Many items were taken from here and here.

**Code**

- [ ] Are CI builds passing? If no, why?

- [ ] Is the code easily understood?
- [ ] Does the code work? Does it perform its intended function, the logic is correct, etc?
- [ ] Does the error handling work?
- [ ] Is memory usage acceptable, even with large inputs?

- [ ] Is code covered by functional or unit tests?
- [ ] Are error paths covered by functional or unit tests? All errors which are relatively easy to check must be checked: error conditions like "open() failed after stat() was successfull" or "array size greater then INT_MAX" may be ignored for being just as unlikely as uneasy to test, but otherwise having bugs in code which does error handling is way too common to be ignored.
- [ ] For new code, are unit tests written where needed?

- [ ] Are invalid parameter values handled where needed?
- [ ] Can any global/static variables be replaced?
- [ ] Are variables/functions named intuitively?
- [ ] Can any function attributes be used?

- [ ] Is there any redundant or duplicate code?
- [ ] Is the code modular enough?
- [ ] Can any of the code be replaced with library functions?
- [ ] Do loops have a set length and correct termination conditions?
- [ ] Can any logging or debugging code be removed?
- [ ] Are there any unneeded assert statements?

- [ ] Does the code conform to the style guide?
- [ ] Optimization that makes code harder to read should only be implemented if a profiler or other tool has indicated that the routine stands to gain from optimization. These kinds of optimizations should be well-documented and code that performs the same task simply should be preserved somewhere.

- [ ] Are return values being checked?
- [ ] Are there any use after frees?
- [ ] Are there any resource leaks? Memory leaks, unclosed sockets, etc.
- [ ] Are there any null pointer dereferences?
- [ ] Are any uninitialized variables used?
- [ ] Are there any cases of possible arithmetic overflow?

**Documentation**

- [ ] Are there any superfluous comments?
- [ ] Where needed, do comments exist and describe the intent of the code?

- [ ] Are any comments made outdated by the new code?
- [ ] Is any unusual behavior or edge-case handling described?
- [ ] Are complex algorithms explained and justified?
- [ ] Is code that depends on non-obvious behavior in external libraries documented with reference to external documentation?
- [ ] Is the use and function of API functions documented?
- [ ] Are data structures/typedefs explained?
- [ ] Is there any incomplete code, e.g., code marked `TODO`, `FIXME`, or `XXX`?