**Moolya WAY of testing:**

Routine school -- > focus more on progress

Quality school -- > focus more on quality & process

QA will have the priority on release

Analytical school -- > more focus on logical & Testing techniques

**CDT school (Context driven testing) -- Commercial market driven software testing**

**Ex: Amazon, flipkart, mintra**

* Testing provides information to project
* Testing finds bugs
* **A bug is anything that could bug a stackholder**
* Testing is skilled, mental activity

Expolratory testing – Concurrent test desighn and execution

Rapid learning

Implications:

* Expect changes, Adapt test plans based on results
* Focus on skills over the process
* Un challenged assumptions are dangerous
* **Assume nothing, Questions eveerything**
* **Question every thing in the sence quetioning developer on the issue without agree to his answers**

**Moolya way of testing is based on CDT**

**Usuablity testing:**

Usability testing is a method of testing the functionality of a website, app, or other digital product by observing real users as they attempt to complete tasks on it.

Aproach of Usability for different schools:

**For**

CDT:

* Definitely Do it
* Usability bugs are bugs

Routine:

* Do it if requested by management

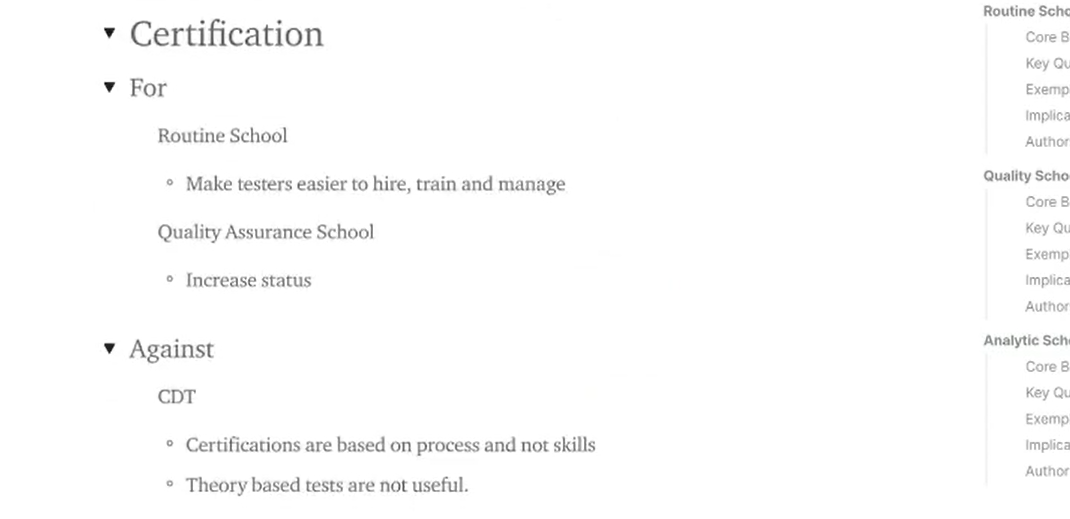
Againest:

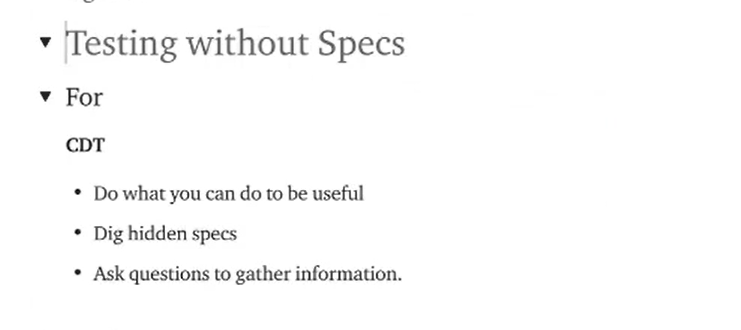
Analytical:

* Not a form of testing
* Outside testing let some one do it

QA:

Relucant because hard to proove non-compliance

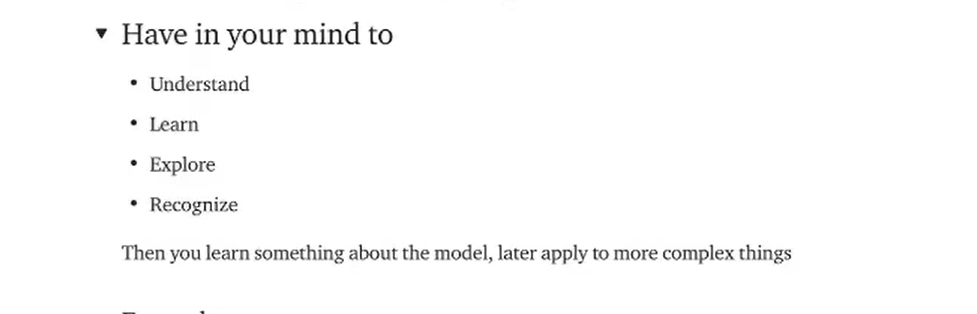




**Model:**

A simplified representaion of more complex things

Like prototype, representaion of an idea or process



**Examples:**

* Google maps
* calendar
* Weather forecast
* wireframes
* Requirement docs
* Test ideas/test cases

