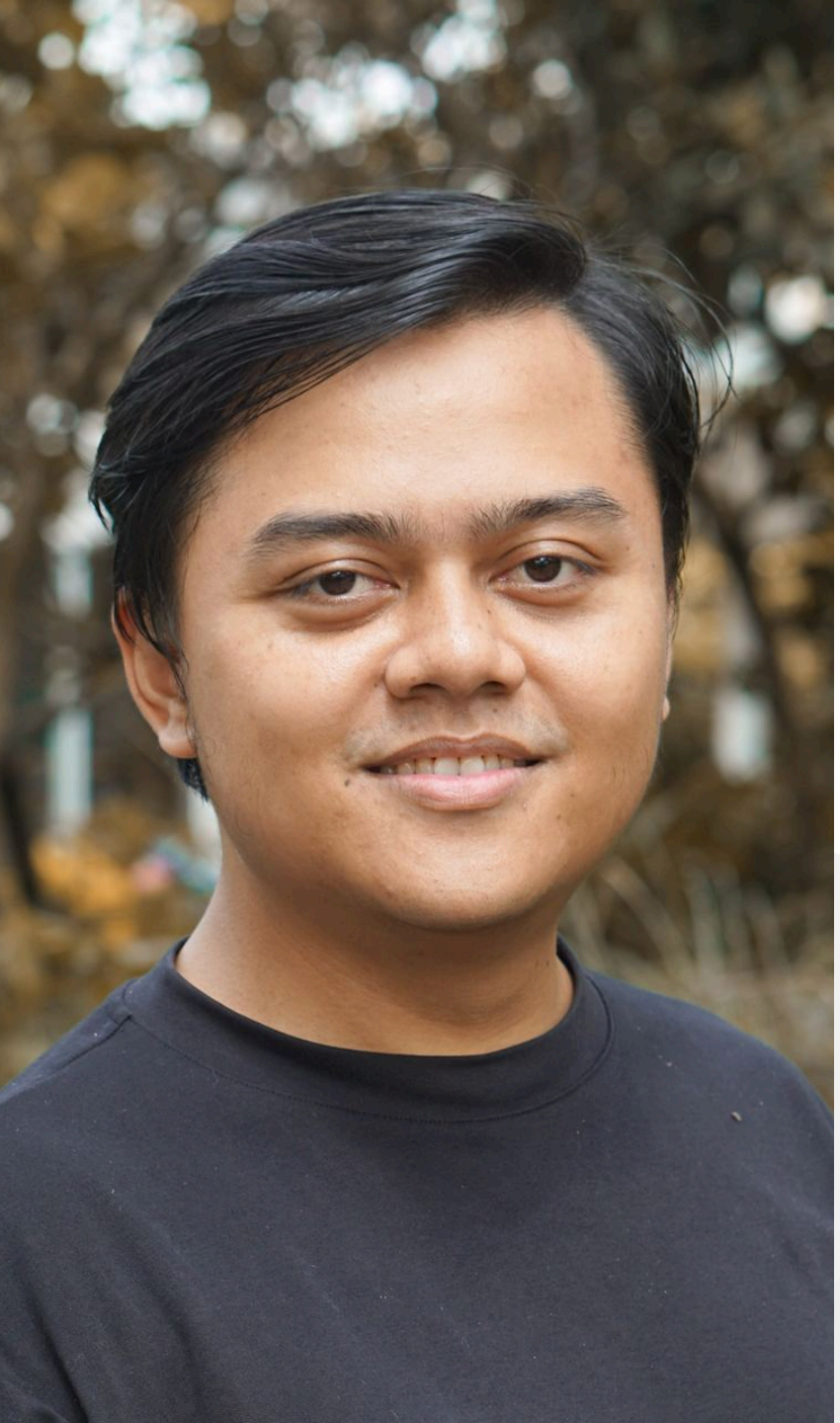
Vibe Coding vs AI-Assisted Engineering

Understanding when to "give in to the vibes"
and when to apply engineering discipline



