**Assignment questions:**

1. Give 3 to 4 examples of acceptance testing, For a software or a system of your suggestion?
2. If you are building a software for a nuclear power plant, which testing model you would choose? and why?
3. Explain the customer testing role in different testing models?
4. What the term continuous beta means? and what is the reason behind using such an approach?
5. What are the most important parts of the system, that a tester should examine thoroughly?

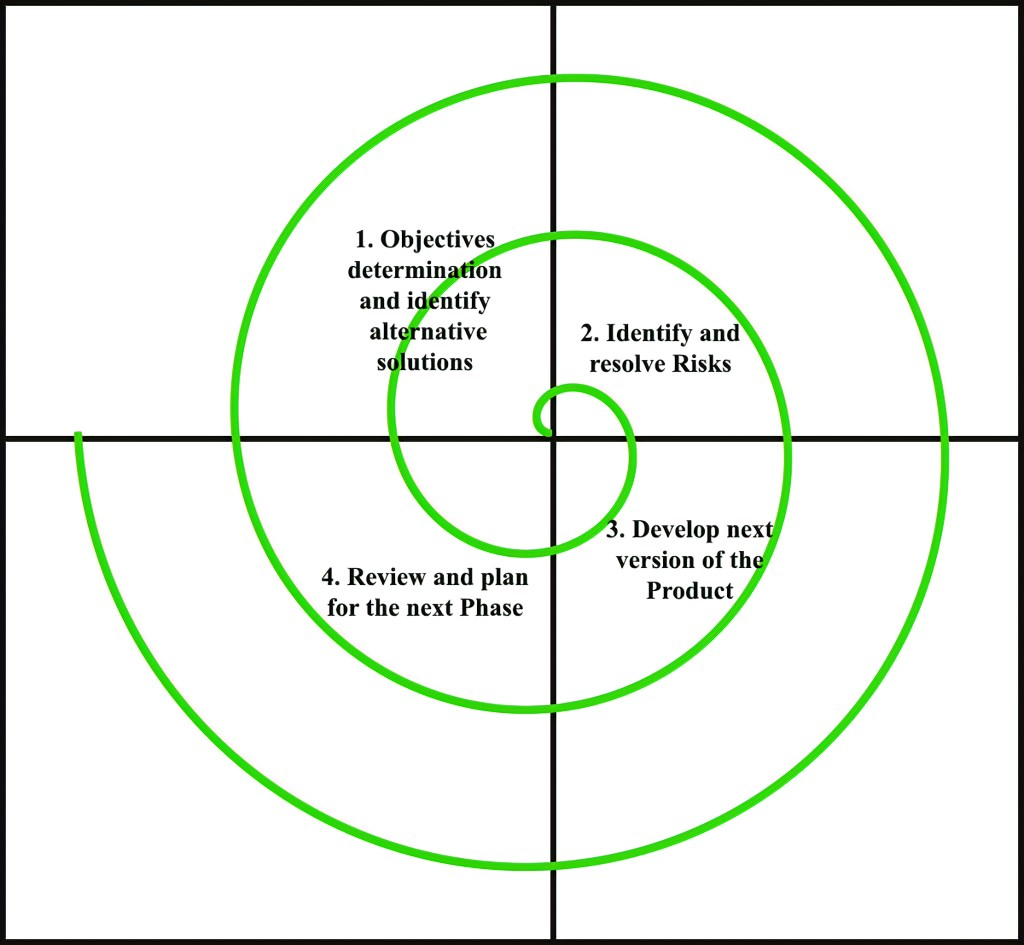
1.

a) Acceptance testing of a bus ticketing system. The booking, seat reservation, bus routing, price adjustments, etc. are tested by the clients to check if they work correctly and satisfy the requirements.

b) Acceptance testing of a car dealership. The display of each vehicle, their availability, contact forms, ability to remove or add a vehicle, etc. are tested by the clients to check if they work correctly and satisfy the requirements.

c) Acceptance testing of a library management system. The display and availability of books, reservation, borrowing and returning, etc. are tested by the clients to check if they work correctly and satisfy the requirements.

2. If we are building a software for a nuclear power plant, I believe the spiral model would work the best. This model is similar to the Agile mode – incremental development (and testing). This model is suitable for highly complex and large system, which fits the description for a nuclear power plant’s software. Furthermore, it is flexible, allowing us to change, add, or remove functionalities and features in later phases of development. The most important characteristic is that this model places heavy emphasis on risk analysis, and risks are what we want to avoid in a nuclear power plant. Although this increases the overall cost of the project due to employment of specialized experts, it will enable us to detect and fix any errors or faults early.



Source: www.geeksforgeeks.org

3. Since no specific models were specified, I have grouped them into the following:

Waterfall, RUP, and other iterative models: Customers can test the product at the final stages of development, when the product has almost finished. Customers only affect the final phases of the process.

Agile and other incremental models: Customers can test the product at an interval during development, in between sprints. The customers would have more influence and impacts on the development process.

4. Continuous beta, or perpetual beta, is a practice of software development that keeps software or systems at beta development state for an extended period of time, not releasing it as complete. This is a result of a rapid and agile approach to development.

This practice can be often found in software that requires constant updates to maintain its usability, which leads to continuous beta’s “rather political” relationship with open source software. Open source developers adhere to the practice of “release early and release often”, where new features are incrementally added to the product on an usual basis.

5. I would say that every part of a system is equally important; however, the core of the program and what the customers want the most should be prioritized. Let take an Android application for example. If the backend – server-side – is broken, there would be no information to show the customers. The backend might also contain sensitive users’ data, such as address, social security number, banking details, etc. which would be catastrophic if leaked out. On the other hand, we cannot neglect what the customers desire. If the backend is well-desgined and sophisticated, it would still not be usable if the frontend is confusing and unpleasant. Therefore, a good team of testers should always test a system thoroughly, examining these components with great effort.