Tabula Rasa
Pitfalls and Possibilities with Clean Slate Initiatives

Arnold W. Bragg, Editor in Chief

Tabula Rasa: [Latin] (1) clean slate; (2) blank tablet; (3) a fresh start; (4) a new beginning.

Suppose you could travel back in time to your first day in high school, knowing everything you know today. Your first task would be to convince Dad to mortgage the house and buy stock in IBM (or DEC or Microsoft or Cisco or Google, depending on your age). Your next task would be to decide how to avoid the painfully stupid mistakes you made at your alma mater. You have a clean slate. Nobody remembers the cherry bomb in the locker because it hasn’t happened yet (and you’ll be sure to outrun the principal this time).

I love clean slate initiatives. The concept is simple: someone decides that an IT system (or architecture, infrastructure, process, policy, or technology) is so hopelessly and fundamentally flawed that we can’t overcome its deficiencies by simply addressing and correcting them one by one—some deficiencies might be entangled with others in a spooky, quantum-esque manner such that mending one perturbs the others in ways we can’t imagine—and a fresh start seems to be the only way out of the quagmire.

Clean slate approaches are also useful when a system reaches its technological limits and can’t be enhanced or extended. Wireless architectures, for example, often rely on fresh starts when moving from one generation to the next. Clean slate initiatives are also driven by market forces, legal pressures, and various other forms of voodoo.

A clean slate shuns most incremental and backward-compatible enhancements in favor of new solutions that build on lessons learned. Among the clean slate approach’s pitfalls, however, the worst is to insist that the solution be “unconventional,” “revolutionary,” “elegant,” or “flamboyant.” My colleague shouts down clean slate proposals as “impossible to build, impossible to sell, impossible to deploy, and impossible to use!” Her point is clear: the clean slate solution must be realistic—the IT graveyard is full of technically superior solutions that were too impractical or impolitic to survive.

Transferring from Latin class to Spanish makes sense for our time traveler; transferring from Latin to Esperanto (despite the fact that it was designed to serve as a universal second language) doesn’t.

Another pitfall is to expect the clean slate solution, by definition, to be significantly “better” than an evolutionary solution. Georgia Tech’s Constantine Dovrolis uses a biological argument to make his case that an evolutionary approach leads to less costly and more robust designs (“What Would Darwin Think about Clean-Slate Architectures?” ACM SIGCOMM Computer Comm. Rev., vol. 38, no. 1, 2008). He notes that evolutionary designs solve problems without breaking the existing systems, and thus needn’t be short-sighted or simplistic. Robustness trumps optimality.

Retrospection is key to avoiding these pitfalls. Before we begin anew, we absolutely must understand where we are and how we got here, and we must draw on that knowledge and experience as we decide what to do differently. Effective clean slate approaches are not memoryless. As Dovrolis says, we need a clear understanding of “the context or environment in which a proposed solution will be deployed, as opposed to designing … in a vacuum” (p. 34). Although this understanding of context is inherent in evolutionary designs, it must also play a major role in clean slate initiatives.

Paraphrasing George Santayana, if our time traveler can’t remember his high school mistakes, he is condemned to repeat them. Won’t that be fun?
A clean slate approach also requires clairvoyance. A new architecture or system takes $M$ months to design, implement, test, and deploy. We therefore need to “guesstimate” where today’s solution will be in $M$ months—it will almost certainly evolve, but we don’t really know how. We must not compare the clean slate design, $M$ months from reality, with today’s flawed solution. Consider, for example, IPv6, which is in some ways a clean slate architecture. People still compare it with IPv4 circa 1996. One reason for IPv6’s painfully slow adoption is that IPv4 has evolved dramatically during the past decade. Comparing the clean slate design to the older flawed version obscures the true value of the new solution.

Our time traveler considers dropping band in favor of co-ed volleyball to improve his social life, but he has no volleyball experience on which to build. Before switching, he must assess potential incremental improvements in his band persona, based on introspection and his new gift of clairvoyance, with realistic volleyball expectations. Does he know, for example, that the basketball team plays volleyball in the off-season?

And what of our time traveler? His success depends on how well he applies the knowledge he takes back with him in his effort to engineer something better: incremental improvements will be a significant part of his new beginning. (He also needs to drop physics because time travel just might violate the first and second laws of thermodynamics.)

Readers may contact editor in chief Arnold Bragg at awbragg@yahoo.com.