

Maurizio Pelizzone

WordCamp Pisa 2026 – 13-14 novembre 2026

WordCamp PISA 2026

Ideas are Cheap, Execution is Cheaper

Quando scrivere software costa quasi niente, il conto arriva dopo





CHI SONO

Maurizio Pelizzone

Dev Ninja, Vibe Coder, Prompt Engineer, WordPress expert & Speaker

https://www.linkedin.com/in/mauriziopelizzone/
maurizio@mavida.com

- WordPress dal 2004, Mavida dal 2001
- Plugin su misura e integrazioni con CRM e gestionali
- WordPress Meetup Torino, dal 2015
- L'AI nel lavoro di ogni giorno, dal 2023

Oggi aiuto le aziende a far dialogare WordPress con il resto del mondo digitale: lavoro più fluido, intelligente e scalabile.



Quante idee avete nel cassetto?

E quante ne avete realizzate?

Tenetene una in mente: ci torniamo alla fine.

IL VECCHIO ADAGIO

«Ideas are cheap, execution is everything»

1

L'idea è la parte facile

Ce l'hanno tutti: non vale niente finché non esiste.

2

Il limite era il tempo

Per scrivere il codice servivano settimane di lavoro.

3

Serviva un team e un budget

Per questo valeva chi le idee le portava a termine.

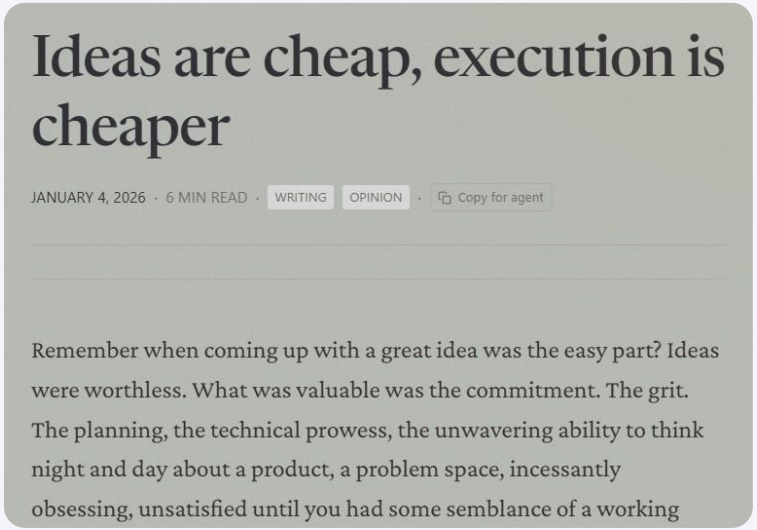
WordCamp PISA 2026

Il ribaltamento



DAVE KISS · 4 GENNAIO 2026

Da dove nasce questo talk



Ideas are cheap, execution is cheaper

JANUARY 4, 2026 · 6 MIN READ · WRITING · OPINION · Copy for agent

Remember when coming up with a great idea was the easy part? Ideas were worthless. What was valuable was the commitment. The grit. The planning, the technical prowess, the unwavering ability to think night and day about a product, a problem space, incessantly obsessing, unsatisfied until you had some semblance of a working

Dave Kiss, davekiss.com, 4 gennaio 2026

GIÀ A GENNAIO

Claude Code come lo strumento con cui realizzare più progetti in pochi giorni di pausa.

«I described them to Claude Code, and it built them.»

L'ARTICOLO

«Ideas are cheap, execution is cheaper»

davekiss.com/blog/
ideas-are-cheap-execution-is-cheaper

DUE FILOSOFIE

Con l'AI, ma in che modo?

VIBE CODING

Andrej Karpathy, febbraio 2025

«Accept All, non leggo i diff»

COSA VUOL DIRE

Descrivi quello che vuoi, accetti ogni modifica senza leggerla e guardi solo se il risultato funziona. Gli errori li incolli all'AI: il codice smette di interessarti.

PROGRAMMAZIONE ASSISTITA

Simon Willison, marzo 2025

«Non committo codice che non saprei spiegare»

COSA VUOL DIRE

L'AI scrive, ma ogni modifica passa da te: la leggi, la provi e la tieni solo se sapresti spiegarla a un'altra persona. Del risultato rispondi tu.

