Code Review Criteria

**Before Review Session**

1. **First, allow a certain amount of time for entire review process**
   1. Work with Project Manager and upper-level stakeholders to determine the timeline for Code Review
2. **Prepare before review session (this should be done by each individual)**
   1. Participants: Producer(s), Review leader, Reviewers, and Documenter
   2. Code is easy to understand (Readability)
   3. Code has clean format or architecture
   4. Code should be easy to maintain
   5. Code should be easy to debug
   6. Code is efficient and performs well
   7. Code meets all functional and non-functional requirements
3. **Identify poor code formatting and ensure code is developed in uniform manner (this should be done by each individual)**
   1. Use alignments (left margin) to ensure code block starting and ending points are easily identifiable
   2. Ensure proper naming conventions are followed (ex. Pascal, CamelCase)
   3. Use constants/variables. Do not hard code
   4. Use comments effectively.
4. **Identify individual errors/issues in code**
   1. Document all errors found before meeting and/or attempt to fix errors before meeting
   2. Label these errors as either “Minor” or “Major”

**During Review**

1. **Verify software meets all requirements**
   1. Ensure software meets all non-functional requirements such as maintainability, readability, testability, debuggability, reusability, reliability, security, performance, scalability, etc.
   2. Ensure software meets all stipulations outlined in the requirements
2. **Ensure software has been written according to predefined standards**
   1. Ensure individual code is developed in a uniform manner
   2. Ensure any client’s coding standards are met
3. **Identify System errors/issues in code**
   1. Double check everything in previous bullet with Producer(s), Review leader, reviewers, and documenter
   2. Document all errors found during meeting, do not try to fix during meeting
   3. Label these errors as either “Minor” or “Major”

**After Meeting Review**

1. **Spend time fixing errors**
   1. Prioritize “Major” errors and spend most time on solving these
   2. Next, complete as many (or all) “Minor” fixes as time (and money) allows