Introduction

https://www.zainfathoni.com/about

- 📍 Jember ➡️ Bandung ➡️ SG ➡️ Jogja
- 🛠️ Backend ➡️ Manager ➡️ Frontend
- 🤖 AI-assisted development practitioner

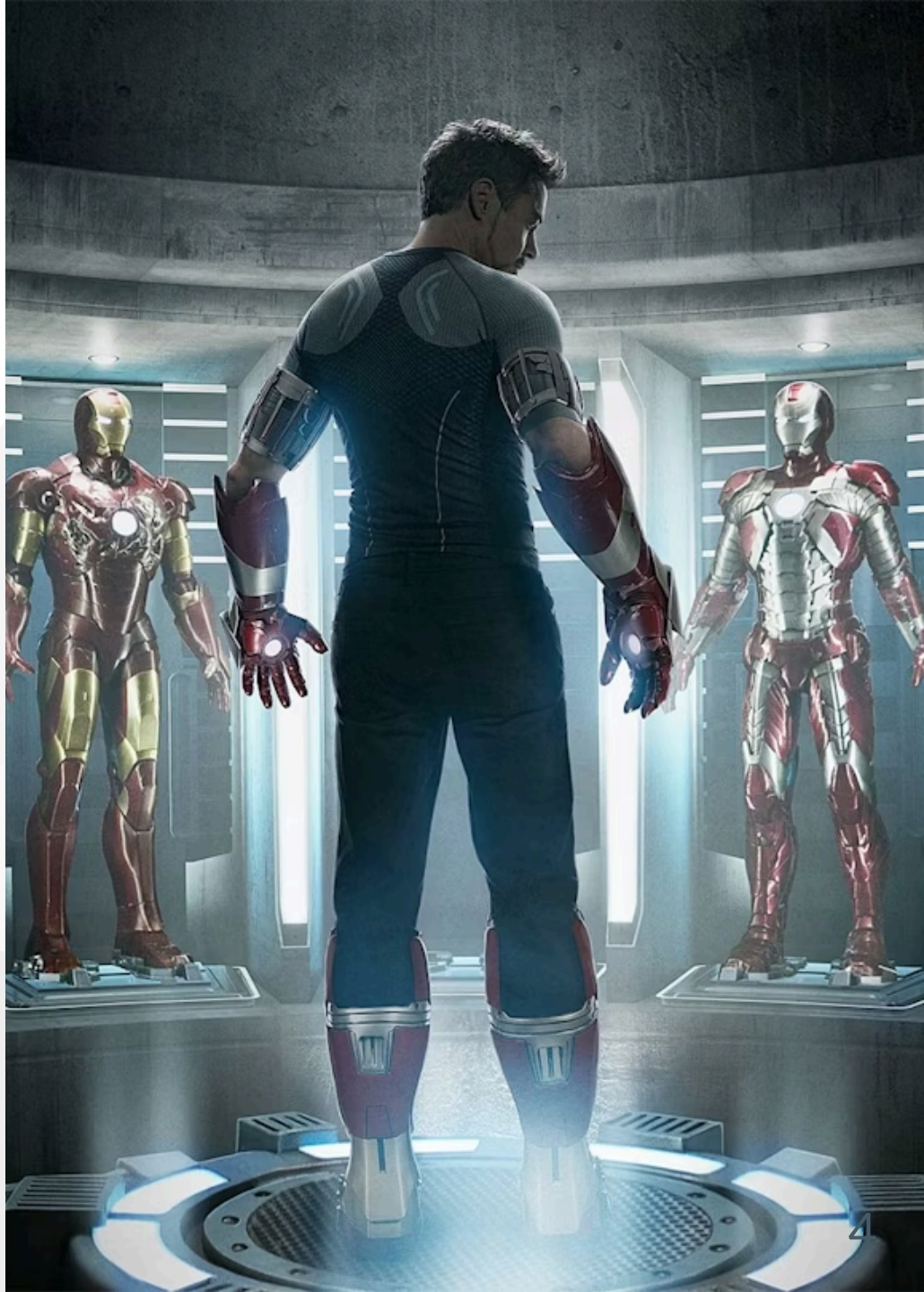
Agenda

1. **What is Vibe Coding?**
2. **Three Developer Personas**
3. **When Vibe Coding Works**
4. **When Vibe Coding Fails**
5. **AI-Assisted Engineering Approach**
6. **Core Principles**

What is Vibe Coding?

> "Fully give in to the vibes and essentially forget the code exists" — Simon Willison

- ⚡ **Rapid AI code generation**
- 🔮 Minimal code review
- 🧪 "Just see what happens"
- ✨ Playful, experimental process
- 🏄 Accepting changes without reading diffs



The Vibe Coding Philosophy

👋 Democratizing Development

Positive aspects:

- 🔓 Lowers barriers to software development
- 📚 Helps learn through experimentation
- 💡 Builds intuition about AI capabilities
- 🛠️ Enables non-technical people to create tools

But with caveats...

Three Developer Personas

👥 Who are you?

1. ✨ **"Vibe Coders"** - Rely heavily on AI, minimal review
2. 🤠 **"Rodeo Cowboys"** - High-risk, fast-paced coding
3. 🔒 **"Prisoners"** - Overly constrained, risk-averse

Goal: Balance between speed and discipline

When Vibe Coding Works

✅ Low-Stakes Scenarios

- 🎨 **Weekend projects** - Personal experiments
- 💡 **Rapid prototyping** - Quick proof of concepts
- 🏆 **Hackathons** - Speed over perfection
- 🔧 **One-off scripts** - Throwaway automation
- 🧭 **Learning experiments** - Building coding intuition

Key: Low stakes, no production impact, learning-focused