Lo strumento è lo stesso. **Cambia chi ha letto il codice.**

PRO E CONTRO

Quando va bene l'uno, quando serve l'altra

Vibe coding

PRO

- Dall'idea a qualcosa che gira, in poche ore
- Alla portata anche di chi non programma
- Ideale per prototipi e progetti usa e getta

CONTRO

- Nessuno sa cosa c'è dentro al codice
- Bug e falle si scoprono quando è online
- Difficile da mantenere e da far crescere

Programmazione assistita

PRO

- Sai cosa pubblichi e ne puoi rispondere
- Il codice si mantiene e passa di mano
- Gli errori si fermano prima del rilascio

CONTRO

- Più lenta: leggere e provare costa tempo
- Bisogna saper leggere il codice
- La revisione diventa il collo di bottiglia

Scrivere codice costa quasi niente. **Scrivere codice buono, no.**

WordCamp PISA 2026

Un caso pratico: Construct

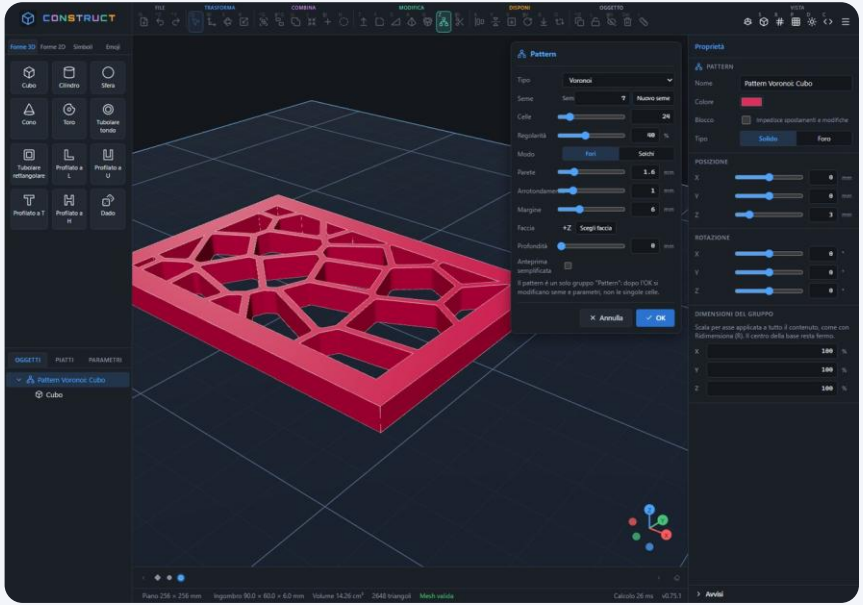


IL CASO PRATICO · POC / MVP

Troppo semplice o troppo complesso

- Un CAD 3D nel browser, pensato per stampare
- Gratis, licenza MIT, senza account
- Funziona anche offline: il calcolo è sul tuo computer

POC / MVP: nessun utente, nessun dato d'uso



construct.mavida.com · github.com/miziomon/construct

IL METODO

Prima del codice, un metodo

1	**Documento di progetto**	Il problema, per chi e cosa non faccio.
2	**Fattibilità e confronto**	Cosa esiste già, dove si ferma, cosa mi serve davvero.
3	**Architettura tecnica**	Componenti, flusso dei dati, dipendenze scelte.
4	**Feature del POC e dell'MVP**	Cosa entra, cosa no, come misuro che funziona.
5	**Piano a fasi**	Ogni fase si chiude con una code review e la suite verde.

In Construct: un solo documento, docs/studio-fattibilita.md (13 sezioni), già nel primo commit del 5 ottobre.

L'APPROCCIO

Prima del codice: ricerca, analisi, studio

RICERCA

Capire il terreno

Cos'è, come funziona, che parole usa chi ci lavora ogni giorno.

ANALISI

Guardare chi c'è già

Software e progetti simili: come funzionano, dove si fermano.

STUDIO

Decidere da dove partire

Cosa prendere, cosa lasciare fuori e in che ordine procedere.

Un CAD 3D è fuori dalla mia comfort zone. Con questo approccio l'ho sviluppato in modo consapevole.

In Construct: uno studio di fattibilità scritto prima della prima riga di codice.

L'ANALISI DEI PROGETTI SIMILI

Chi l'ha già fatto, e come?

Sapere cos'è e come funziona è metà della ricerca. L'altra metà: guardare i software e i progetti simili, con quattro domande.

1

Come funziona?

Lo provo, leggo la documentazione e, se c'è, il codice.

2

Cosa fa bene?

Gli elementi interessanti, da ritrovare nel mio progetto.

