Peer Review Check List

1	Has a code merge from the core development branch been done before the pull request was raised?
2	Are there any conflicts in the feature branch?
3	Has the code been written using the latest version of the respective Software Development Kit (SDK)?
4	Is the code project using the latest version of the development framework? If yes or no, are there any referenced libraries that are now mismatched and also need to be updated?
5	Does the code include good comments? Things that explain the reason why logic has been implemented to assist future developers looking at the same code.
6	Building on point 5. Is the code structured in a way that is easy to understand?
7	Have functions and methods been created in a way that is generic?
8	Similar to point 7. Have functions and methods be written or re-written in a way that offers the ability to create further overloaded versions of the same?
9	Is there any ambiguity in the code?
10	If/when business logic needs to be included in a process, has it been parameterised to offer future flexibility?
11	Have the local technical standards been followed in the new code created?
12	Has performance been considered when processing the data?
13	Does the underlying storage/compute resource used have any limitations that need to be considered as a result of the newly created code?
14	Has the cost of the underlying storage/compute resources been considered?
15	Are there any wider industry best practices that should be considered in the code created?
16	Has the code been written with logging and auditing in mind?
17	Are exceptions handled and if so, is the output meaningful? This also alludes to a defensive mindset; can we predict where an error in the code will occur?
18	If/when an error occurs in the code and an exception is raised, does the data transformation leave the source or target dataset in a transaction safe state with data integrity preserved?
19	Is authentication handled correctly for the solution with as many layers of protection as is reasonably possible in place? If so, does the code created expose any weaknesses that short circuit the layers of security?
20	Does the code under review require an element of testing or re-testing that wasn't originally planned for?

Source: mrpaulandrew.tech