[DP79] De Millo, Lipton, Perlis. 1979. Social processes and proofs of theorems and programs. Commun. ACM 22, 5 (May 1979), 271-280.

The paper about the social processes that take place in mathematics regarding proofs and how this process can or cannot be applied to verification of computer programs. At first it is argued that computer programming should become more like mathematics. One instance how that can be done is by automatic program verification.

Maths is a social, informal, intuitive, organic human process, a community project. It uses proofs to verify theorems. Other than what its name suggests, a proof is not actually what the layman thinks it is: a definitive proof that has to be absolutely correct. In maths, a proof can only probably express the truth. These proofs and theorems have to be believed in. This belief is created by a social process that tries to make mathematicians feel confident about a theorem or a proof.

The social process starts with a proof. A proof is just a spoken message or at most a sketch. If the proof generates no excitement among friends and collegues if is most probably discarded. However, if the proof generates excitement, a polished version will be made. After this polished version gets published — it is not uncommon that papers get rejected from journals — more people will read it. After a certain cooldown period, if a larger audience likes the proof, they may belive in it. Then, the believers will paraphrase the proof and create multiple versions of the proof or the theorem. This will help others to understand the original authors ideas. A believable theorem gets used as part of other proofs and in the real world, i.e. bridges are build by inserting actual numbers in the formulas. All these steps improve the confidence that the theorem or proof is correct. After enough battle-testing, the theorem is not longer thought to be probably true but absolutely true. It is included in the body of knowledge of mathematics.

This social process works very well in mathematics. There are however doubts that program verification will work the same way or if it will work at all. It is argued that program verification will not play the same role in Computer Science as proofs in maths do. Verifications cannot really be read by a person. They are too long, even for simple programs. Thus, they cannot acquire credibility gradually (as in maths); either completely believe the result of the verifier or don't believe in verification at all. At this point, the analogy of program verification and mathematical proof fails as there are too many differences.

The reason why program verification will fail is because it does not have the same preconditions regarding programs (proofs) like maths. The missing social process is just a manifestation of that problem.

[H+04] Hevner, March, Park, Ram. 2004. Design science in information systems research. MIS Q. 28, 1 (March 2004), 75-105.

Text...