**Regression testing** plays a vital role in software engineering to maintain the stability and dependability of a software product. It focuses on retesting previously verified features to ensure that recent code modifications have not introduced new issues or negatively impacted existing functionalities.

By detecting and resolving any regressions or unexpected issues, **regression testing** ensures that the overall quality of the software remains intact. This practice is crucial for development teams to provide reliable and consistently high-quality software products to their users.

### What is Regression Testing?

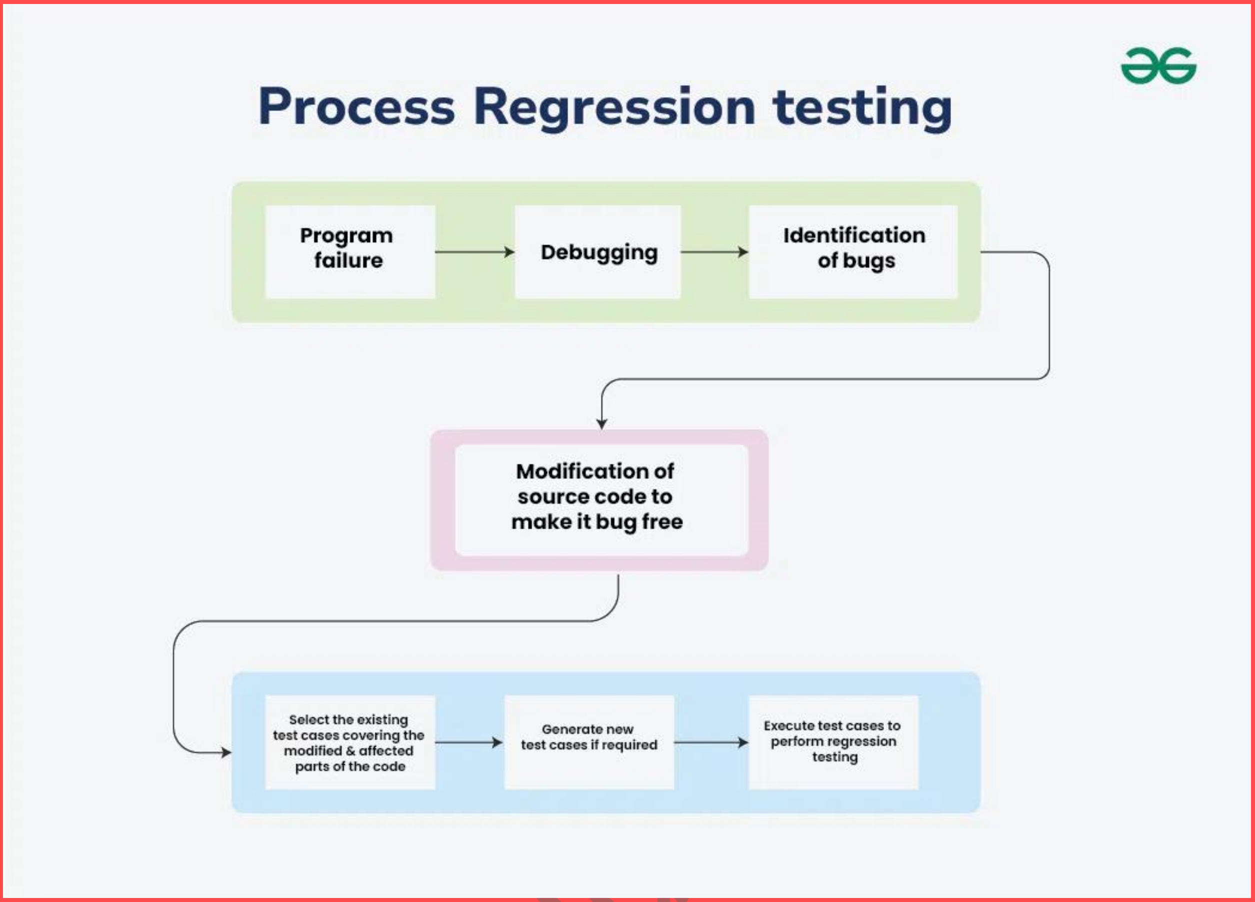
Regression testing is like a software "health check" that happens after any changes are made to the code. The main goal is to run tests to ensure everything continues to work as expected, even after updates or changes. This helps maintain the software's reliability and ensures that new changes haven’t unintentionally broken existing features, keeping the software consistent and trustworthy throughout its development.

