SQA steps to follow:

1. Testing Techniques:

- **Functional Testing:** Verifies that the software functions as expected, covering various scenarios and use cases.
- **Non-Functional Testing:** Evaluates aspects like performance, security, usability, and reliability.
- **Unit Testing:** Tests individual components or modules of the software in isolation.
- **Integration Testing:** Checks how different modules or components interact with each other.
- **System Testing:** Tests the entire software system as a whole.
- **User Acceptance Testing (UAT):** Involves end-users to test the software in a real-world environment.
- **Stress Testing:** Determines the software's ability to handle extreme loads or conditions.
- **Security Testing:** Identifies vulnerabilities and weaknesses in the software's security mechanisms.
- **Performance Testing:** Measures the software's performance under different conditions.
- **Regression Testing:** Ensures that new code changes haven't introduced any new defects.
- **Automation Testing:** Automates repetitive testing tasks to improve efficiency and reduce errors.

2. Review and Auditing:

- **Code Reviews:** Peer review of the code to identify potential issues and ensure adherence to coding standards.
- **Design Reviews:** Examining the software design to identify potential problems and ensure it meets requirements.
- **Requirements Reviews:** Ensuring that the requirements are clear, complete, and consistent.

- **Auditing:** Regularly checking that SQA processes and procedures are being followed.
  3. Process and Methodologies:

- **Waterfall Methodology:**

  A sequential approach to software development, where each phase is completed before the next one begins.

- **Agile Methodology:**

  An iterative and incremental approach to software development, focusing on flexibility and collaboration.

- **Software Engineering Practices:**

  Applying established software engineering principles and techniques to ensure quality.

- **Standardization:**

  Establishing and following industry standards and best practices.

- **Documentation:**
  Maintaining comprehensive documentation of the software, processes, and testing results.
  4. Other Important Aspects:

  - **Risk Management:** Identifying and mitigating potential risks that could impact software quality.
  - **Continuous Monitoring:** Continuously monitoring the software and its performance to identify and address potential issues.
  - **Collaboration:** Fostering close collaboration between the QA team and the development team.
  - **Early and Continuous Testing:** Testing early in the development lifecycle to catch defects early and reduce costs.
  - **Defect Tracking:** Using a robust defect tracking system to manage and track defects.
  - **Cybersecurity Testing:** Identifying and addressing potential security vulnerabilities.

- **Maintaining Records and Reports:** Keeping records of test results, audit results, review reports, and change requests.