4gt technique

2. 22 Fourth Generation Techniques The term “ Fourth Generation Techniques” (4GT) encompasses a broad array of software tools that have one thing in common. Each enables the software engineer to specify some characteristic of software at a high level, the tool then automatically generates source code based on the developer's specification. The 4GT paradigm for software engineering focuses on the ability to specify software using specialized language forms or a graphic notation that describes the problem to be solved in terms that the customer can understand.

Currently, a software developmentenvironmentthat supports the 4GT paradigm includes some or all of the following tools: nonprocedural languages for database query, report generation, data manipulation, screen interaction and definition & code generation; high-level graphics capability; and spreadsheet capability. Like other paradigms, 4GT begins with a requirements gathering step, the customer would describe requirements and these would be directly translated into an operational prototype.

But this is unworkable, the customer may be ensure of what is required, may be ambiguous in specifying facts that are known, and may be unable or unwilling to specify information in a manner that a 4GT tool can consume. For small applications, it may possible to move directly from the requirements gathering step to implementation using a nonprocedural fourth generation language (4GL), for a larger efforts, it is necessary to develop a design strategy for the system, even if a 4GL is to be used.

The use of 4GT without design (for large projects) will cause the same difficulties (poor quality, poor maintainability and poor customer acceptance). Implementation using 4GL enables the software developer to represent desired output in a manner that results in automatic generation of code to generate the output. Obviously, a data structure with relevant information must exist and be readily accessible by the 4GL.

To transform a 4GT implementation into a product, the developer must conduct thorough testing, develop meaningful documentation, and perform all other solution integration activities that are required in other software engineering paradigms, the 4GT developed software must be built in a manner that enables maintenance to be performed expeditiously. The developers decided to use 4GT because it is the methodology that fits the system. The 4GT is a systematic approach to problem solving and is composed of several phases: