**Lesson 2 Cíntia Domingos**

**Beet Seed**

**1.** Object chosen: French horn

Tests:

- Insert the mouthpiece into the lead pipe: does it fit snugly? It needs to fit perfectly, so the air does not leak

- Blow into the mouthpiece: the air should come out of the bell and make a sound

- Put the right hand into the bell: is the surface smooth and comfortable? Can you close the bell pipe with your hand or is it too wide? It should be possible to close the bell pipe with the right hand to perform a specific technique (called *bouché*)

- Remove the tuning slide and try to play. It should not be possible, no sound should come out of the bell

- Remove every slide and try to insert them incorrectly: it should not be possible. Each slide should only fit its own pipe perfectly.

- Press and release each valve lever individually: do they go down and come back up? How much pressure is required to press them? They should be light and soft, allowing for speed.

- When a lever gets stuck and doesn’t go up or down, put oil on it and try again. It should be able to move again.

- Press the first valve lever together with the thumb lever – check they don’t touch when pressed, so they don’t interfere with each other when the instrument is being played

- Experiment with the levers while blowing on the mouthpiece – does it make a sound? Do the notes change?

- Drop the horn to the floor: it should not be too damaged from a single drop

- Hold the horn for a long time: is it comfortable to have the left hand placed on the valves and on the finger rest? Are your fingers too stretched out or too close together? Is it too heavy?

**2.**

**Validation**: validation checks if the requirements for the product match the needs of the customer. In other words, it means checking if the product being made is the right product for the customer’s needs.

**Verification**: it means checking if the product is being made correctly, if it fulfils the requirements.

**Beet Sprout**

**2.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Type of Company** | **Pros** | **Cons** |
| 1 | Product | * Create company’s own product * Familiarity with the processes, technologies used in this company * In-depth knowledge related to the product * Stability and predictability * High pay * Other benefits (health insurance, snacks, schedule flexibility) | * Less variety of projects * Less flexibility in choice of technology * Less innovation (focus often on maintaining existing products) * May have a more rigid hierarchy |
| 2 | Startup | * Possibility to make a lot of money if the company is successful * High pay * Innovative projects * Working with the latest technologies | * Risk of the company/product failing * Uncertainty in terms of the future * Instability * Requires a lot of experience * No availability to train juniors * A lot of work in the beginning * Limited resources |
| 3 | Outsource | * Diverse projects * Broad skill set development * Networking opportunities * Stability | * Limited control over projects * High pressure to meet clients’ requirements |
| 4 | Outstaff | * No geographical borders * Remote work * No intermediaries like in outsourcing * Possible higher pay * Flexibility * Freedom to choose the projects on which to work * Professional development | * Possible instability * Frequent job search * More competition (candidates from all over the world) |
| 5 | Academy | * Provides training * Supportive workplace environment * Satisfaction from helping other people * Predictable schedule | * Possible lower pay * Smaller career growth |
| 6 | Recruitment Agency | * Exposure to wide range of IT roles * Networking opportunities * Insights of the IT market (salaries, skills demand, emerging tech) | * High pressure to achieve targets * Having to inform candidates of rejections * Lower pay comparatively to a tech role |
| 7 | Contractor Company | * Training and certification * Support from the main company * Opportunities to change job location | * Bureaucracy and legal requirements * Stability |

**3.** An example of an unsuccessful verification I experienced was when I bought a game on Steam and it would crash every time on start-up, meaning I was not able to play it at all even though my computer met all the required specs and updates. I found out that thousands of people had the same problem as me and were asking for a refund of the game, which is a huge mishap for the company, causing it to receive lots of negative reviews, lose the confidence of the customers and having its reputation tarnished. This emphasizes the importance of considering as many scenarios as possible during the tests. This issue suggests that a critical defect exists in the software, which was not found during the verification process and leads to a failure in meeting the software requirements.

As for an example of an unsuccessful validation, I could mention the platform that my university used for online student enrolments. When a significant number of students were trying to enrol at the same time, the website would crash and take a long time to recover. This would happen continuously for the first couple of days of enrolment when the number of students accessing the platform was bigger. The server that should be able to accommodate a large number of users simultaneously was becoming overwhelmed, which means a requirement for the server capacity was overlooked during the development. There was no error message, warning or guidance on how to deal with this, which led to the students being frustrated and losing vacancies in the classes they wanted. Consequently, the phone number for support was also burdened with constant calls of people looking for help.

**Mighty Beet**

**2.**

**“Testing show the presence of defects, not their absence”**

This principle indicates that testing can show the existence of some defects, but it does not prove that those are the only existing defects. There could be more that have not been found and testing only reduces this probability, it doesn’t erase it.

As a translator, I can apply this principle to my job. Reviewing (“testing”) a translation and finding a grammatical error or a typo doesn’t mean there won’t be more mistakes and typos or even translation mistakes. Just like in testing, reviewing the text also only reduces the probability of mistakes being sent to the end-customer.

**“Exhaustive testing is not possible”**

It is usually not possible to test every single combination of every single possible input/output/condition/data in a system. The test planning needs to prioritize the test objects and know when to stop.

This principle can also be applied to translation. A translation is never done, there is always something that could be improved, but it’s impossible to keep working on the same text indefinitely, because the end-customer usually wants it “for tomorrow”. The translator needs to prioritize the challenges they face, find solutions in a timely manner and know when to stop thinking of better ones.

This can also be applied, for example, to travel planning: it is not practical to consider every combination of every existing route. The travellers need to focus on the main route and on packing essentials. They don’t need to plan every single detail of every minute of the trip to enjoy it.

**“Beware of the pesticide paradox”**

According to this principle, test cases are like a pesticide: if they are used too much, they will lose their effect and become useless. If a test is run indefinitely, there will come a time where it will not find any bugs, but this doesn’t prove they are there. Test cases need to be updated and/or improved, just like the software is updated/improved.

I think this principle applies very well to students. Students shouldn’t always use the same material to study or review for exams. They should introduce new notes, new questions, new strategies to increase their knowledge.