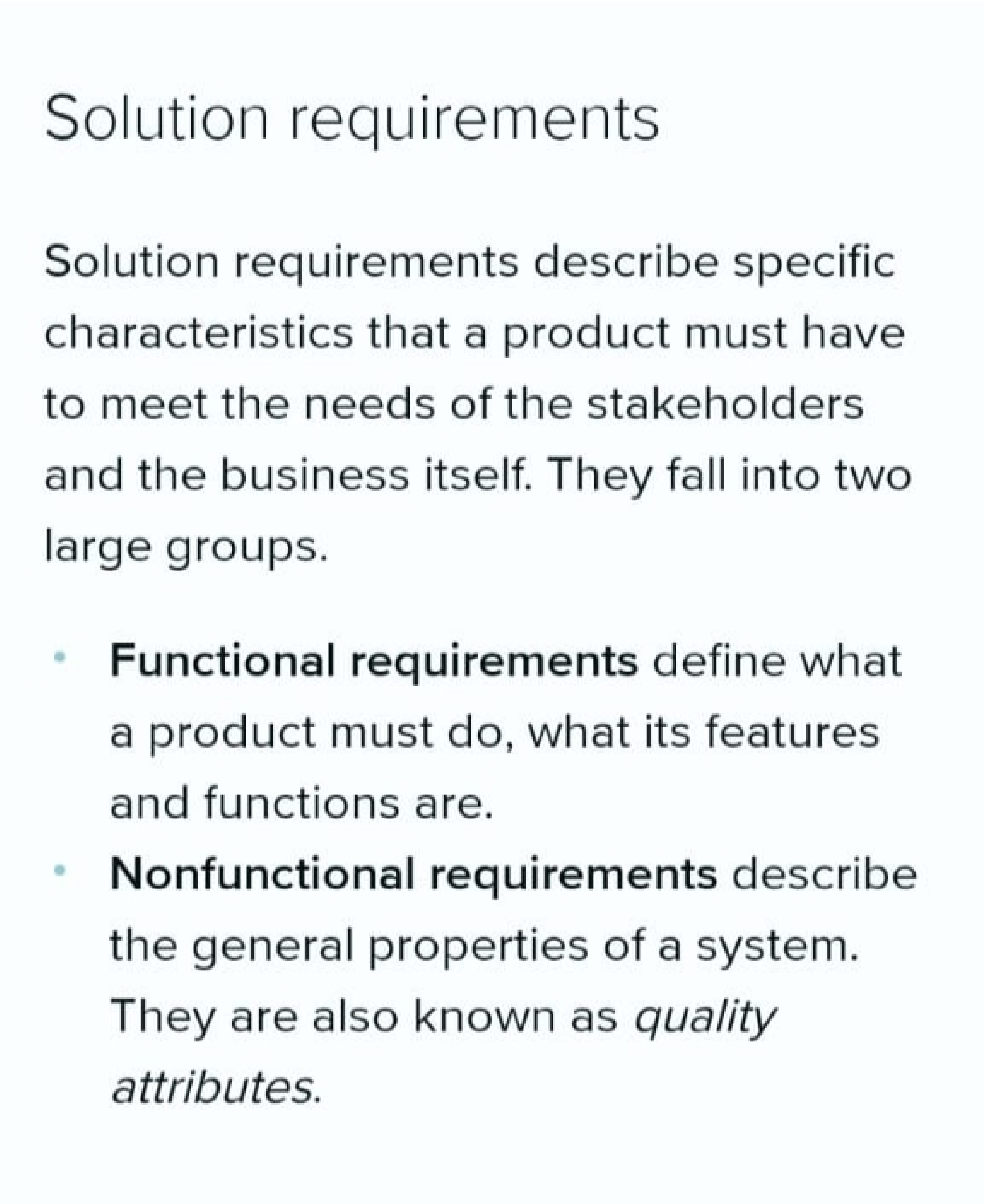
Project Development Phase:

Functional Features:



Solution requirements

Solution requirements describe specific characteristics that a product must have to meet the needs of the stakeholders and the business itself. They fall into two large groups.

- **Functional requirements** define what a product must do, what its features and functions are.
- **Nonfunctional requirements** describe the general properties of a system. They are also known as *quality attributes*.

Code layout, Readability, Reusability:

Some code is written in such a way that it is hard, or even impossible, to write a good unit test for it. Review some anti-patterns, code smells, and bad practices that we should avoid when writing testable code.

In software development, code reuse is the technique of reusing existing code or components to create new software applications or modules. Instead of beginning from scratch each time, developers can save time, effort, and resources by reusing existing code. Developers or testers can ensure code reliability by following Modularize code, defining clear interfaces, minimizing dependencies, writing flexible code.

