Design patterns are beneficial because they promote flexibility and reuse. However, they can cause unnecessary complexity when applied to simple problems. Design patterns are also inflexible and restrict adaptability, requiring a designer to follow a certain template which may not exactly or entirely suit their needs.

An instance where using a design pattern may not be beneficial is in the case of overengineering. For example: using a factory pattern which adds excessive classes and interfacesm, when only a few concrete classes are needed. A designer uses a pattern much larger than the scope of their needs, which could lead to complications and errors. When developers utilize functionality based on only anticipated needs, it results in code that's needlessly harder to maintain.

Sources: refactoring.guru , https://www.gofpattern.com/design-patterns/module7/limitations-of-design-patterns.php, https://www.geeksforgeeks.org/factory-method-for-designing-pattern/