### When to Do Regression Testing?

- **When new features are added:** If new functionality is introduced, regression testing ensures it integrates well without causing issues to existing code.
- **When a bug is fixed:** After identifying and correcting a defect, regression testing verifies that the fix did not create new issues.
- **When code is optimized:** Any modifications made to enhance code performance are checked to confirm they haven’t negatively impacted other parts of the software.

### Process of Regression Testing

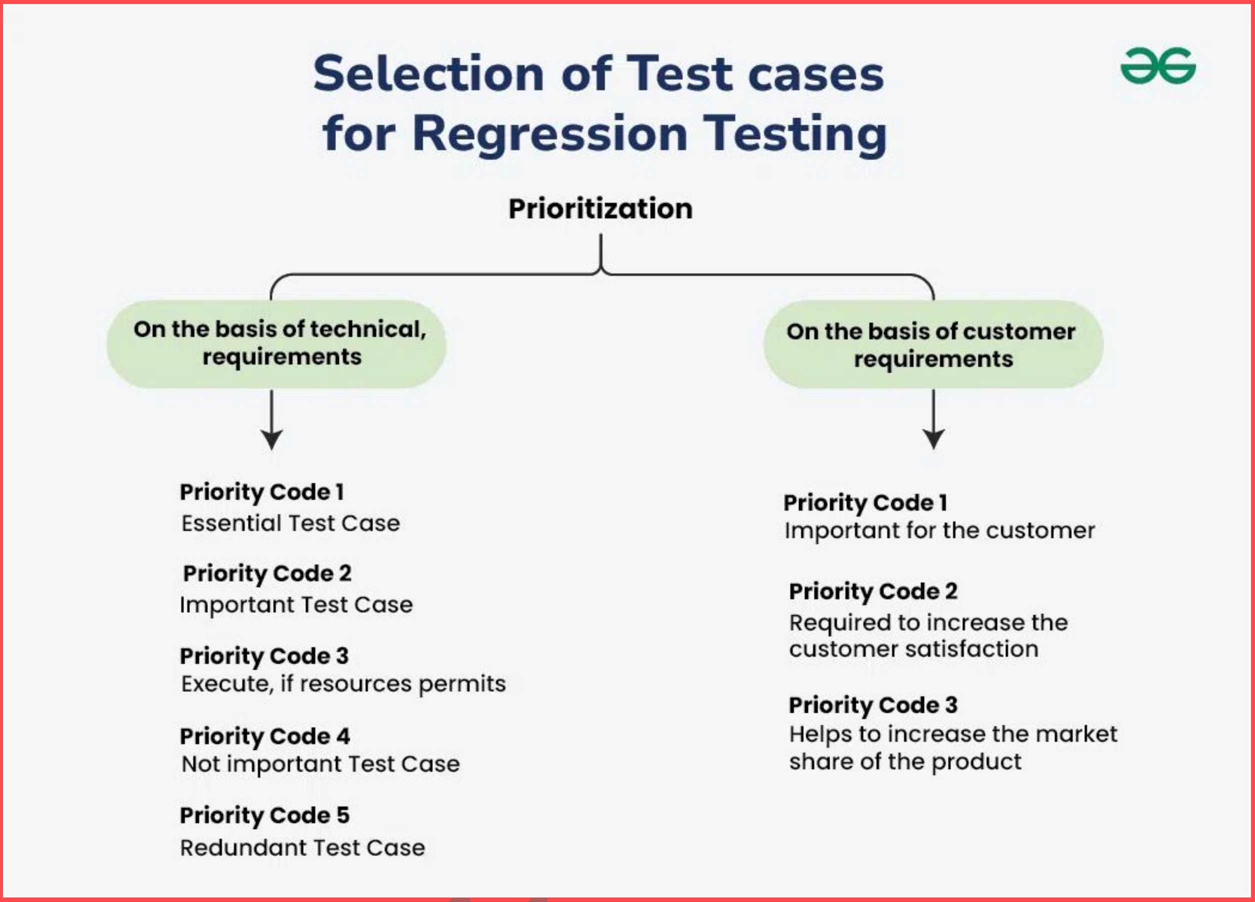
1. **Code Changes:** Changes may occur due to adding features, optimizations, or bug fixes. These changes can initially cause previously successful test cases to fail.
2. **Debugging:** The source code is reviewed and debugged to identify any bugs or issues.
3. **Modifications:** Necessary changes are made to the code based on the identified issues.
4. **Test Case Selection:** Relevant test cases from the existing test suite are selected to cover all modified or affected parts of the code. If needed, new test cases may be created.
5. **Regression Testing Execution:** The selected test cases are run to confirm that the software still functions correctly and that the recent changes haven’t introduced new issues.


