One approach popular for requirements analysis is Use Case analysis..  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
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.  
Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash.  
However, readability is more than just programming style.  
 Code-breaking algorithms have also existed for centuries.  
Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
There are many approaches to the Software development process.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
 Allen Downey, in his book How To Think Like A Computer Scientist, writes:  
 Many computer languages provide a mechanism to call functions provided by shared libraries.  
Programming languages are essential for software development.