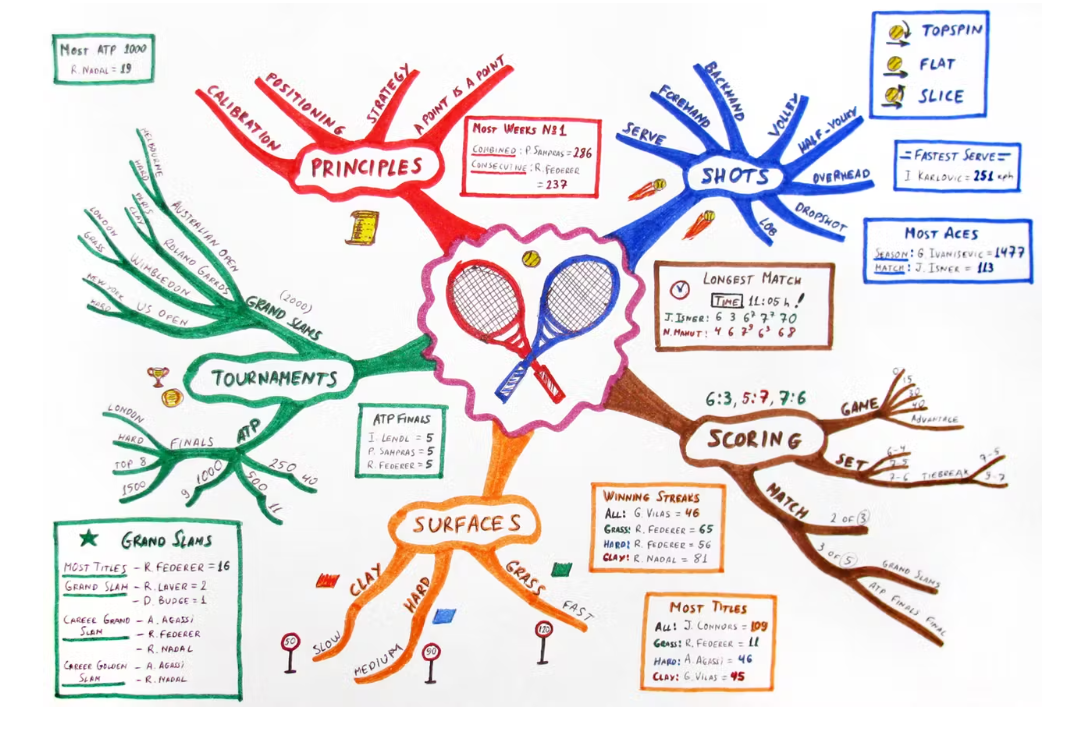
**Mind maps:**

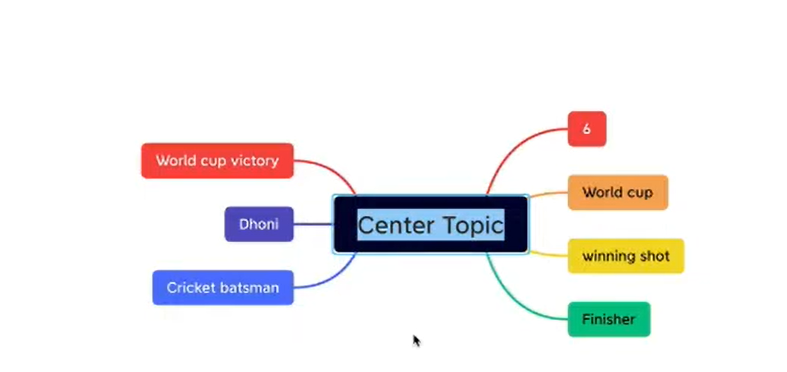
Memory : is non linear & associative

* Visually organize information
* Hierarchical & captures relationships
* Mindmaps is a way of modeling
* Radical , Non linear, and intituiley arranged

Examples:

Tennis:





**Models in testing:**

Models are the central to proccess of knowledge building

Use existing models to understand about the software you have to test:

* Requirement documents
* Wireframes
* Mock screens
* Architecture diagrams

You build models how to test and what to test:

