Scott Rosenberg's "Big Ball of Mud" essay weighs the pros and cons of clean and elegant architecture in software development. An ideally organized, perfectly human-readable program is likened to a well-built building with perfect architecture. Most programs, however, are more like "Big Balls of Mud," with messy, haphazardly assembled code.

Rosenberg makes the assertion that "Big Balls of Mud" are prevalent largely because they work and they're easier to produce in the short-term. It's a bit more complicated than that, however. In the broad scope of design decisions, elegance of design is an important, but not a crucial aspect of the functionality code. The "Big Ball of Mud" analogy, while sometimes used as a derogatory critique, actually just describes one of many viable design styles.

Where Rosenberg's opinion on "Big Ball of Mud" type program differs from architecture-first design philosophy is his suggestion that functionality trumps elegant design in the short-term. Because sloppy but functional programs get to market sooner, they get market feedback sooner. Feedback from users, is ultimately one of the most important aspect of software design, and can fix many of the inevitable issues caused by any design method.

I agree with Rosenberg's thesis. There is a certain type of program for which "Big Ball of Mud" design is not only acceptable, but a better choice, when you consider the time and budget requirements of investing in solid, elegant architecture. We are entering an era of software development where code is becoming more and more complex. I believe we will reach a point where human-readability of code is not practical, maybe not even possible. I don't think the principles of software architecture will be abandoned, but they will very soon take the back seat to functionality and ease of production.