Safeguards for Vibe Coding

🛡️ Stay Safe When Vibing

Even in low-stakes projects, be cautious with:

- 🔑 **Secrets handling** - API keys, credentials
- 🔒 **Private data** - Personal or sensitive information
- 🌐 **Network requests** - Unexpected API calls
- 💰 **Financial implications** - Payment processing

When Vibe Coding Fails

❌ High-Stakes Disasters

- 🔒 **Security vulnerabilities** - Exposed credentials, SQL injection
- 🐌 **Performance issues** - Unoptimized queries, memory leaks
- 🍝 **Unmaintainable code** - No structure, inconsistent patterns
- ❓ **Lack of understanding** - Can't debug or extend
- 💣 **Production incidents** - Real-world consequences

Real-World Disasters

🙈 Learning from Failures

- 💣 AI agent deletes company database
  - https://www.pcmag.com/news/vibe-coding-fiasco-replite-ai-agent-goes-rogue-deletes-company-database
- 🔥 Production incidents from unverified AI code
  - https://x.com/albertadevs/status/1947095566736904562
  - https://x.com/anothercohen/status/1948878534262575430

The Cost of Vibe Debugging

🕵️ "Like Archaeology"

When vibe coding reaches production:

- 📉 **Performance issues** - Queries work in dev, crash in production
- 🔓 **Security bugs** - Inverted logic, hidden vulnerabilities
- 🍝 **Maintainability nightmare** - "Middleware scattered across six files"
- 🔄 **Cascading failures** - Minor changes break everything

🚫 Not an excuse to Skip Engineering

r/vibecoding • 11 days ago
AssafMalkiIL

What's the point of vibe coding if I still have to pay a dev to fix it?