* Test cases
* Test coverage
* Test stratagy

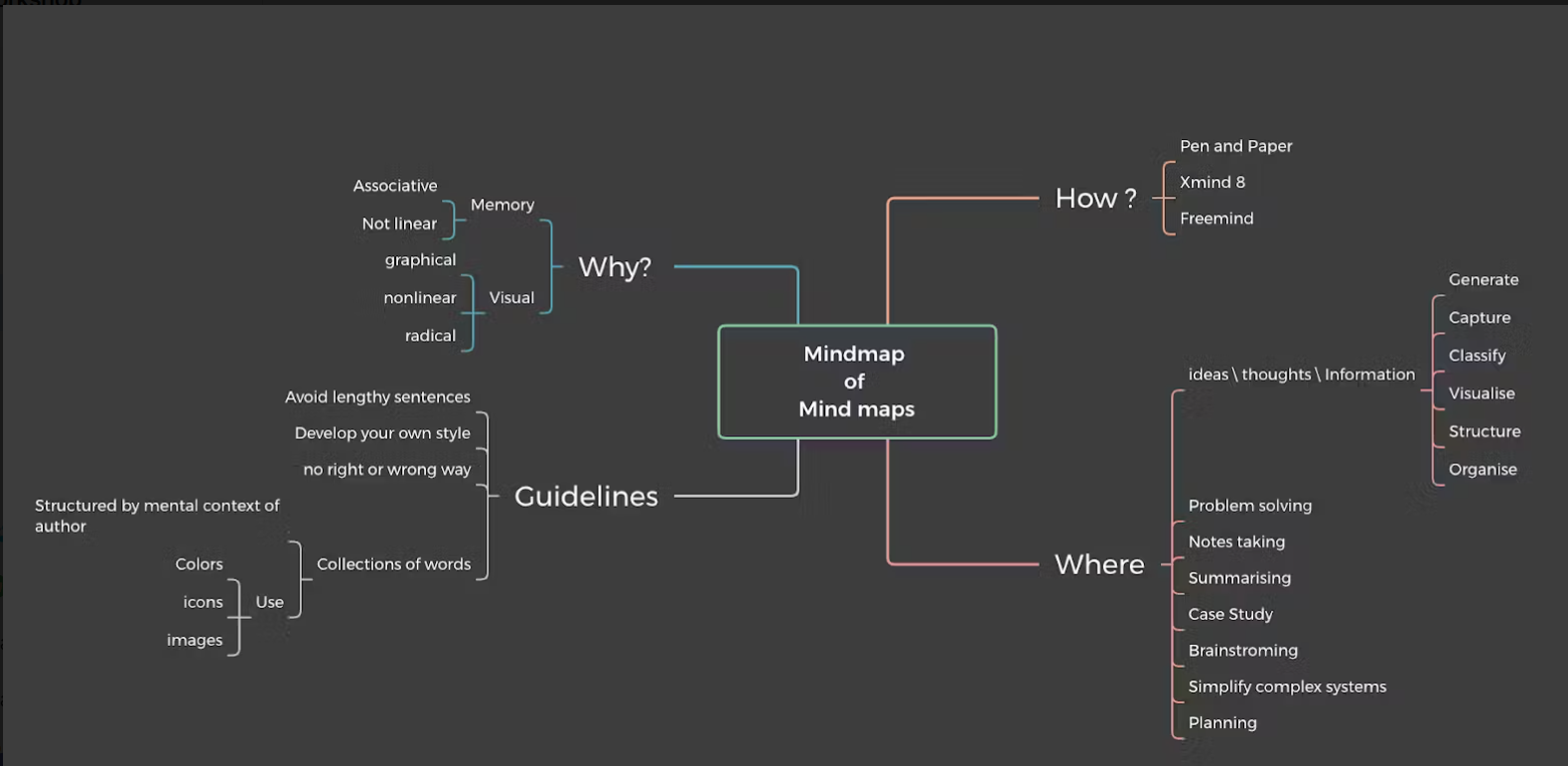
Avoid while creating mind maps:

* Avoide lengthy sentences

Tools used to create mindmaps:

* Xmind

How to create mindmaps:



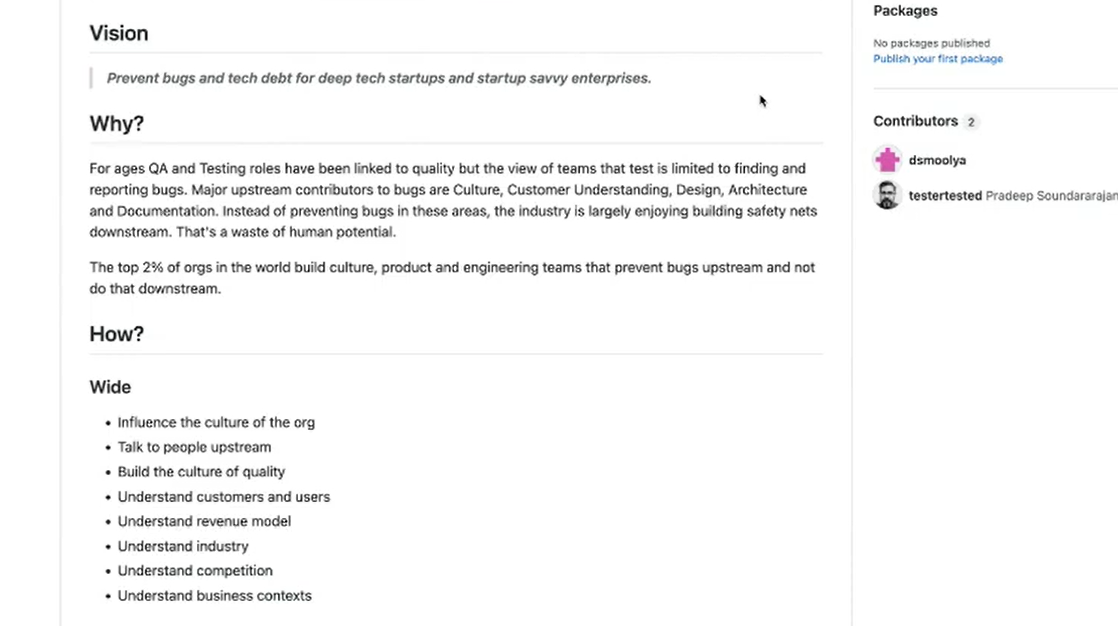
**Moolya vision: preventing bugs and tech debt for deep tech startups and savy enterprises**