3

Dove si ferma?

Il punto in cui smetto di usarlo, e il motivo.

4

Cosa prendo?

Il punto di partenza. Tutto il resto rimane fuori.

Gli elementi scelti sono il punto di partenza del progetto. La scelta resta mia.

Per Construct ho messo a confronto Tinkercad, OpenSCAD, Blender e Fusion, prima di scrivere una riga di codice.

DALLO STUDIO AL PIANO

Fasi piccole, obiettivi scritti prima

Cosa entra, cosa no

L'elenco per il primo rilascio. Quello che resta fuori è scritto subito.

Obiettivi verificabili

Per ogni fase, criteri che si possono controllare: numeri e prove, niente aggettivi.

Una fase alla volta

Ogni fase si chiude e si rilascia da sola, prima di aprire la successiva.

Gli obiettivi scritti prima sono il metro con cui si chiude la fase.

Il piano lo scrive l'AI insieme a me. Le decisioni su cosa entra e cosa no restano mie.

IL CANCELLO TRA LE FASI

Avanti solo con la suite verde

1 · Test prima

Dai criteri misurabili della fase: i test nascono con la fase.

2 · Codice

L'AI implementa, io leggo le diff, una sottofase alla volta.

3 · /code-review

Revisione del codice dell'intera fase. Si corregge prima di proseguire.

4 · Suite verde

tsc, Vitest, Playwright. Un solo rosso e si resta nella fase.

Solo dopo la revisione e con tutto verde si apre la fase successiva.

In Construct, dal 14 ottobre: 13 pull request e 8 commit di correzioni nati da una revisione.

LE REGOLE DI FINE FASE

A fine fase decide un validatore

PRIMA

Gli obiettivi della fase

Scritti prima di cominciare.

DOPO

Quello che è stato fatto

Codice, test, risultato a video.

IL VALIDATORE

Confronta il fatto con gli obiettivi

Un revisore che non ha scritto il codice.

OK

Fase successiva

Gli obiettivi sono raggiunti.

KO

Correzioni puntuali

Si corregge e si rifà il giro.

Se qualcosa non va, il revisore lo fa correggere prima che arrivi alla valutazione umana.

La regola si scrive una volta nelle istruzioni del progetto (CLAUDE.md) e vale per ogni fase.

DOPO LA PRIMA VERSIONE

Solo l'uso dice cosa migliorare

Prima versione

Il piano porta fin qui. Più in là non vede.

L'uso

Usandola saltano fuori i miglioramenti possibili.

Una feature, o un gruppo

Diventa un piccolo piano a fasi, con i suoi obiettivi.

Sviluppo e revisione

Lo stesso giro di prima, fino all'OK del revisore.

Si torna a usarla, e il giro ricomincia

Anche dopo il go-live il processo è lo stesso: piano a fasi, sviluppo, revisione.

Il tema del blog, dopo il go-live: altre otto versioni nate dall'uso, ciascuna con la sua fase.

IL METODO

Chi ha fatto cosa

Ho deciso io

- Il problema e cosa non fare
- Cosa riusare e cosa scrivere da zero
- Le regole di fine fase
- Quando una funzione è finita

Ha scritto l'AI

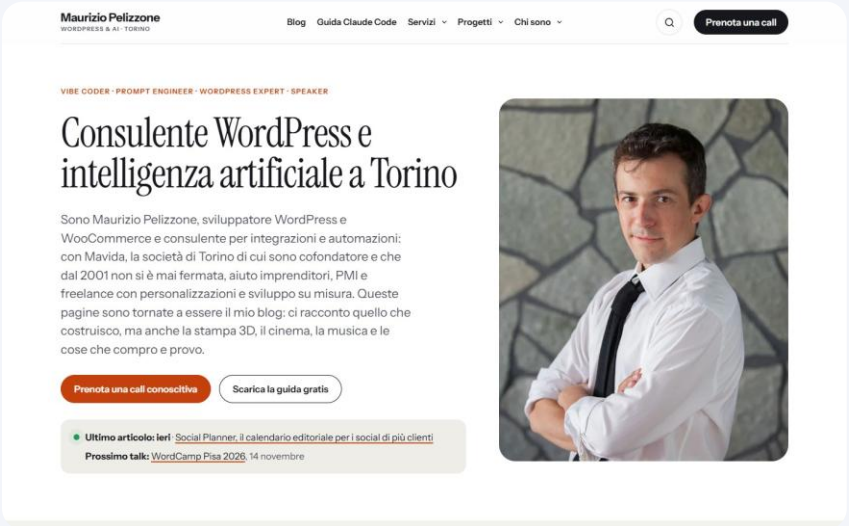
- Quasi tutto il codice
- I test e la documentazione
- Le ottimizzazioni, provate una a una

E QUANDO QUALCOSA NON VA

Il revisore lo trova e lo fa correggere prima che arrivi alla valutazione umana. A me restano le decisioni.

