

A Tale of Two Systems: A Modern-Day Software Fable

Pete Goodliffe

Architecture is the art of how to waste space.

—Philip Johnson

A software system is like a city—an intricate network of highways and hostelryes, of back roads and buildings. There's a lot going on in a busy city; flows of control are continually being born, weaving their life through it, and dying. A wealth of data is amassed, stored, and destroyed. There are a range of buildings: some tall and beautiful, some squat and functional, others dilapidated and falling into disrepair. As data flows around them there are traffic jams and tailbacks, rush hours and road works. The quality of your software city is directly related to how much town planning went into it.

Some software systems are lucky, created through thoughtful design from experienced architects. They are structured with a sense of elegance and balance. They are well-mapped and easy to navigate. Others are not so lucky, and are essentially software settlements that grew up around the accidental gathering of some code. The transport infrastructure is inadequate, and the buildings are drab and uninspiring. Placed in the middle of it, you'd get completely lost trying to find a route out.

Where would your code rather live? What kind of software city would you rather construct?