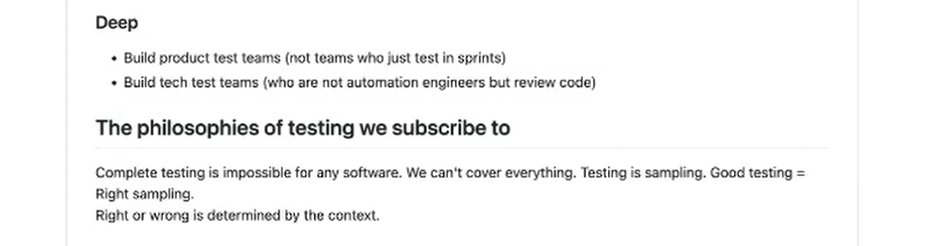
**Tech debt:**

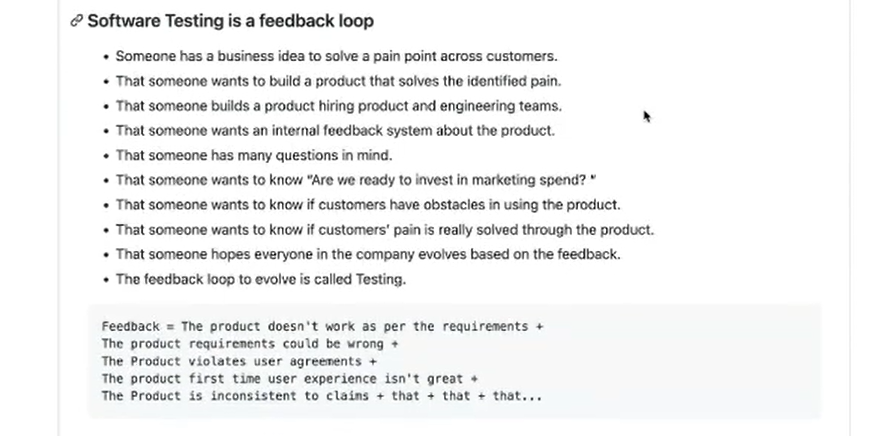
Technical debt (also known as tech debt or code debt) describes what results when development teams take actions to expedite the delivery of a piece of functionality or a project which later needs to be refactored. In other words, **it’s the result of prioritizing speedy delivery over perfect code.**

**Deep tech:**

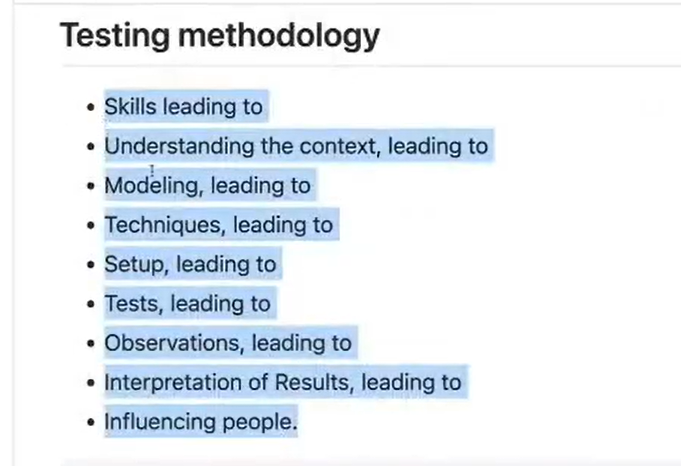
Deep technology (also deep tech or DeepTech) or hard tech is a classification of organization, or more typically [startup company](https://en.wikipedia.org/wiki/Startup_Company), with the expressed objective of **providing technology solutions based on substantial scientific or engineering challenges.**

