[S05] Santini, S. (2005). We Are Sorry to Inform You... Computer, 38(12), 128-128.

“We are sorry to inform you” is a sentence a paper author might receive when having his paper returned from a peer review. This indicates that the authors paper was rejected from a journal. Having the whole publication process in mind, the authors suggest that getting a notice of rejection is sometimes just the result of a peer reviewer having a bad day.

The article shows some peer reviews for fundamentaly important papers (and thus influential authors) in computer science. However, all of the reviews reject the papers due to various reasons. In the following I summarize the reviewers critique.

1. **E.W. Dijkstra: Goto Statement Considered Harmful:** Too academic / purist, objects current practices
2. **E. F. Codd: A Relational Model of Data for Large Shared Data Banks:** Impractical / complicated (“foreign keys”), no evaluation, overly formal
3. **A. Turing: On Computable Numbers, with a Application to the Entscheidungsproblem:** Overly formal, no application
4. **C. E. Shannon: A Mathematical Theory of Communication:** No application, overly formal, not sexy
5. **C. A. R. Hoare: An Axiomatic Basis For Computer Programming:** Too preliminary
6. **R.L. Rivest, A. Shamir, L. Adelman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems:** Not practical, objects current practices (“existing encryption standards”).

To conclude, looking back in history, the reviewers were maybe right with their opinion. However, research is often times not applicable in the short-term but really shines when picked up decades later. An interesting question remains: How much damage is done today by rejecting papers and their reviewers?

[J+93] Johnson, R., Beck, K., Booch, G., & Cook, W. (1993). How to get a paper accepted at OOPSLA. ACM SIGPLAN NOTICES, 28, 429-429.

OOPSLA (Object-Oriented Programming, Systems, Languages & Applications) is an annual research conference. This paper tries to give authors some guidelines how to get your paper accepted at this conference.

The introduction mentions that previous papers on how to get accepted at OOPSLA were to general. In this paper, guidelines for four different research areas are presented: (1) theoretical papers, (2) experience papers, (3) method papers and (4) programming language papers.

Theoretical papers (1) should present a single, focused idea, whose claims are proven in the body of the paper. OOPSLA is not a theory conference. Therefore the presented theory must prove or disprove an accepted practice or theory and thus, be of interest to the community.

Experience papers (2) are currently a topic of broad interest to the research community at OOPSLA. A good experience paper focuses on a single topic by telling a story about a particular subject in software development. It must not have a lengthy theoretical foundation (long reference list) but rather more context for the reader in order to relate to the experiece. Additionally, experience papers should help the readers to apply insights to the readers context.

Method papers (3) should enable the practioneer to apply the knowledge gained to his specific problem. Being clear, focused and respecting the work of others is also important. Comparison papers are mostly not a good idea because they have been done often and provide little to no insights.

Programming language papers (4) are difficult to get accepted. Depending on the specific topic (and thus audience), different aspects must have more elaboration than others. There are papers about “Whole Cloth Languages”, “Language Extensions”, “Language Comparisons”, “Language Critiques” and “Language Semantics”. Each topic has different types of readers. Thus, the author has to tailor his paper for this kind of reviewer.

The last section is no guideline for a particular research area but a general approach on how to write a paper for OOPSLA, drawing conclusions from many errors that submitters made.