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.  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
Relatedly, software engineering combines engineering techniques and principles with software development.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
There exist a lot of different approaches for each of those tasks.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
There are many approaches to the Software development process.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Many applications use a mix of several languages in their construction and use.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
Programming languages are essential for software development.  
Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a particular programming language, commonly referred to as coding).  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
There exist a lot of different approaches for each of those tasks.