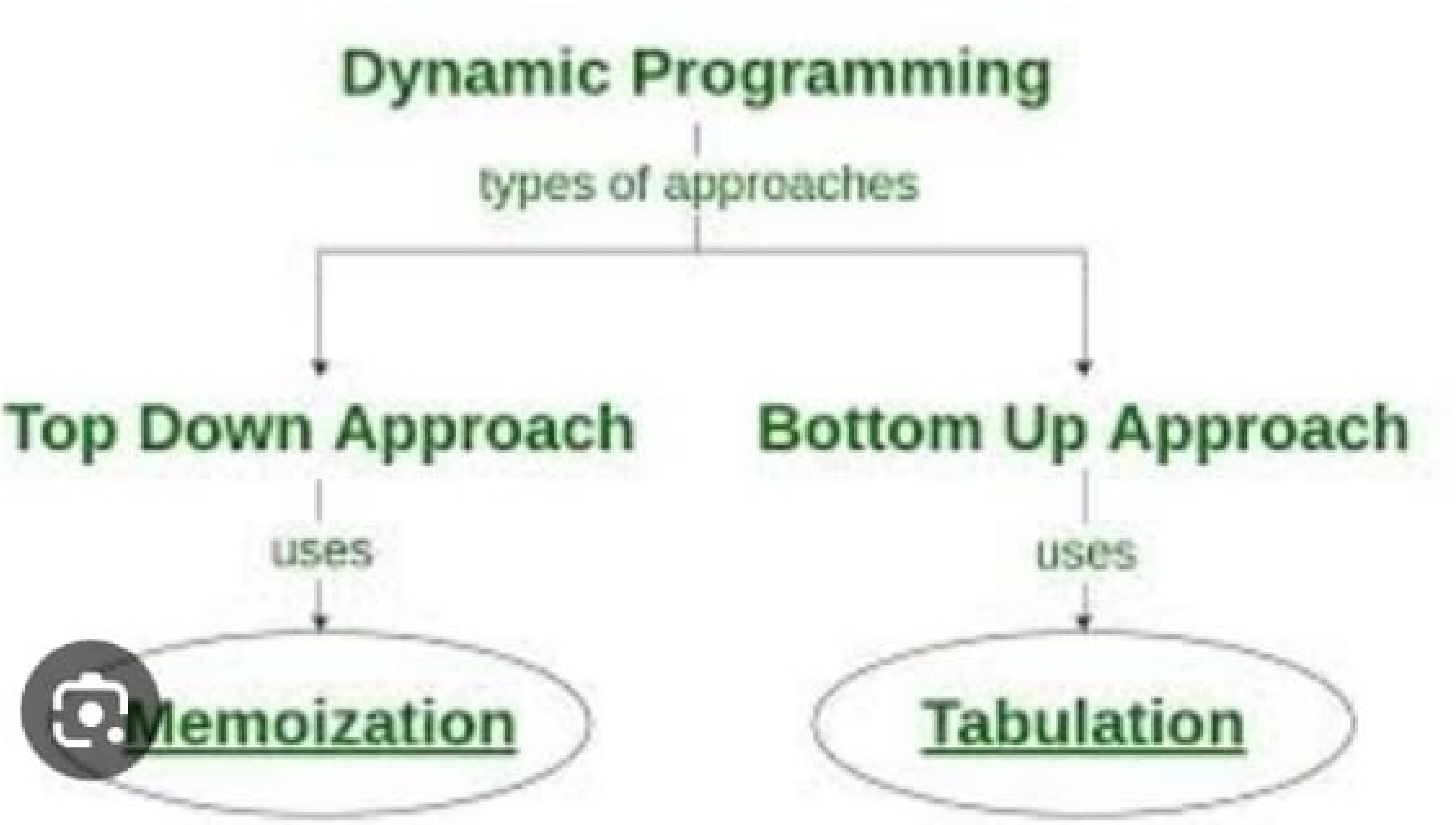
Dynamic Programming:

Dynamic programming is a problem-solving technique that tackles complex problems by dividing them into smaller subproblems that overlap. It breaks down the problem into manageable parts and solves them individually to find an optimal solution.

It aims to find the optimal solution by efficiently solving these subproblems and combining their solutions.

Dynamic programming stores the results of subproblems in a table or cache, allowing for efficient retrieval and reuse of previously computed solutions.

At its core, dynamic programming relies on two fundamental principles: optimal substructure and overlapping subproblems.



Debugging and Traceability:

Debugging is the process of finding and fixing errors or bugs in the source code of any software. When software does not work as expected, computer programmers study the code to determine why any errors occurred. They use debugging tools to run the software in a controlled environment, check the code step by step, and analyze and fix the issue.

The term debugging can be traced back to Admiral Grace Hopper, who worked at Harvard University in the 1940s. When one of her colleagues found a moth impeding the operation of one of the university's computers, she told them they were debugging the system. Computer programmers were first recorded as using the terms bugs and debugging by the 1950s, and by the early 1960s, the term debugging was commonly accepted in the programming community.

Bugs and errors happen in computer programming because it is an abstract and conceptual activity. Computers manipulate data in the form of electronic signals. Programming languages abstract this information so humans can interact with computers more efficiently. Any type of software has several layers of abstraction, with different components communicating for an application to work correctly. When errors occur, finding and resolving the issue can be challenging. Debugging tools and strategies help to fix problems faster and improve developer productivity. As a result, both software quality and the end-user experience improve.

Traceability is the ability to trace something. Across industries, including healthcare, manufacturing, supply chain, and software development, traceability ensures that final deliverables don’ t stray too far away from original requirements.

Exception Handling:

Exception handling is a mechanism that separates code that detects and handles exceptional circumstances from the rest of your program. Note that an exceptional circumstance is not necessarily an error.

When a function detects an exceptional situation, you represent this with an object. This object is called an exception object. In order to deal with the exceptional situation you throw the exception. This passes control, as well as the exception, to a designated block of code in a direct or indirect caller of the function that threw the exception. This block of code is called a handler. In a handler, you specify the types of exceptions that it may process. The C++ run time, together with the generated code, will pass control to the first appropriate handler that is able to process the exception thrown. When this happens, an exception is caught. A handler may rethrow an exception so it can be caught by another handler.