

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. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Techniques like Code refactoring can enhance readability. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Techniques like Code refactoring can enhance readability. Ideally, the programming language best suited for the task at hand will be selected. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Programs were mostly entered using punched cards or paper tape. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Also, specific user environment and usage history can make it difficult to reproduce the problem. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Code-breaking algorithms have also existed for centuries. Following a consistent programming style often helps readability. 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. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. 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. There are many approaches to the Software development process. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.