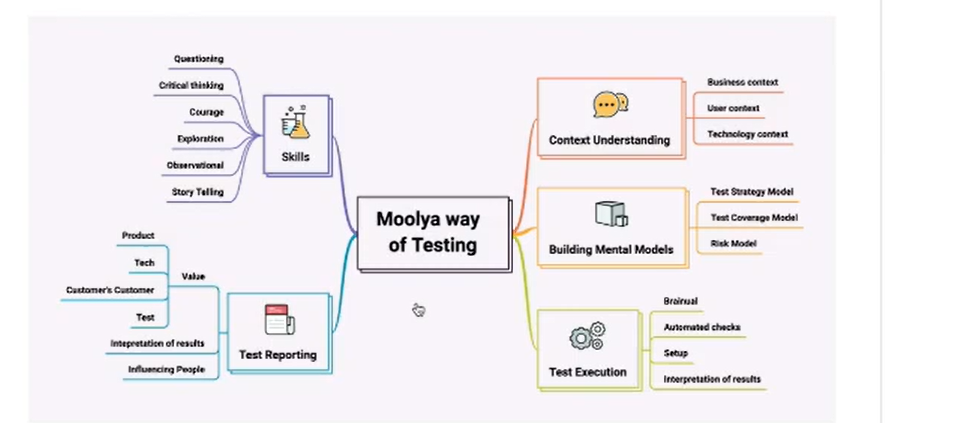
****

****

****

**Test Methodology:**

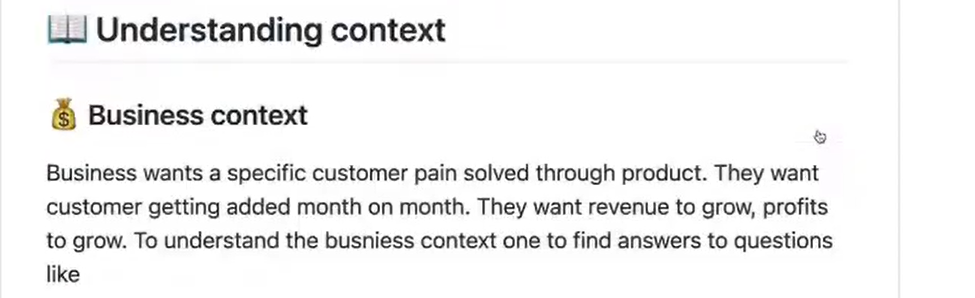
****

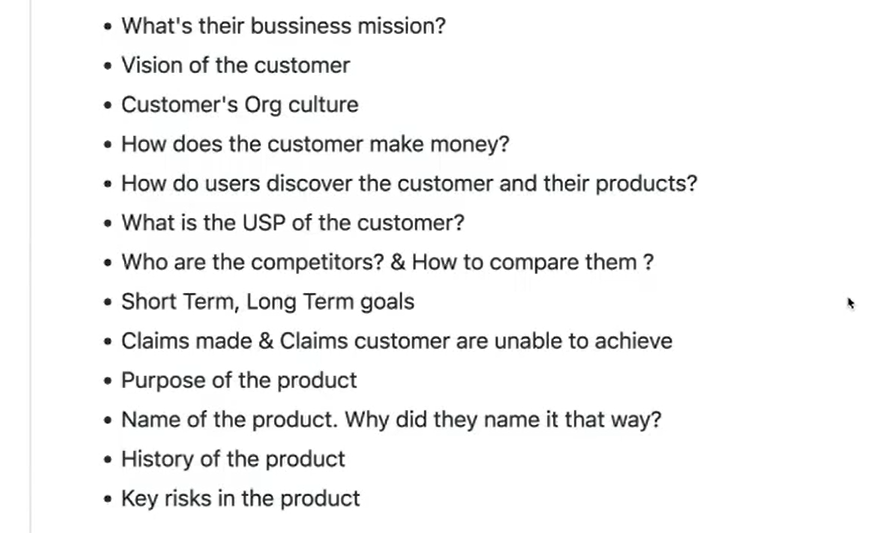
****

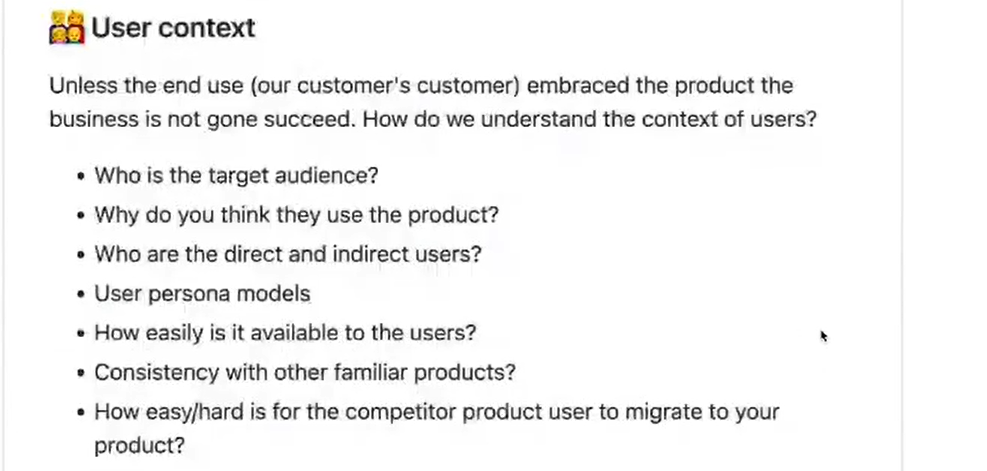
Always stratagy(plan) works on context. If the context changes the startagy should change.

