

What is Software Engineering?

What is “software engineering”?

The answer in my own words: Software engineering is sequencing instructions to perform and automate computer operations.

I’ll explain how I got there...

A textbook definition is that software engineering is the process of designing, developing, testing, and maintaining software systems.

But that obviously doesn’t get us very far. What’s a “software system”? A software system is a collection of Software Components that work together to provide some functionality at its interface (the part where you interact with it).

So, what’s a “software component”? A software component is a collection of instructions that are used to operate a computer.

To understand what an “instruction” is, we need to understand what a “computer” is and how it operates. A computer is just a bunch of switches called “transistors”—the only meaningful difference is that transistors don’t have moving parts to start or stop the flow of electrons through a circuit like switches do. These *switches* are either flipped on or off. Each of the two states (“on” and “off”) are called a “bit.” Unique permutations of those bits either represents some unique information (“data”) or a unique operation you can perform on that data. And an instruction is just a unique permutation of those bits—in other words, a unique sequence of “on” and “offs”.

Creating instructions allows us to automate operations. Instructions can be sequenced themselves to perform more complex operations on data. These operations include things like retrieving, processing, storing, and transferring data.