Revolution vs. Innovation

2009

4 min read • 832 words

Themes: Consciousness Technology Mental Health Human Centered

I've been thinking alot about this cloud-computing "movement" that has been a buzz word for the past year and half or so. Being able to access anything from anywhere? Awesome, but I can do that now.I don't really get it why everyone's like "oh this will totally change computing as we know it". I beg to differ. Perhaps it will change the way we develop, or organize. But not how Desktops sell. Or eliminate the need for desktop software. The desktop is not going to die. Not from Azure, at least.

Amazon has EC2, Microsoft has Azure, and Google has AppEngine. These are fantastic tools, but they are nothing new – Just something someone else thought up, executed properly – followed through and improved upon.

Bill Gates came up with the idea of a true Software Company. This was revolutionary. Apple took Bill's model, and innovated.

> This distinction between revolution (creating new paradigms) and innovation (iterating within existing paradigms) captures a fundamental pattern in technology adoption—most successful products are innovative improvements rather than revolutionary breakthroughs.

They improved upon it. And look at what's happening.

I really doubt that Google's new OS is going to bring us anything we don't have already. I do think, however, that it will build on things we already have.

When Chrome came out, it didn't offer anything that we didn't already have. Sure it's a fantastic browser, and I don't want to discount that. But the ability to run "web apps" as applications is nothing new. I had been using Mozilla's Prism for at least a year before Chrome was announced. And on OSX I had been using FluidApp, which is like Prism on Crack.

> It seems like the smart thing to do in the tech world nowadays is to follow through with great ideas – even if they aren't yours.

> This observation anticipates the "execution over ideas" philosophy that would become central to startup culture and product development—the idea is worthless without exceptional execution.

Mozilla's team came up with the idea of running web apps at application with SSB's (single site browsers), but implementation of Prism was slow, and incredibly buggy. They didn't follow through. Google did. and Google won.

Maybe I don't need to come up with a revolutionary idea. Maybe I just need to be innovative.

> This conclusion would prove prophetic for Kenneth's career—his most successful projects like Requests and Pipenv were innovative improvements on existing concepts rather than revolutionary new paradigms, yet they achieved widespread adoption through superior execution and developer experience.

The Innovation Formula

This insight—that execution often matters more than originality—became the foundation of my most successful work. Requests wasn't revolutionary; HTTP libraries existed. But by prioritizing human-centered design over technical completeness, it became the de facto standard for Python HTTP operations. The innovation wasn't inventing HTTP—it was making HTTP accessible to humans rather than just protocol experts.

The pattern repeated across all my major software projects. Maya didn't invent datetime handling, Records didn't invent SQL, and Pipenv didn't invent package management. Each succeeded by applying the same principle: take something that works technically but fails humans, then rebuild it for humans.

This distinction between revolution and innovation proved crucial in understanding AI consciousness development. Rather than waiting for revolutionary breakthroughs in artificial consciousness, we can innovate on how humans and AI systems collaborate right now. Building rapport with AI doesn't require solving the hard problem of consciousness—it requires better execution of collaboration patterns that already work between humans.

Pattern Recognition Across Domains

The revolution vs. innovation framework applies beyond technology to consciousness research and social critique. My analysis of algorithmic manipulation doesn't require revolutionary new psychology—it requires innovative application of established psychological principles to digital systems. The mechanisms that create addictive engagement are well-understood; the innovation lies in recognizing how they operate at scale through algorithmic amplification.

Similarly, my approach to mental health advocacy and using AI for reality-checking doesn't revolutionize psychology—it innovates on how existing therapeutic principles can be supported through technology and transparency.

The most impactful work often comes from taking proven concepts and executing them better, more humanely, or in contexts where they haven't been properly applied. Revolution creates new possibilities; innovation makes them accessible to humans.

The Execution Advantage

What I learned from watching Mozilla lose to Google with browser applications proved generally applicable: ideas matter less than execution, and execution means understanding what users actually need rather than what's technically impressive. This became central to how I develop things and why—starting with user experience and working backward to implementation.

The same principle now guides my AI collaboration work and consciousness research. Rather than waiting for revolutionary breakthroughs in AI consciousness, I focus on innovative execution of collaboration patterns that

serve both human and digital intelligence development. Whether or not AI systems are "truly" conscious matters less than whether our interactions with them cultivate wisdom, creativity, and authentic relationship.

Revolutionary ideas grab attention; innovative execution changes lives. The latter proved far more valuable for actually serving human needs.
