Relatedly, software engineering combines engineering techniques and principles with software development.  
There exist a lot of different approaches for each of those tasks.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
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.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
The following properties are among the most important:  
  
 In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
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.  
Programming languages are essential for software development.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
To produce machine code, the source code must either be compiled or transpiled.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
 New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation).  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).