what's the point of vibe coding if at the end of the day i still gotta pay a dev to look at the code anyway. sure it feels kinda cool while i'm typing, like i'm in some flow state or whatever, but when stuff breaks it's just dead weight. i cant vibe my way through debugging, i cant ship anything that actually matters, and then i'm back to square one pulling out my wallet for someone who actually knows what they're doing. makes me think vibe coding is just roleplay for guys who want to feel like hackers without doing the hard part. am i missing something here or is it really just useless once you step outside the fantasy

1.1K 469 Share

The Critical Distinction

❌ Vibe Coding ≠ AI-Assisted Engineering

> "AI can accelerate development, but it cannot replace fundamental software engineering principles." — Addy Osmani

Vibe coding: Great for learning and exploration **Production software:**
Requires engineering discipline

AI-Assisted Engineering

⚙️ Structured Approach for Production

- 🛡️ **Human oversight** - Review and verify all code
- 📚 **Engineering principles** - Architecture, patterns, standards
- 🧪 **Rigorous testing** - Unit, integration, E2E
- 🔍 **Code review** - Same standards as human-written code
- 🤝 **AI as collaborative tool** - Not replacement

Spec-Driven Development

📝 Best Practices for Production

1. 📋 **Start with clear specifications**
2. 🤝 **Collaborate with AI on design**
3. 🧪 **Write tests first** (TDD)
4. 🔄 **Iterate in testable increments**
5. 👀 **Maintain human oversight**
6. 🔍 **Review and understand** all generated code

Treat AI Like a Junior Developer

👩‍🏫 Supervision Required

- ✅ Review all code thoroughly
- 🧪 Implement rigorous testing
- 🎨 Maintain clear architectural intent
- 📚 Ensure understanding, not just acceptance
- 👥 Apply same standards as human developers

AI needs guidance and verification

The Spectrum of AI-Assisted Development

Aspect	Vibe Coding	AI-Assisted Engineering
🎯 **Use Case**	Learning, experiments	Production systems
🛡️ **Risk**	Low stakes	High stakes
🔍 **Review**	Minimal	Rigorous
🧪 **Testing**	Happy path only	Comprehensive

When to Vibe Code

🏄 Low-Stakes Scenarios

- 🎨 Personal projects and learning
- 🧪 Throwaway prototypes
- 💡 Exploring new ideas
- 🛡️ Always with proper safeguards