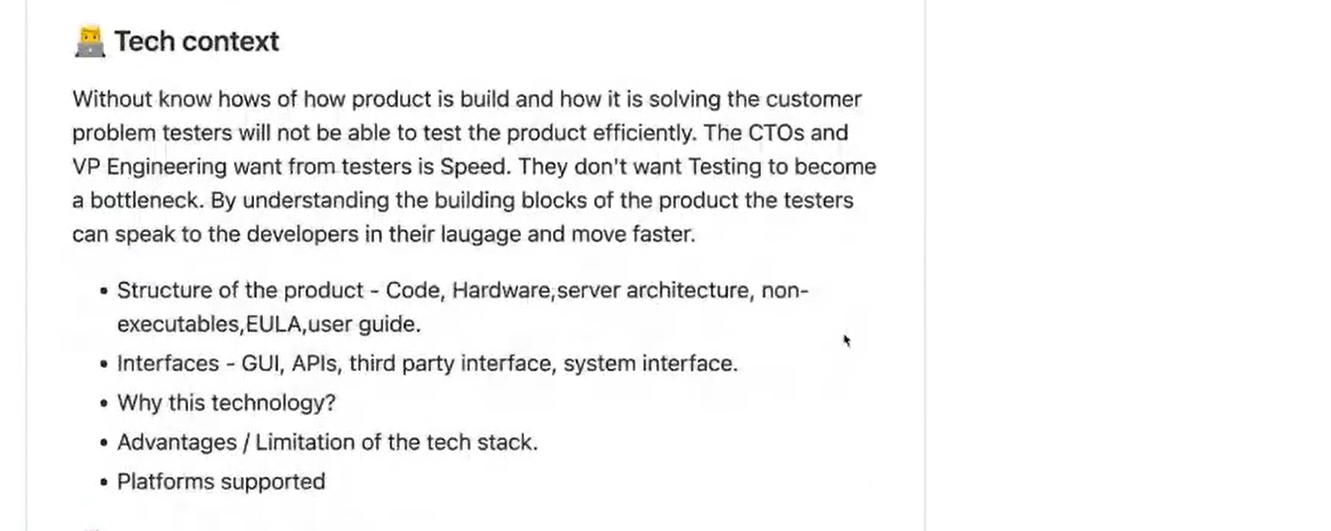
Here stratagy refers to test cases and test plan or test desighn

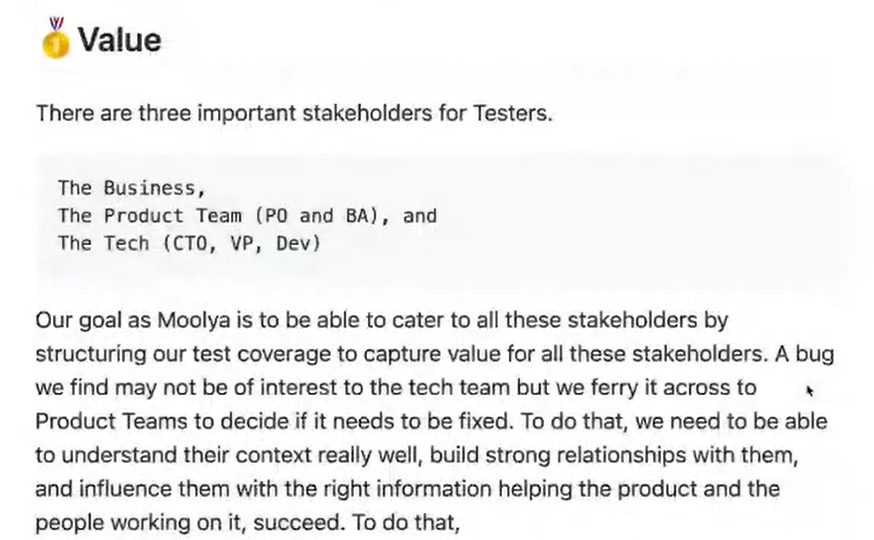
Context detrmines what model need to aproach. Requirement document is only the resource to understand context











**Stake holders for testers:**

* Business team
* Product team (PO (project owner), BA (Business Analyist))
* Tech team (CTO, VP, Dev)

