Software Engineering, not Computer Science

It is often debated whether software development is art or science. While it can be argued either way, they ask the wrong question. What should be asked is “what should professional software development be?” It should unquestionably be engineering, though it is not right now.

Science and engineering are two fields that many fail to distinguish from one another. While both fields learn what is true, scientists use this knowledge to test hypotheses and extend knowledge in their field, while engineers use it to solve practical problems. Scientists are held responsible only by other scientists, while engineers are held responsible by the public. While an undergraduate degree in science prepares students to continue their research, an undergraduate degree in engineering preprares students to join the workforce.

Computer science students often find themselves in a poor situation when they are performing the role of engineers. They often focus on the wrong aspects of development, not recognizing the need to be efficient in what they do. With too little educational programs focused on software engineering and too high of a necessity for software development jobs, this was the inevitable result.

Engineering is often seen as too formal a term for “computer programming”. Many don’t understand that the basis of engineering is on practicality. Development, as an engineer, should be focused on taking the steps appropriate for the project. While a broken program is definitely not a preferred outcome, neither is a program that had too many resources waste on it. A part of the goals is that it would be run with minimal defects and ensure maximum satisfaction, but the other half is that it needs to accommodate the short schedule, a small team size, and often a low cost.

Today’s world of software development doesn’t look much like engineering, though it should. If we ask “should software development be engineering”, and taking our thoughts down that line, we can improve on the industry, finally answering the most important question of all: “what will the software industry look like after all these questions have been answered?”