



Problems as Change Agents

The idiom, "If it ain't broke, don't fix it!" is at the heart of Engineering. It makes sense:

- If a device is functioning within specification, there is no need to tune it.
- If software is working as required, one should not mess with its code.
- If curriculum is achieving its goal of teaching students exactly what they need to know about the subject matter, there is no need to revise it.

“ ..when are we to think outside of the metaphorical box? ”

However, if this is correct, then when are we to pit our imagination against our knowledge by thinking outside of the metaphorical box in order to ideate about new revolutionary technological advancements? When are we to design the next great thing no matter its size? When are we to upgrade? Well, perhaps there is no better time than when we get the unique opportunity presented to us by the discovery of a problem.

By definition, a problem arises when the system is not performing as envisioned, designed, or developed. Otherwise, it is not a problem, but rather a new feature or an update to an existing feature slated for next version release. The concept of "versioning" in software development takes full advantage of that distinction

“ The entrepreneur sees a problem as...an opportunity to evolve in an otherwise mathematically harmonious system. ”

in the minds of producers and consumers. Many software products are pushed into the market with known, but not critical, problems that are gift wrapped with a promise to consumers of new features and swift updates in order to grab market share before competitors do. This trick works because humans often resist change. Customers who have spent money on a system are willing to spend even more

money to update and patch their existing system with service packs and workarounds rather than uproot and replace their system with one that is clearly superior.

Perhaps, this is the difference between one who is an entrepreneur and one who is not. The entrepreneur sees a problem as a change agent or an opportunity to evolve in an otherwise mathematically harmonic system.

Navid Shaghghi

Lecturer and Researcher Ethical, Pragmatic, and Intelligent Computing (EPIC) Laboratory
Departments of Math & Computer Science (MCS), Computer Science & Engineering (CSEN), Bioengineering (BIOE), and Information Systems and Analytics (ISA)
College of Arts & Sciences, School of Engineering, and School of Business