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. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Programs were mostly entered using punched cards or paper tape. 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. One approach popular for requirements analysis is Use Case analysis. 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. Computer programmers are those who write computer software. Different programming languages support different styles of programming (called programming paradigms). Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Scripting and breakpointing is also part of this process. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Use of a static code analysis tool can help detect some possible problems. 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. Ideally, the programming language best suited for the task at hand will be selected. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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. Unreadable code often leads to bugs, inefficiencies, and duplicated code. It is usually easier to code in "high-level" languages than in "low-level" ones. Normally the first step in debugging is to attempt to reproduce the problem. One approach popular for requirements analysis is Use Case analysis.