Process Regression testing
Program failure
Debugging
Identification of bugs
Modification of source code to make it bug free
Select the existing test cases covering the modified & affected parts of the code
Generate new test cases if required
Execute test cases to perform regression testing

**Techniques for Selecting Test Cases for Regression Testing**

1. **Select All Test Cases:**
   - This method involves selecting all available test cases from the existing test suite. While it is the safest and simplest approach, it is often inefficient and time-consuming due to the large number of tests.
2. **Random Test Case Selection:**
   - In this method, test cases are picked at random from the test suite. This approach can be useful only if every test case has similar effectiveness in finding defects. However, since this is rarely the case, it is not commonly used.
3. **Select Modification-Related Test Cases:**
   - This technique focuses on selecting test cases that specifically cover modified code and the sections affected by these changes. By targeting these areas, it ensures that changes do not introduce new issues.
4. **Select High-Priority Test Cases:**
   - Each test case is assigned a priority based on factors like its ability to detect critical bugs, customer needs, and more. During regression testing, test cases with the highest

priorities are chosen first. For instance, a test case with a priority code of 1 is more important than one with a priority code of 2, making it more likely to be included in the regression tests.



## Selection of Test cases for Regression Testing

Prioritization

On the basis of technical, requirements

**Priority Code 1**
Essential Test Case

**Priority Code 2**
Important Test Case

**Priority Code 3**
Execute, if resources permits

**Priority Code 4**
Not important Test Case

**Priority Code 5**
Redundant Test Case

On the basis of customer requirements

**Priority Code 1**
Important for the customer

**Priority Code 2**
Required to increase the customer satisfaction

**Priority Code 3**
Helps to increase the market share of the product

## Advantages of Regression Testing

1.

2. **.Automated Unit Testing:**
   - Automated regression tests can help efficiently validate the functionality of each unit, reducing manual effort and errors.

3. **Comprehensive Test Coverage:**
   - Regression testing ensures that all parts of the system, including previously tested features, are thoroughly covered, ensuring no areas are overlooked.

4. **System Integration:**

   - It verifies that the modified code works seamlessly with the integrated components of the system, ensuring overall system stability.

5. **Faster Test Execution:**

   - With automation, regression tests can be executed faster, ensuring quicker feedback and reduced testing time.

6. **Improved Developer Productivity:**

   - By identifying defects early and reducing manual effort, developers can focus on implementing new features or optimizing existing code.

7. **Parallel Testing:**

   - Automation allows for parallel execution of tests, which speeds up the testing process and improves efficiency.

8. **Reduced Costs:**

   - Early detection of defects minimizes the cost of fixing issues later in the development process, making regression testing a cost-effective approach.

9. **Improved Product Quality:**

   - Regression testing helps maintain product quality by ensuring that new code does not negatively impact the system's existing functionality.

10. **Reusability:**

    - Test cases developed for regression testing can be reused in future testing cycles, saving time and effort.

11. **Scalability:**

- Regression testing can easily be scaled to accommodate growing and evolving software systems, ensuring that new changes do not affect the system's integrity.

11. **Time Efficiency:**

- Automated regression testing helps save time by quickly identifying issues and providing immediate feedback on changes.