Programmable devices have existed for centuries..  
One approach popular for requirements analysis is Use Case analysis.  
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.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
There are many approaches to the Software development process.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
 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.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 It is very difficult to determine what are the most popular modern programming languages.  
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.  
 Code-breaking algorithms have also existed for centuries.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.