

How to evaluate the effectiveness of a given software development methodology? Naur argues that it is not easy to measure the effectiveness of a given software development methodology. He pointed out that it can be difficult to compare and to measure the improvements that are brought by the methodology. Indeed, Shaw (1946) pointed out, under the heading of "The Surprises of Attention and Neglect, that any attention is given to the neglected activity is likely to improve the situation.

Naur suggests an alternative to empirical evidence to evaluate the effectiveness of a software development methodology in relation to programmer's intuition in software development

1. The application of rules, techniques, methods, etc. in all software development activities are depends on programmer's intuition (i.e. when and how)
2. The need to address the shortcomings of programmers' intuitive insight and behaviour - omissions and flaws. Omissions can be avoided by repeating (by different person; c.f. reviews, pair-programming). Flaws can be avoided by reviews or walkthroughs

The following remarks are against formalism In evaluating the relationship between programmers' flaws and the form of expression (methodology) due to intuitive behaviour and insight, Naur (1984) argues that the informal quality of a form of expression is as equally important as its formal characteristics, as he wrote: "The claim is often made that certain forms, or formalizations, will guarantee the absence of flaws of arguments. What seems to lie behind such claim is the fact that by the use of certain kinds of formalizations it is possible to formalize the connections between statements. While this property is of great interest as a matter of principle, and also is the necessary basis for mechanical proof construction and verification, and occasionally is used in the reasoning carried out by people, it provides no guarantee for the absence of flaws in the arguments used in software development making use of formalizations." (p.77 Naur 1984) "Avoiding flaws in that modelling undoubtedly depends to some extent on the form or language used for the description. However, what is the most suitable form or language for any situation will depend not only on the degree of formalization of the description, but equally on the character of the aspect of the world to be modelled and the background experience of the programmer who has to establish the description." (p.77 Naur 1984)

"... software development in all its phases, and irrespective of the techniques employed in its pursuit, must and will always depend on intuition ... a view of software development that makes the application of rule-based methods and notations the basic issue is misguided." (p.78 Naur 1984)

Follow-up (find the reference for these facts):

1. Four (4) stages of skills acquisition in learning / organization behaviour / management
2. Experts solve problems slower than novices, because they have a bigger set of knowledge. Consequently, they may take longer to diagnose the problem before they come up with a solution.

Interesting References:

1. Quine, W.v.O.: Word and object, MIT Press, Cambridge, Massachusetts, 1960.
2. Popper, K.R.: The logic of scientific discovery. Hutchinson, London, 1959.
3. Ledgard, H., Whiteside, J.A., Singer, A., Seymour, W.: The natural language of interactive systems. Communications of ACM, 23(10), pp. 556-563
4. Naur, P.: Programming as theory building. Microprocessing and Microprogramming, 1985, pp.253-261
5. Shaw, B.: The doctor's dilemma. Penguin Books, Harmondsworth, Middlesex, England, 1946

~~~~~

---