**LEVEL 1**

Object: Cup

Environment: Kitchen room

Requirements: Cup should be white. Volume should 0.5L. The form should be half rounded. The material is ceramic. Should be able to hold liquid. Can be usable for drinking.

*The liquid in most use cases would be water*

Functional testing: (we test if our cup corresponds to stated requirements. If not, then our product is already low quality)

* cup is white
* cup can hold liquid
* person can pick up the cup
* liquid can be drunk from the cup
* material is ceramic
* can hold volume 0.5l
* form half rounded

Performance testing: (if a cup can handle the load from outer forces, and keep it in work state)

* Cup full of liquid
* Cup full of hot liquid
* Cup full of cold liquid
* If handle can handle full cup
* If handle cup for some time
* If put with different force cup on the table
* See if the cup will break from dropping

Usability test: (find out, if using of cup is pleasant (due to it`s personal experience, still some general stuff can be tested))

* If the holding of the cup feels good
* Cup holds comfortable in different palms

Security test (if cup would not be dangerous for user)

* Cup can be held if it purred with hot liquid
* Cup can be held if it purred with cold liquid
* Cup without sharp edges from manufacturing
* Cup without toxic wastes after

Compatibility test: (see if we can use in our daily life cup with different products)

* Cup can be washed in a sink
* Cup can be washed in a washing machine
* Cup can be heated in a microwave
* Cup can be cooled in a fridge
* Cup can be placed on a shelf
* Cup can be placed on hang pole for cups

**Validation** – if created requirements corresponds to the client/user desires

**Verification** – if created part of product responds to the specification and requirements

**LEVEL 2**

|  |  |  |
| --- | --- | --- |
|  | pros | cons |
| Product | * Deep knowledge in the company's product. * The circle of professional acquaintances is wide * Due to the use of one technology, there are less risk for unwanted results * Get used to taking care of the product | * Salaries can be deviating from highest to below the border. * There comes a time when the work becomes boring. |
| Startup | * More opportunities to learn * Flexible hours * Unique experience * Minimal supervision * Opportunities for innovation | * Uncertain job security * Long hours and less payment * Constant changes and limited resources |
| Outsource | * A variety of projects provide an opportunity to develop and learn new things * Opportunity to use new technologies * Get new experience in communications and interaction from project to project. | * Work for speed, first of all * Without the ability to plan your activities, the chances of making a mistake increase. * No orders, no money |
| Outstaf | - | - |
| Academy | * Opportunities to summarize your knowledge * Get the trainer experience for future position improvement (QA, Lead and etc) | * A lot of communication and preparation would be needed for lessons * Management of the learning team can be tedious |
| Recruitment Agency | - | - |

**Validation/Verification example from real life**:

I will give an example a little bit local but try to explain it as well as possible. It`s from my current working place.

The part of our process, we should create anatomical 3D models from computer tomography images and later create patient-specific guides and implants from them. Creation instructed by SOP that defines, what parts should be created and how accurate the chosen type of surgery should it be

**Verification** would be here if all needed parts were created for the case (skull, teeth, lower jaw and etc)

**Validation**, if these parts are created with the right accuracy for future design in dependence from surgery type

So, if we don`t have all the needed parts or the accuracy deviates more than needed, Verification/Validation would be failed

**LEVEL 3**

**Exhaustive testing**

In theory, exhaustive testing is impossible. To implement this kind of testing, you need to test a product with all possible input data and scenarios whose quantity is not infinity but close to it (due to the very large number, it can be named infinity)

**Early testing saves money**

The earlier we will find a possible defect in our product, the cheaper from effort and budget side it will be.

In the design step, it will be a few minutes to solve this problem. On coding step, it can take hours. Test step – also add some hours from a tester to find a defect and return it to the development step and test it again after.

When we get on release state, and we found this defect and it needed to be fixed, it creates not only huge amount of time needed for rollback to the previous steps and additional budget spending for working days, but also it delays product release and possible happiness of client (if it depends on money lost or burned terms for client due to delay or discovered bug)