

# AI-Enhanced Development



The New Reality  
From 30-Minute Standups to 5-Minute Updates

Jim Vogel  
Lead Developer @ Gravitate

Press Space to continue

What if I told you...

We reduced our daily standups from 30 minutes to 5?  
And made them MORE effective?  
With an AI that never forgets to ask for updates?

# The Standup Bot Story

## Before AI 😞

- 30-minute daily standups
- Chasing updates manually
- Forgetting to update tickets
- Context switching disruption
- Team frustration

## After AI 🚀

- 5-minute sync-ups
- Automated Jira watching
- Slack reminders at day's end
- Morning summaries ready
- Listen during coffee ☕

Result: 25 minutes saved daily × 5 devs = **10+ hours/week**

# My AI Journey

- **Started with Stable Diffusion** - Fascinated by AI art generation
- **Simple automations** - Vim hotkeys, code review scripts
- **Growing sophistication** - As models improved, so did my usage
- **Now: Full AI orchestration** - Agents that handle entire workflows

## The Key Insight

"I'm 10x more productive, happier, and focused on what matters"



# Software ≠ Just Code

## The Full Journey

- **Ideas** → Requirements
- **Stories** → Understanding
- **Problem solving** → Architecture
- **Team communication** → Coordination
- **Testing** → Quality assurance
- **Documentation** → Knowledge transfer
- **Deployment** → Customer value

## Where LLMs Excel

- **Language** understanding
- **Communication** enhancement
- **Documentation** generation
- **Story** clarification
- **Problem** breakdown
- **Team** coordination

## The Reality

Code is maybe **20%** of the software journey

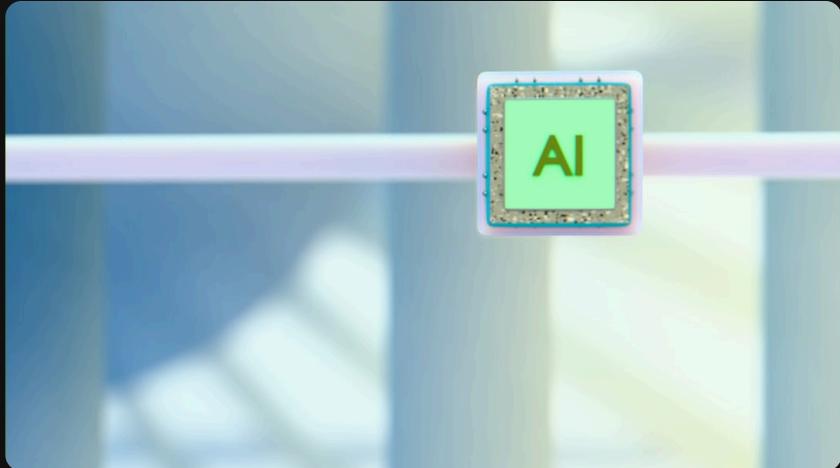
Part 1: The New Reality

# The Cost of Software is Cratering

# The Cost Revolution

## Traditional Development

- High cost per line of code
- Precious, must be protected
- Months to refactor
- Fear of change



## AI-Enhanced Development

- Near-zero marginal cost
- Disposable and replaceable
- Hours to rewrite
- Embrace change



Those who move fast RIGHT NOW have a massive advantage

## Part 2: What This Means for Your Business

# The 10x Developer is Real

But not what you think...

## It's About:

- Eliminating friction
- Focusing on value
- Automating repetition
- Thinking strategically

## My Favorite Example

"I'm not the bad guy anymore - my bot handles code reviews!"

transition: fade

# Rapid Prototyping: The Game Changer

*"Working is more valuable than perfect"*



# The 20-Minute Analytics Agent

1. **Client calls** - "We need an analytics solution"
2. **While they explain** - I'm building in Claude  
Code
3. **10 minutes in** - Basic structure working
4. **20 minutes** - Live demo with their data

## Client Reaction:

"Dumbfounded it was that easy"

## The Value:

Not the code, but the **immediate tangibility**

## What I Built:

- SQL query generator
- matplotlib visualizations
- Analysis summaries
- Slack integration

# From Napkin to Prototype

- **Draw on napkin** (or Excalidraw)
- **Photo → AI** with vision capabilities
- **Specify tools** - "Use React + Tailwind"
- **Get working UI** in minutes

