However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory..  
Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
Normally the first step in debugging is to attempt to reproduce the problem.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
Scripting and breakpointing is also part of this process.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.