LO STESSO METODO, IN CASA WORDPRESS

Il nuovo tema del mio blog



La home del mio nuovo tema WordPress.

Child di Blocksy, Gutenberg dentro

18 blocchi dinamici su misura, inseribili dall'editor.

Scritto dagli agenti di Claude Code

A me: brief, prototipo, piano e 24 decisioni.

Controlli prima di ogni merge

Coding Standards, PHPStan, 218 test, Lighthouse.

L'ARTICOLO SUL BLOG

maurizio.mavida.com/2026/11/10/tema-wordpress-claude-code/

Con il prompt commentato e una versione da riusare.

CONSTRUCT E IL TEMA WORDPRESS

Due progetti, lo stesso metodo

	CONSTRUCT · CAD 3D	TEMA WORDPRESS · IL MIO BLOG
1 **Progettazione**	Studio di fattibilità e analisi dei software simili	Requisiti e tre scelte di progetto, prima del codice
2 **Prototipo validato**	Un POC per mettere alla prova le scelte più rischiose	Undici tavole HTML, approvate prima di sviluppare
3 **Pianificazione**	Piano a fasi: ognuna si rilascia da sola	Brief, piano approvato, una pull request per fase
4 **Sviluppo**	L'AI scrive una fase alla volta, con i test	Un agente sviluppa sul branch della fase
5 **Revisore**	Code review e suite verde prima della fase dopo	Un agente nuovo dà OK o KO: merge solo dopo l'OK

Funziona anche, e soprattutto, con WordPress. Standard, test e controlli esistono già: il revisore sa che cosa guardare.

IL MESSAGGIO

Se il codice è gratis, cosa vale?

Giudizio

Decidere cosa deve esistere, e cosa no.

Gusto

Riconoscere che una cosa è fatta bene.

Distribuzione

Far arrivare l'idea a qualcuno. Conta più di prima.

Il problema giusto

Scegliere cosa vale la pena provare.

«In a world where you can build anything, the scarcest resource is judgment about what should exist.»

Dave Kiss, davekiss.com, 4 gennaio 2026

WordCamp PISA 2026

Il conto



DEMO ≠ PRODOTTO

Il rischio: scambiare la demo per il prodotto

ARTCRAFT APPS · OTTOBRE 2026

- Sette app open source in Rust che ricalcano la suite Adobe
- Fatte con Claude Opus 5.5, dichiarate «super early alpha»
- La trasformazione libera annulla le operazioni

TRA «FUNZIONA SUL MIO FILE» E «CI LAVORO OTTO ORE AL GIORNO»

- Test
- Casi limite
- Responsabilità
- Manutenzione

Cioè: ingegneria.

Francesco Ronchi, LinkedIn, ottobre 2026 (sezione «Quindi?»)

REGOLAMENTO (UE) 2024/2847

Cyber Resilience Act

10/12/2024

Entra in vigore

11/09/2026

Obbligo di segnalare le vulnerabilità sfruttate: già attivo

11/12/2027

Si applica tutto il regolamento

CHI È IL FABBRICANTE

Chi sviluppa, o fa sviluppare, un prodotto con elementi digitali e lo mette sul mercato. Umano o AI, non cambia.

CHI È FUORI

Software libero non monetizzato, donazioni senza scopo di lucro, progetti per uso personale.

Art. 3 e 71 del regolamento · Non è consulenza legale

COSA CHIEDE A CHI PUBBLICA

Cosa chiede a chi fa vibe coding

1 · SAPERE

Cosa c'è dentro

Componenti, dipendenze, versioni: la SBOM. Construct: 119 componenti, 0 vulnerabilità note.

2 · DIRLO

Accorgersene in fretta

Allerta entro 24 ore, notifica entro 72, relazione finale entro 14 giorni.

3 · RESTARE

Aggiornare per anni

Supporto di almeno 5 anni, con aggiornamenti di sicurezza gratuiti.

