**Issue 1**: Case where there are a lot of calls or communication between two files, classes, or components written in a different programming language. (A method that performs a call in a loop to a method written in another language. Or, a method that passes an object to another language that is then returned back, etc.)

**Issue 2:** Case where there is a class like a blob class that contains a lot of foreign methods. These methods should differ in term of concerns ( A class that contains many foreign methods (like for JNI, for example, native methods declaration). These methods should vary in term of responsibility. For example, methods related to compression, other to the management of file, audio, etc.))  
  
**Issue 3:** Case where the tasks or methods existing in the class that is multi-language could have been written in a single language. (a class that contains foreign methods that could have been implemented in the same language. For example for JNI, we have a native method implemented in C while it can have been written in Java and there is no good reason to introduce JNI.)  
  
**Issue 4:** Case there are many classes that contain foreign code, while the foreign code could have been grouped in the same or few cases. In this case, the concerns are not well separated and the foreign code (like jni) is not well isolated. (classes that contain calls of code written in another programming language. In that case, it is better to put the foreign code in only one class of them and isolate the foreign code.)  
  
**Issue 5:** Case of not using the right programming language for the right task. For example a calculation task implemented in Java that it is better implemented in C, etc.  
  
**Issue 6:** Case of migration issues. example of a project that had problems to be migrated from one language to another. For example from Cobol to Java, or from VB, etc.