* In the paper “On the Criteria To Be Used in Decomposing Systems into Modules”, the discussion about the changeability of modularization and hierarchical structure are the two key points that I think are still relevant today. Unlike other abandoned system structures such as monolithic, modularization is commonly used in today’s software design in order to allow system change or scale in an efficient way. As for hierarchical structure, I think it is also a very common thing in today’s software system such as architecture and data field: it is frequently mentioned during the study of Java, and hierarchical data structure is widely used in data design. In the paper “Information distribution aspects of design methodology”, the structure defined and documentation systems are the key points still relevant today. The study of structures in software has never stopped, the structures are changing and evolving as the time goes by, so is the definition of the word “structure”. Now the researchers and designers are still working on the “structure” that best fits their current needs. For documentation, nowadays it is a mandatory thing for a software that helps both clients and developers to understand and learn the details of the system and how to use it. I was surprised this idea was already raised in 1972, however it is still relevant today. In the paper “Software Aging”, I think the theme is still relevant today. Once a software is deployed, although it can be scaled or modified to meet the changing needs, its aging and retirement is an inevitable ending: just a matter of time. It is because the requirements, technologies and people are changing all the time, even the latest things right now will be outdated in the future, therefore this paper is still relevant today. The discussion about the software aging in the paper such as issues: performance, reliability, cost, thoughts about reviews and documentations, are very informative and enlightening for the latest software design and planning. They not only discussed the necessity of the software aging in different angles, but also provided suggestions to respond to those consequences and problems, to achieve a proper end and plan for the software system. Therefore this paper is still meaningful and relevant to today’s software field.
* I like the paper “On the Criteria To Be Used in Decomposing Systems into Modules” best, because modularization is a standard for all software projects. In software design, “high cohesion low coupling” is a well-known rule concluded from the many year’s modularization experience and research. I think almost all software systems are following this rule, which gives the systems the ability to change and scale in an easy, proper way. Therefore I like this topic best, as it is still relevant to software design today.