

Past, Present and Future of Software Evolution

From Software-Now to Software-over-Centuries

Jean-Marie Favre

Software Language Archeologist
& Software Anthropologist
LIG, ACONIT
University of Grenoble, France

Software Evolution evolves as the very notion of Software evolves itself. Computer Scientists traditionally considered Software as a static and mathematical object. This initial and narrow view should be dramatically revisited if we want to fully understand what Software really is in practice. Considering a broader context, namely Informatics, helps to show that Software Evolution is actually a natural phenomenon. In fact, the traditional view on software missed two very important facts. First, software products are human-intensive artefacts involving many stakeholders, not only programmers. Secondly, time plays many different roles depending on the timescale considered. As the time passes, and after more than half century, it becomes clear that software products are not instant-based artefacts; they should on the contrary be considered as business, scientific and cultural assets. As a result, Software Evolution, but also Software History and Software Preservation are topics of increasing importance. All these topics result from various combinations of Software, Time and Stakeholders. We shows that various timescales should be considered, including milliseconds (processor's time), seconds (end-user's interaction time), minutes and hours (end-user's session time), days (developer's time), weeks and months (project manager's time), years and decades (current software product lifetime), and may be centuries or millenniums; who knows? Obviously, we can't predict the future, but as we are entering the so-called "Information Age", the importance of Software is likely to be key to future civilizations. Studying the past is a good way to forecast what might happen in the future. We show how historical studies could reveal useful to analyse the past and present of Informatics, as well as to predict and anticipate some forthcoming issues such as the evolution of Software Languages. We claim that Software Language Engineering and Software Linguistics should be studied in the context of Software Evolution, and the other way around.