

Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Also, specific user environment and usage history can make it difficult to reproduce the problem. Whatever the approach to development may be, the final program must satisfy some fundamental properties. There are many approaches to the Software development process. 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. 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. 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. 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. Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly. Techniques like Code refactoring can enhance readability. Techniques like Code refactoring can enhance readability. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. Following a consistent programming style often helps readability. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Normally the first step in debugging is to attempt to reproduce the problem. 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. Programming languages are essential for software development. The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine. Many applications use a mix of several languages in their construction and use. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. One approach popular for requirements analysis is Use Case analysis.