**Unserstand the context**

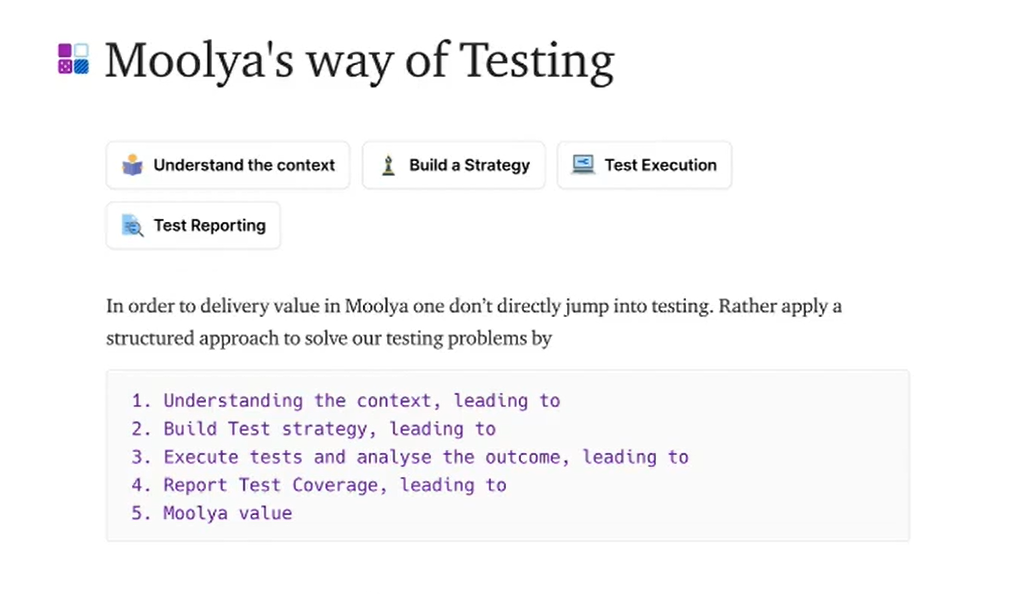
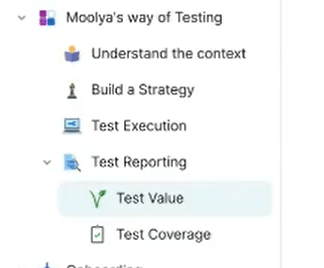
**Build the mental models**

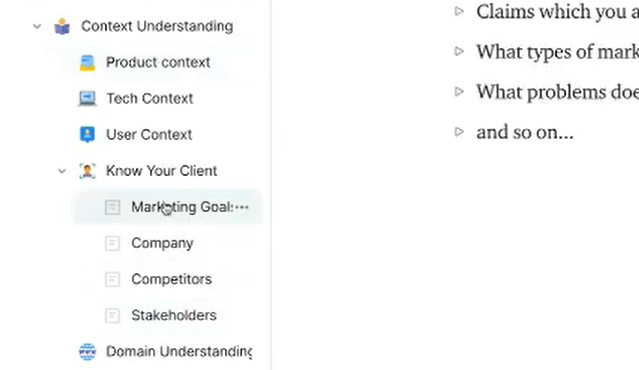
**Do the test execution**

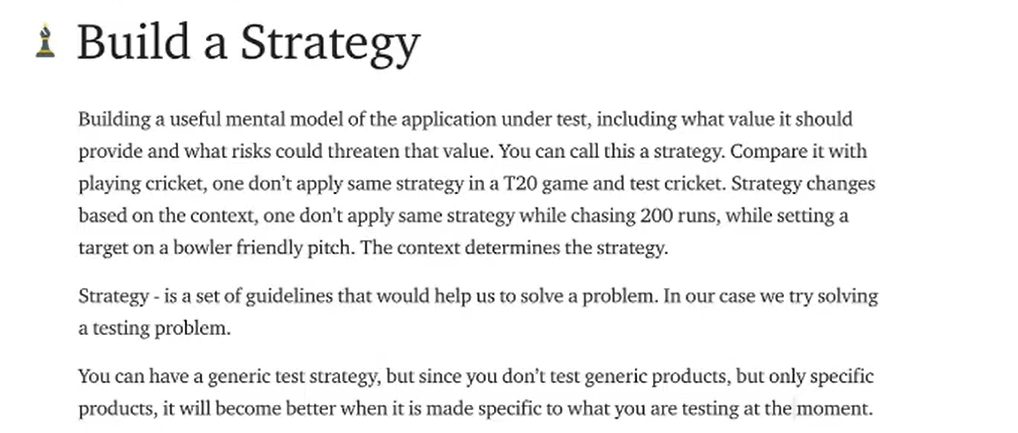
**Report the testing**

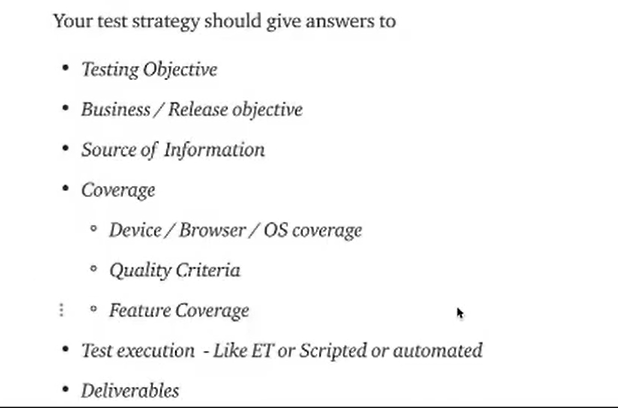
**Contextual keypad**: showing right text keypad when we entering the data in the text field

Like entering numbers showing only number keypad, while entering text showing only letters keypad

