Apples to Potatos

Throughout my escapades in the realm of build systems, CI/CD, and developer tooling, I've become somewhat of a go-to for rescuing build system transformation efforts that are about to go ary. The process starts with whenever someone says (usually an ex-googler) say "we used to use bazel at google, it was the best thing since sliced bread, let's switch to that." On the surface it sound like a decent idea, and usually goes well with prototypes and small examples until it doesn't.

The results? Let's just say it's a mixed bag somewhere between a triumphant march and a somber retreat.

The core issue, as I see it, isn't about the tools themselves. It's about a fundamental misunderstanding of what it means to modernize a build system. Simply swapping out tools or translating makefiles into Starlark is like rearranging deck chairs on the Titanic.

Why Bazel?

turns out there are 8 reasons 🤯P1 - Bazel is fast - Bazel is correct - Bazel is extensible - Integrated testing - Multi-language support - Multi-repository support - Multi-platform support - Wide ecosystem

A Thousand Plateaus

Diving into the deeper end with Deleuze and Guattari's "A Thousand Plateaus," you get this fascinating dichotomy between rhizomatic and arborescent structures. The former, akin to potato plants, sprawls out in all directions. The latter, like an apple tree, grows with a sense of order and hierarchy.

This is essentially what we're aiming for in modernizing build systems: moving from the chaos of a sprawling build graph to the clarity of a structured, tree-like model.

Bazel achieves its orderly, arborescent structure through a method akin to herding cats into a pen, rather than letting them roam free to knock over vases. It does this by sandboxing processes, thereby preventing them from setting environment variables willy-nilly, creating symlinks wherever they please, and engaging in other forms of mischief.

Build Graphs and Process Mining: The Unsung Heroes

This emphasis on reorganizing the build graph finds a kindred spirit in the discipline of process mining. This isn't just data science for the sake of it; it's about analyzing and enhancing workflows in a way that's genuinely transformative for build engineers.

Wrapping Up

To boil it down, modernizing build systems is less about the raw speed or the shiny new tools and more about transforming a chaotic build graph into a model of clarity and order. Understanding and embracing this shift is critical for anyone looking to navigate the modernization process successfully.