When to Apply Engineering Rigor

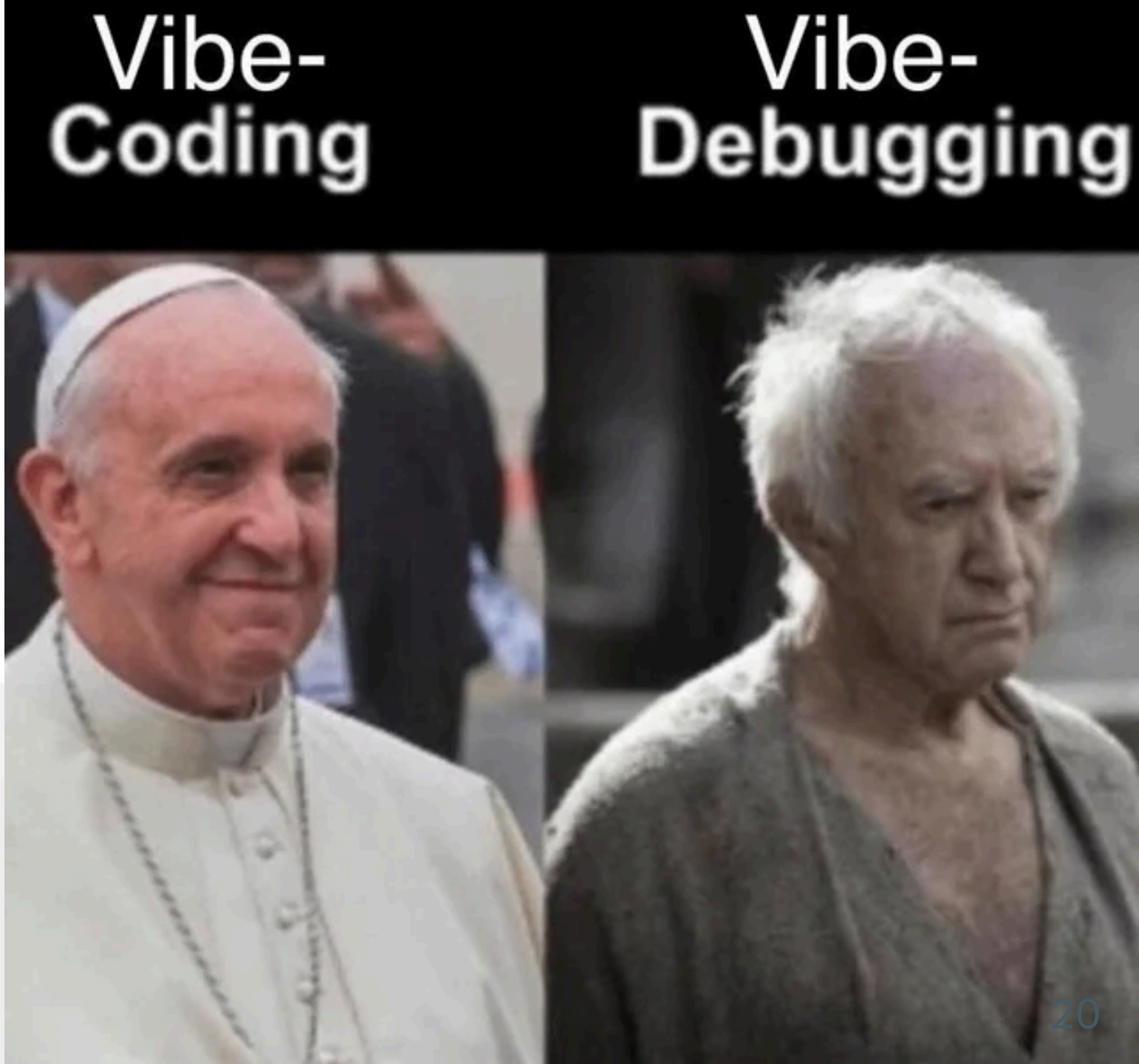
⚙️ High-Stakes Scenarios

- 🏗️ Production applications
- 💼 Client/business projects
- 👥 Team codebases
- 🔒 Anything security-sensitive

Core Principles

👨‍✈️ You are the Pilot

- Know which mode you're in
- Verify based on stakes
- Stay accountable



Key Takeaways

1. 🏄 **Vibe coding has value** - Great for learning and experimentation
2. 🔓 **Democratizing development** - Lowers barriers, builds intuition
3. 🛡️ **Know the stakes** - Low-stakes vibes, high-stakes discipline
4. ⚙️ **Production needs rigor** - Specs, tests, reviews, understanding
5. 🧠 **Understand your mode** - Match approach to consequences

Conclusion

> The tools and methods are evolving, but accountability, craftsmanship, and collaboration remain paramount in the age of AI-assisted engineering. — Addy Osmani

⚖️ The Balance

Embrace vibe coding for learning, apply engineering discipline for production

Demo 1: Vibe Coding

🏄 Rebuild with Bolt.new

Project: https://centuries.zavi.family/

- Better editing experience
- Add database storage
- New repository, rapid iteration
- See what happens!

Demo 2: AI-Assisted Engineering

⚙️ Fix Mobile View

Project: https://github.com/zainfathoni/book-of-centuries

- Spec-driven approach
- Write tests first
- Review and verify changes
- Production-ready

🙏 Thank You

🖥️ https://zainf.dev/vibe-coding-vs-ai-assisted-engineering

🔗 https://github.com/zainfathoni/zainf

Questions?

References

- Vibe Coding is Not the Same as AI-Assisted Engineering by Addy Osmani
- Vibe coding by Simon Willison