten van vandaag verdwijnen, worden middelmatig of trekken opeens een andere kant op. Daarom is het slim om naast een toolset ook een werkstroom op te bouwen.

Je eigen werkwijze is daarbij een asset. Hoe noteer je. Hoe test je. Hoe beslis je wat de volgende stap is. Waar landt output. Welke dingen wil je kunnen meenemen als een tool straks verandert.

Toolonafhankelijk werken betekent niet dat je puristisch moet zijn. Het betekent dat je voorkomt dat je hele praktijk vast komt te zitten in één hypefase.

HOOFDSTUK 5 - VAN IDEE NAAR EERSTE BRUIKBARE VERSIE

Een idee is nog geen opdracht. Dat is een van de eerste lessen van vibecoding.

“Ik wil een slimme app voor x” is bijna altijd te vaag. Een tool kan daar best iets op teruggeven, alleen leer je er weinig van. Je wilt sneller naar een opdracht die testbaar is.

De werkloop is meestal vrij simpel.

Beschrijf het probleem zo concreet mogelijk. Kies daarna de kleinste bruikbare versie. Laat een eerste versie bouwen. Test

alleen de kern. Kies dan bewust wat je nu doet: doorbouwen, versimpelen of opnieuw beginnen.

Die eerste versie is geen visitekaartje. Het is een testobject. Je kijkt of de logica klopt, of de taak helder genoeg was en of het probleem werkelijk kleiner wordt.

Veel mensen maken hier weer dezelfde fout. Ze willen meteen mooier, slimmer en uitgebreider. Terwijl de echte vraag nog openligt: werkt de kern wel?

Soms moet je doorbouwen. Soms moet je opnieuw beginnen. Dat tweede is niet altijd verlies. Vaak laat het gewoon zien dat je de eerste ronde nodig had om het probleem goed te begrijpen.

HOOFDSTUK 6 - SMAAK, OORDEEL EN PRODUCTGEVOEL

Zodra software makkelijker maakbaar wordt, verschuift het verschil naar iets anders. Naar smaak, oordeel en productgevoel.

Een tool kan iets genereren. Dat maakt nog geen goed product.

Smaak is hier ook niet decoratie. Het gaat om zien wat overbodig is, waar ruis ontstaat, wanneer iets te veel wil en wanneer iets eindelijk eenvoudig genoeg wordt.

Meer output maakt oordeel schaarser. Als iedereen sneller iets kan laten ontstaan, wordt het belangrijker wie kan zien wat de moeite waard is.

Productgevoel zit in datzelfde gebied. Zie je welke stap weg kan? Zie je waar een gebruiker vastloopt? Voel je aan dat iets technisch werkt en toch niet prettig is? Begrijp je wanneer een kleine oplossing beter is dan een grote?

De gebruiker ziet geen prompt. Die ziet een ervaring. Daarom moet je als maker verder kijken dan de slimheid van de tool. Uiteindelijk telt wat iemand ermee kan.

Goed vragen stellen is trouwens ook creatief werk. Een scherpe opdracht is geen administratieve voorfase. Daar begint al veel van de kwaliteit.

HOOFDSTUK 7 - HET SPEELVELD VERANDERT SNEL, DUS JOUW HOUDING MOET MEEBEWEGEN

De naïeve fase van dit onderwerp is voorbij. Iedereen ziet nu wel dat AI en vibecoding iets verschuiven. De vraag is minder of het relevant is en meer hoe je ermee omgaat zonder je hoofd te verliezen.

Het speelveld blijft snel veranderen. Modelmakers en toolmakers schuiven naar elkaar toe. Interfaces worden toegankelijker. Dingen die vorige maand nog spectaculair waren, voelen even later gewoon.

Nieuwsgierigheid blijft nuttig, alleen niet genoeg. Je hebt ook een houding nodig.

Die houding hoeft niet ingewikkeld te zijn. Blijf dichtbij echte taken. Bouw niet groter dan het probleem vraagt. Bewaar wat waarde krijgt. Laat je niet gek maken door elk nieuw speeltje. Kijk steeds of iets je praktijk echt sterker maakt.

Eigenaarschap wordt daarbij belangrijker. Hoe meer je maakt, hoe meer het uitmaakt waar je dingen bewaart, hoe je output terug laat keren en welke delen van je werkwijze van jou blijven.

De makers die hier op termijn het meest aan hebben, zijn vaak niet de luidste testers van iedere nieuwe tool. Het zijn de mensen die

een praktijk opbouwen.

HOOFDSTUK 8 - WAT BLIJFT ALS DE HYPE WEG IS

Elke technologiegolf maakt eerst veel lawaai. Nieuwe termen, nieuwe tools, nieuwe beloftes. Dat lawaai opent ook iets. Mensen gaan anders kijken en durven eerder iets te proberen.

De hype zelf verdwijnt altijd weer. De interessantste vraag is wat er daarna overblijft.

Voor mij zijn dat een paar dingen.

Niet iedereen wordt programmeur, maar veel meer mensen worden maker.

Kleine, gerichte software wordt structureel interessanter. Een tool voor je eigen werk. Een oplossing voor een kleine community. Een interne workflowhulp. Een nicheproduct dat precies goed genoeg is.

Menselijke kwaliteit blijft zwaar tellen. Oordeel, aandacht, smaak, discipline, nieuwsgierigheid en volharding verdwijnen niet naar de achtergrond. Ze worden zichtbaarder.

De behoefte aan een eigen archief en eigen grond onder de voeten groeit ook. Maken zonder geheugen is opwindend. Maken met geheugen is duurzamer.

De beste makers bouwen daarom niet alleen output, maar een praktijk.

Dat is uiteindelijk ook wat dit boek verdedigt. Een makershouding. Wachten tot alles perfect duidelijk is helpt niet. Groter denken dan het werk vraagt ook niet. Verliefd worden op gereedschap al

helemaal niet. Leer kijken. Leer kiezen. Durf iets te maken dat eerst nog niet bestond.

Als de hype weg is, blijven niet de toolnamen over. Wat blijft is dat veel meer mensen software kunnen benaderen als iets wat zij mede kunnen vormen.

MAKER MANUAL 01: VIBECODING - LICENTIE

Dit manuscript valt onder de Creative Commons-licentie CC BY-SA 4.0.

Dat betekent in gewone taal:

- delen mag
- hergebruiken mag
- bewerken mag
- ook commercieel gebruik is toegestaan

Voorwaarden:

- noem Erwin Blom als bron
- vermeld dat het manuscript met hulp van AI is omgezet tot
Maker Manual
- deel afgeleide versies onder dezelfde licentie

Officiële licentietekst:

[Creative Commons Attribution-ShareAlike 4.0 International](https://creativecommons.org/licenses/by-sa/4.0/)