**Code Review Checklist**

**General**

* Does the code work? Does it perform its intended function, the logic is correct etc.
* Is all the code easily understood?
* Does it conform to your agreed coding conventions? These will usually cover location of braces, variable and function names, line length, indentations, formatting, and comments.
* Is there any redundant or duplicate code?
* Is the code as modular as possible?
* Can any global variables be replaced?
* Is there any commented out code?
* Do loops have a set length and correct termination conditions?
* Can any of the code be replaced with library functions?
* Can any logging or debugging code be removed?

**Security**

* Are all data inputs checked (for the correct type, length, format, and range) and encoded?
* Where third-party utilities are used, are returning errors being caught?
* Are output values checked and encoded?
* Are invalid parameter values handled?

**Documentation**

* integration testing
* Do comments exist and describe the intent of the code?
* Are all functions commented?
* Is any unusual behavior or edge-case handling described?
* Is the use and function of third-party libraries documented?
* Are data structures and units of measurement explained?
* Is there any incomplete code? If so, should it be removed or flagged with a suitable marker like ‘TODO’?

**Testing**

* Is the code testable? i.e. don’t add too many or hide dependencies, unable to initialize objects, test frameworks can use methods etc.
* Do tests exist and are they comprehensive? i.e. has at least your agreed on code coverage.
* Do unit tests actually test that the code is performing the intended functionality?
* Are arrays checked for ‘out-of-bound’ errors?
* Could any test code be replaced with the use of an existing API?