Scripting and breakpointing is also part of this process.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Relatedly, software engineering combines engineering techniques and principles with software development.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Techniques like Code refactoring can enhance readability.  
This is interpreted into machine code.  
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.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
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.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Scripting and breakpointing is also part of this process.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.