Order management task

Background :

- we have solution that manage order status and print order receipt , we have 5 status : pending, accepted,processing, delivering, receiving, rejected.
- We have product class that manage product details and calculate the price based on the attributes.
- we have shipping class that manage the shipping method and calculate the tax based on the countries and the method type .
- we have user and address class that interact with our solution .

Your assessment :

- re-factoring this code (restructure this design , changing names , separate the responsibility, add new classes if you need that).
- we need to extend the statuses, shipping methods and countries so we need mechanism to do that in our solution.
- we need to print the receipt with another formatting like xml , json , and so on to interact with another system.
- we need to handle the exceptions .
- use design patterns and any data structure you want but avoid do over-engineering or making a lot of layers of abstraction.

Our expectation :

- you know phpunit if you don't know it, Google it.
- you know characterization test, if you don't know it search about it.
- you are using php latest version .
- when you finish this task upload it in Google drive and send the link for it and not upload it in github .
- write readme file to explain your effort.

Our Evaluation :

- the testOrderHaveReceipt test must be the same without any change.
- We make sure you did not change existing plain text output in any way.
- We evaluate the cleanliness of your code, in terms of the size and tidiness of classes and methods, the quality of the names of your variables, classes, methods, the accuracy and usefulness of the comments if they are found, etc…

- We evaluate your design approach, how you used OOP to your advantage, how do you use patterns and abstractions and whether they add value or are just there to show off your knowledge.
- We evaluate your JSON representation for proper understanding of JSON nature and how it represents data elements.

Bonus :

- use docker.
- write unit test to cover this code.
- add canceled and returned status.
- add new shipping method called (fetcher).
- add xml formatting to represent the receipt .