Sanzioni fino a 15 milioni di euro o 2,5% del fatturato mondiale (art. 64)

Art. 13, 14, 64 · SBOM di Construct: npx @cyclonedx/cyclonedx-npm --omit dev

LA CHECKLIST

Prima di premere Pubblica

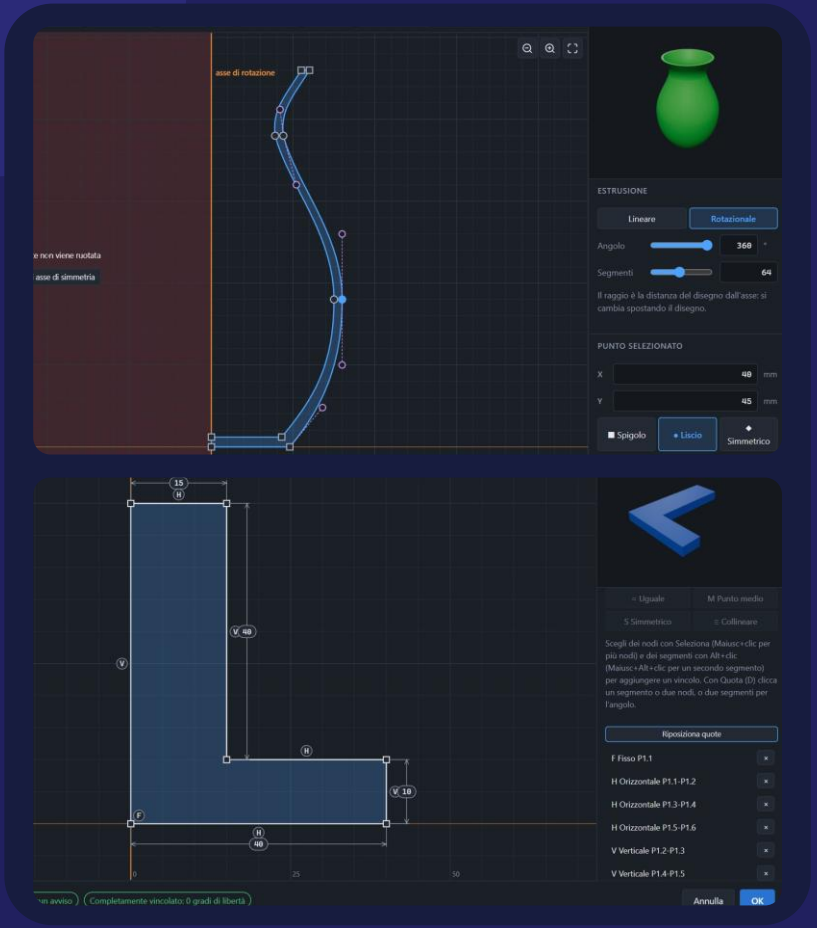
1. So spiegare cosa fa il codice che pubblico.
2. Ho test che fermano le regressioni.
3. So cosa c'è dentro: dipendenze e SBOM.
4. So come mi accorgo di una vulnerabilità e a chi la segnalo.
5. So per quanto tempo lo mantengo.
6. Ho detto agli utenti se è una demo.

DA PORTARE A CASA

Il codice non era il punto

Adesso un'idea costa una sera. Quale provate?

E quando funziona, ricordate che è appena cominciata: chi la pubblica ne risponde.



IL CASO PRATICO · POC / MVP · LICENZA MIT

Construct: un CAD 3D nel browser

Per la stampa 3D: booleane, raccordi, pattern, profili con curve di Bézier, quote e vincoli. Importa ed esporta STL, 3MF e OpenSCAD.

Nessun account, nessun server: il calcolo è sul tuo computer.

construct.mavida.com
github.com/miziomon/construct



GUIDA GRATUITA · VERSIONE 6.2 · CC BY-SA 4.0

Claude Code: una guida pratica

312 pagine, aggiornate al 27 ottobre 2026 e verificate su Claude Code v2.1.311. Novità: i mod, /code-review e /security-review, le parti di un harness. Esempi su PHP, JavaScript e WordPress. Niente hype.

È il materiale da cui nasce il metodo che ho usato per Construct: regole in CLAUDE.md, test, verifica.

github.com/miziomon/claude-code-guide
maurizio.mavida.com/guida-claude-code



Domande?

Grazie per l'attenzione.
Parliamone.

WordCamp PISA 2026

Grazie!



maurizio@mavida.com
maurizio.mavida.com · #WCPISA26