## The Magic:

AI understands design systems and component libraries

## The Result:

Clients can "play with it" immediately



## Part 3: Systems Thinking + AI

Breaking complexity into high-success-rate steps

# Think in Systems

## Example: Code Coverage Bot

**Goal:** Keep coverage above 60%

### System Steps:

1. Watch commits
2. Analyze coverage delta
3. Suggest tests
4. Notify on Slack
5. Track improvements

### Why This Works

- Each step is measurable
- Clear inputs/outputs
- No ambiguity
- Easy to debug
- Humor helps! 😊

"Your commit dropped coverage by 5%. The bot is disappointed. 🤦‍♂️"

# Gall's Law

*"A complex system that works is invariably found to have evolved from a simple system that worked."*

"A complex system designed from scratch never works and cannot be patched up to make it work."

— John Gall, \*Systemantics\*



Start simple, let complexity emerge naturally

## Part 4: The Human Element

What AI amplifies (good and bad)

# AI is an Amplifier

## Amplifies Good Habits

- Clear thinking → Clearer code
- Good communication → Better prompts
- System design → Scalable solutions
- Testing discipline → Comprehensive coverage

## Amplifies Bad Habits

- Laziness → Unreviewed code
- Poor understanding → Technical debt
- No verification → Subtle bugs
- Impatience → Fragile systems



"If you can't explain it, you don't understand it"

# The 60% Problem

"The AI got us 60% there, now let's fix the rest"

## Traditional Mindset

- Code is precious
- Fix what's broken
- Refactor incrementally
- "Don't waste the work"

## AI-Era Mindset

- Code is disposable
- **Regenerate** with better prompt
- Start fresh when stuck
- "Throwing away is faster"



When AI code is 60% right, fix the prompt, not the code

# Warning Signs for Managers

## 1. Developers can't explain their code

- Red flag: "The AI wrote it"
- Good sign: "I used AI to implement my design"

## 2. No code reviews happening

- AI doesn't replace human judgment
- Reviews catch conceptual issues

## 3. Velocity without understanding

- Moving fast in wrong direction
- Technical debt accumulation

## 4. Over-reliance indicators

- Can't debug without AI
- No architectural thinking

## Part 5: Practical Next Steps

How to start tomorrow

# Making AI Work Better

## Prompt Engineering 101

*"Make the LLM interview you"*

- **Have a conversation** - Don't just ask
- **Provide context** - Background, constraints, goals
- **Iterate** - First output is rarely final
- **Show examples** of good output

## MCPs: Integration Superpowers

*Connect AI to your actual tools*

- **Direct integration** - No copy-pasting
- **Example:** AI reads Jira → updates tickets → notifies Slack
- **Context7 MCP** - Latest library docs
- **Unsplash MCP** - Visual content



Better prompts + tool integration = AI that actually helps

# Getting Started: Start TODAY

## Day 1: Foundation

- **Install Claude Code** - \$100/month
- **First hour** pays for itself
- **Pick your champion** - Systems thinker

## Week 1: Level Up

- **Agents** (try fast-agent)
- **Context7 MCP** (fights hallucinations)
- **Unsplash MCP** (visual content)

## Day 2-3: Add Tools

- **Cursor** for coding
- **Linters** for quality
- **First project:** A presentation!

## Week 2: Scale

- Train the team
- Document what works
- Plan next wins



ROI is immediate - Start today!

# Key Takeaways

1. **AI won't replace developers**

But developers using AI will replace those who don't

2. **The cost of software has cratered**

Massive opportunities for those who adapt

3. **It's about amplification, not replacement**

10x productivity is real and achievable

4. **Start small, think in systems**

Even simple automations compound

5. **The time is NOW**

Early movers have exponential advantages

# Thank You

**Jim Vogel**

Tech Lead @  GRAVITATE

AI Stack: Claude Code, Cursor, Custom Agents

Questions?

**Presentation Available:**

[github.com/voglster/modern\\_genai\\_dev\\_pres](https://github.com/voglster/modern_genai_dev_pres)



\*\*This entire presentation was a collaboration between Jim and Claude Code\*\*

\*Including content, images, transitions, and layout\*

Remember: It's not about the tools, it's about the transformation