



November 21, 2024

Allergic to Gambling

BUSINESS INSIGHTS

Written by: Max Hoaglund, Senior Technology Lead

In a fast-paced professional services or agency type environment, the line between experimentation and gambling blurs a bit. It's common for software projects to succeed or (mostly) fail seemingly because of a key gambit—maybe a snap choice of a new or unproven technology made by a hungry manager. Leaders and developers look to strike while the iron is hot and apply their bright idea to a key problem, and it can be a struggle to stay grounded and empirical.

On the one hand, this is a cringingly uncomfortable framing of software projects and the contexts they happen in—it's almost unfair. We're all working in a landscape of fiefdoms where products we rely on are constantly changing, being outmoded, disrupted, and replaced. It should come as no surprise that some choices are impulsive, and many teams struggle to slow down even when they're on a clearly wrong trajectory. We see this all the time, and it's painful. People are born to dream big, chase sunk costs, and jump over red

tape. The software projects of humankind are a great place to witness these traits and the tragedies they cause.

On the other hand, by acknowledging the risk many organizations take on by adopting a given technology or spinning up (and pinning their hopes on) a slick new team, we gain an opportunity to reframe this risk. Experimentation and gambling are both fundamental interactions with risk, and the difference between them is in the apprehension and expectation of the final result. At Clientek we experiment. We're allergic to gambling.

Gambling can be distilled down to the notion of hoping for the best while ignoring the odds. It (often disastrously) instrumentalizes innate human experiences around chance and anticipation to drown out the small rational voice that weighs the real chance of winning. Generally, the jackpot is of a specific size, called out and promised in advance.

The jackpot will pay out one day, who knows when—but it'll be great, and everything will change.

Experimentation promises only to evaluate a postulate, and it relies on the quiet voice that calls for patience, reflection, and measurement. In a software project, there is no room to invest in moonshot promises. There is only room to identify objectives, investigate ways to achieve them, provide a postulate (in the form of an execution plan), and run the experiment. Wash, rinse, repeat. It does you no good to drum up excitement about what you and your organization could achieve in two years if you can't achieve the smallest part of that vision in two months.

We want our clients to have something to show for the years they might spend on a software initiative—that's why we're so obsessed with measuring what happens in every two-week period. Every sprint is an experiment. When you can conduct, learn from, and benefit from dozens of these experiments every month, you don't need a jackpot.

CONTACT US

