Code Reviews

From Google's "Best Practices for a Code Review" (https://google.github.io/eng-practices/review/)

Look for …. Design

Do the interactions of various pieces of code make sense? Does it integrate well with the rest of your system? Is now a good time to add this functionality?

Look for …. Functionality

Does the code do what the developer intended?

Look for … Complexity

Is the code more complex than it should be? "Too complex" usually means ***"can't be understood quickly by code readers."*** It can also mean ***"developers are likely to introduce bugs when they try to call or modify this code."***

Look for … Tests

Ask for unit tests as appropriate for the change.

Code Reviews

From Google’s Best Practices for a Code Review

Look for … Naming Conventions

Did the developer pick good names for everything? A good name is long enough to fully communicate what the item is or does, without being so long that it becomes hard to read.

Look for … Comments

Did the developer write clear comments in understandable English?.

Look for … Style

Google has Java style guides for major languages to illustrate common conventions. Do you see them?

Look for …. Consistency

Do the interactions of various pieces of code make sense? Does it integrate well with the rest of your system? Is now a good time to add this functionality?

Code Reviews

From Google's Best Practices for a Code Review

Look for …. Documentation

Does anything need to change in the JavaDoc or API?

Look for … Solid Code

Can you understand the code? Is it well organized?

Look for … Good Things, Generally

Are the opportunities for compliments? Can you highlight was is done well?