This was the kind of system that would vex a traveling salesman. In fact, the architectural similarity to the London Underground was remarkable: there were many routes to get from one end of the system to the other, and it was rarely obvious how best to do so. Often a destination was geographically nearby but not accessible, and you wished you could bore a new tunnel between two points. Sometimes it would have actually have been better to get out and take a bus. Or walk.

That's not a "good" architecture by any metric. The Metropolis's problems went beyond the design, right up to the development process and company culture. These problems had actually caused a lot of the architectural rot. The code had grown "organically" over a period of years, which is a polite way to say that no one had performed any architectural design of note, and that various bits had been bolted on over time without much thought. No one had ever stopped to impose a sane structure on the code. It had grown by accretion, and was a classic example of a system that had received absolutely no architectural design. But a codebase never has *no* architecture. This just had a very poor one.

The Metropolis's state of affairs was understandable (but not condonable) when you looked at the history of the company that built it: it was a startup with heavy pressure to get many new releases out rapidly. Delays were not tolerable—they would spell financial ruin. The software engineers were driven to get code shipping as quickly as humanly possible (if not sooner). And so the code had been thrown together in a series of mad dashes.

> NOTE
> **Poor company structure and unhealthy development processes will be reflected in a poor software architecture.**

Down the Tubes

The Metropolis's lack of town planning had many consequences, which we'll see here. These ramifications were severe and went far beyond what you might naïvely expect of a bad design. The underground train had turned into a roller coaster, headed rapidly downward.

Incomprehensibility

As you can already see, the Metropolis's architecture and its lack of imposed structure had led to a software system that was remarkably tricky to comprehend, and practically impossible to modify. New recruits coming into the project (like myself) were stunned by the complexity and unable to come to grips with what was going on.

The bad design actually encouraged further bad design to be bolted onto it—in fact, it literally forced you to do so—as there was no way to extend the design in a sane way. The path of least resistance for the job in hand was always taken; there was no obvious way to fix the structural problems, and so new functionality was thrown in wherever it would cause less hassle.

Download at WoweBook.Com