Factory Method Pattern

The Factory Method Pattern is important in software design. It allows you to create objects without specifying the exact class of the object, which will be created. This method also decouples the client code from the specific implementations of the product, which will lead to reducing dependencies and making the system more maintainable, because Clients depend on the abstract Creator interface rather than concrete classes. The testing for that method is also easy because you can add mock objects for the real ones. This is useful for unit testing. You can freely add new objects, which means that Factory method allows you to extend the system by adding new concrete creators or products without changing the existing code.
This method plays a big role in organization, it helps maintain the structure of code and makes it more understandable and maintainable.
In case you don't Implement this method you could face a lot of issues.
The code would have tight coupling between the client and the concrete product classes. This would make it harder to introduce new products or change the product creation process without modifying the client code. You would also face the issue of code duplication, which means when you create objects in multiple places in your code, you will automatically introduce duplicated object creation logic. This would lead to potential errors and maintenance challenges if the process changes. You would always need to change the client code, if you want to introduce new products or variations.
The testing would be harder too, which means that you can not easily substitute objects with mock implementations for testing purposes.

To summarize, the Factory Method Pattern is critical for writing flexible, extendable, and maintainable code by separating the object creation process from client code and adhering to important object-oriented design principles. Without it, you may wind up with less versatile code that is more difficult to maintain and test.