**1. Requirement Completeness**

**All Functional Requirements**: Have all functional aspects of the system been captured (what the system should do)?

**All Non-Functional Requirements**: Are non-functional aspects such as performance, security, usability, and scalability addressed?

**Business Requirements**: Do the requirements reflect the business objectives and goals of the project?

**System Constraints**: Are system constraints (e.g., hardware, software, network) defined clearly?

**Edge Cases and Exception Handling**: Are edge cases and exception scenarios covered?

**User Roles and Permissions**: Are different user roles, permissions, and access levels clearly defined?

**Interdependencies**: Are there any dependencies between requirements identified and addressed?

**2. Requirement Clarity**

**Clear and Unambiguous**: Are the requirements written clearly without any ambiguity or confusion?

**Precise Language**: Is the language specific and precise, leaving no room for interpretation?

**No Jargon or Technical Terms**: Does the document avoid unnecessary technical jargon, or are these terms well-defined?

**Consistency in Terminology**: Are consistent terms and definitions used throughout the document?

**Defined Acronyms/Abbreviations**: Are all acronyms and abbreviations defined when first introduced?

**3. Requirement Traceability**

**Unique Requirement ID**: Each requirement has a unique identifier to track and reference it.

**Requirement Source**: Is the source of the requirement specified (e.g., client, regulatory standard)?

**Linked to Business Objectives**: Are the requirements linked back to the business objectives or project goals?

**Traceability Matrix**: Is a traceability matrix in place that links requirements to test cases, design elements, and features?

**Cross-Referencing**: Are there references to other related requirements or documents where necessary?

**4. Testability of Requirements**

**Clear Success Criteria**: Are the success criteria or conditions for meeting the requirement clearly defined?

**Measurable**: Are the requirements measurable (e.g., response times, throughput, transaction volume)?

**Testable**: Can the requirement be tested through validation or verification?

**Acceptance Criteria**: Are acceptance criteria for each requirement specified and verifiable?

**5. Requirement Consistency**

**No Conflicting Requirements**: Are there no contradictions or conflicts between different requirements?

**Consistency Across Documents**: Do the requirements align with other related documents, such as design specifications, user stories, or functional specifications?

**Prioritization**: Are the requirements prioritized (e.g., must-have, should-have, nice-to-have)?

**Version Control**: Is the version of the requirement document tracked for future reference and updates?

**6. Requirement Feasibility**

**Realistic and Achievable**: Are the requirements realistic given the time, budget, and technology available?

**Scalable**: Can the system meet the requirements while scaling up (in case of future growth)?

**Resource Availability**: Are the necessary resources (e.g., hardware, software, human resources) available to meet the requirements?

**Legal and Compliance Feasibility**: Are all legal, regulatory, and compliance requirements addressed and feasible?

**7. Requirement Change Management**

**Version History**: Does the document include version history to track changes made to the requirements?

**Change Control Process**: Is there a defined process for managing changes to requirements, including impact analysis and approval steps?

**Impact of Changes**: Are potential changes reviewed for impact on cost, scope, schedule, and quality?

**8. Stakeholder Involvement**

**Stakeholder Approval**: Have all relevant stakeholders (e.g., business owners, product managers, developers) reviewed and approved the requirements?

**Clear Ownership**: Is ownership of each requirement clearly assigned (e.g., product owner, business analyst)?

**Stakeholder Feedback**: Has feedback from stakeholders been incorporated into the requirements?

**9. Requirement Prioritization**

**Business Value**: Are the requirements prioritized based on their business value and impact?

**Risk Assessment**: Have the requirements been assessed for potential risks (e.g., complexity, dependencies)?

**Critical vs. Optional**: Are critical requirements separated from optional or lower-priority features?

**10. Compliance and Standards**

**Regulatory Compliance**: Do the requirements comply with relevant regulations (e.g., GDPR, HIPAA, SOX)?

**Industry Standards**: Are industry standards and best practices (e.g., ISO, IEEE) followed in the requirement definitions?

**Security and Privacy**: Are security and privacy considerations clearly defined (e.g., encryption, user data protection)?

**11. Requirement Communication**

**Clear Communication**: Are requirements written in a way that is easily understood by all stakeholders (e.g., developers, testers, clients)?

**Visual Aids**: Are diagrams, mockups, flowcharts, or other visual aids used to clarify complex requirements?

**Traceability to Documentation**: Are all requirements linked to other relevant documents such as user stories, use cases, and design documents?

**12. Documentation Quality**

**Format and Structure**: Is the requirement document well-organized and formatted with headings, bullet points, and consistent sectioning?

**Documentation Standards**: Does the document adhere to internal standards and templates (e.g., naming conventions, writing style)?

**Usability of Document**: Is the document easy to navigate and search for specific requirements?

**13. User Stories and Use Cases (if applicable)**

**Clear User Stories**: Are user stories clearly defined with "As a [role], I want [feature] so that [benefit]" structure?

**Use Case Coverage**: Do use cases capture all interactions and scenarios between users and the system?

**Acceptance Criteria for User Stories**: Are clear acceptance criteria specified for each user story or use case?

**14. Risks and Assumptions**

**Identified Risks**: Are potential risks associated with requirements documented (e.g., technical challenges, unknown dependencies)?

**Assumptions**: Are assumptions clearly stated and justified, especially regarding system behavior, user input, or external systems?

**Additional Considerations**

**Dependency Management**: Are there any dependencies between requirements that need to be tracked (e.g., upstream/downstream)?

**Localization/Internationalization**: Are requirements for localization and internationalization (if applicable) addressed?

**Performance and Scalability**: Are performance and scalability requirements well defined (e.g., transaction speed, system load)?

**Post-Implementation Considerations**: Are post-implementation requirements such as maintenance, monitoring, and support addressed?