

The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Ideally, the programming language best suited for the task at hand will be selected. Scripting and breakpointing is also part of this process. Different programming languages support different styles of programming (called programming paradigms). Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. There exist a lot of different approaches for each of those tasks. 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. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. Use of a static code analysis tool can help detect some possible problems. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Also, specific user environment and usage history can make it difficult to reproduce the problem. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.