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..  
 Following a consistent programming style often helps readability.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
  
One approach popular for requirements analysis is Use Case analysis.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 Programs were mostly entered using punched cards or paper tape.  
There exist a lot of different approaches for each of those tasks.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input.  
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.  
However, readability is more than just programming style.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.