It is usually easier to code in "high-level" languages than in "low-level" ones.  
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.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
This is interpreted into machine code.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
There are many approaches to the Software development process.  
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.  
One approach popular for requirements analysis is Use Case analysis.  
This is interpreted into machine code.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.  
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 Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Integrated development environments (IDEs) aim to integrate all such help.