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

The paper is about the social processes that takes 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. It is just a spoken message or at most a sketch. If the proof generates no excitement among friends and collegues it 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 like maths with proofs. 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.

The aim of the paper is to inform the community of Information Systems researchers about how to do research in Information Systems (IS).

In general, there are two research paradigms: behavioural science and design science. Behavioural science has its roots in the natural sciences. It seeks to develop and justify theories and aims at the truth. Design science has its roots in engineering. It seeks to create innovations or artifacts and aims at utility.

The paper then introduces a framework for doing IS research using design science. It says that regarding a top-down approach, research must address the interaction between business strategy, Information Technology strategy, organizational structure and IS infrastructure.

Given a business need, IS research is conducted. It uses the knowledge base (foundations and methodologies) of IS to solve the business need. The development is done in two complementary phases. In the first, design science develops artifacts and theories. In the second, behavioural science assesses, justifies and evaluates these artifacts and theories. The gained knowledge will be used back in design science in order to refine artifacts or theories. Finally the artifact can be used in the business environment. Knowledge gained by building the artifact is added back to the knowledge base for future research and application.

Lastly, the paper introduces seven guidelines for doing design science research in IS: (1) Research must provide a viable artifact, (2) Research must solve a business problem, (3) Quality, utility and efficacy of the artifact must be demonstrated, (4) Research must provide contributions back to science, (5) Rigorous methods in both development and evaluation must be used, (6) Design using available means while including the issues of the target environment, (7) Communicate research results in a suitable way both for management and technology people.