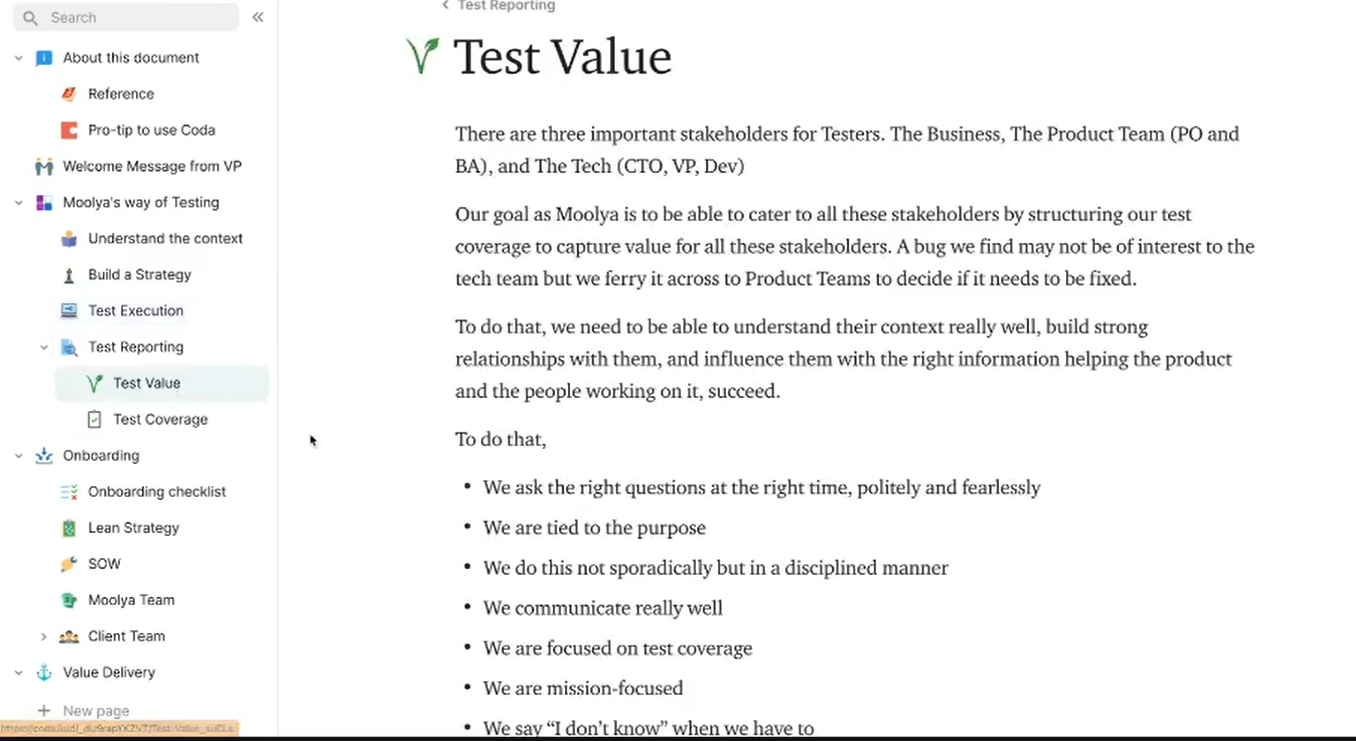
 

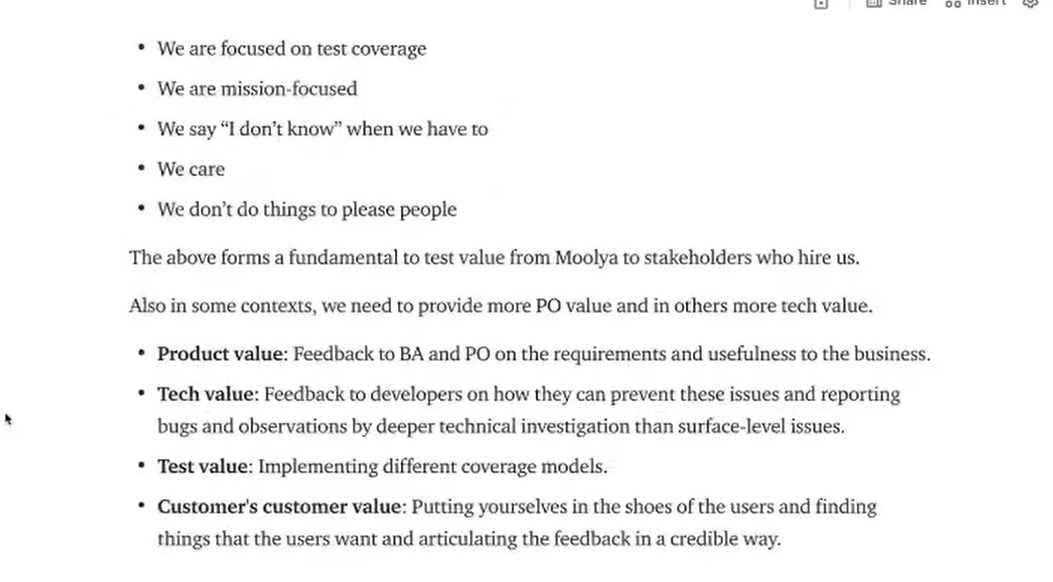


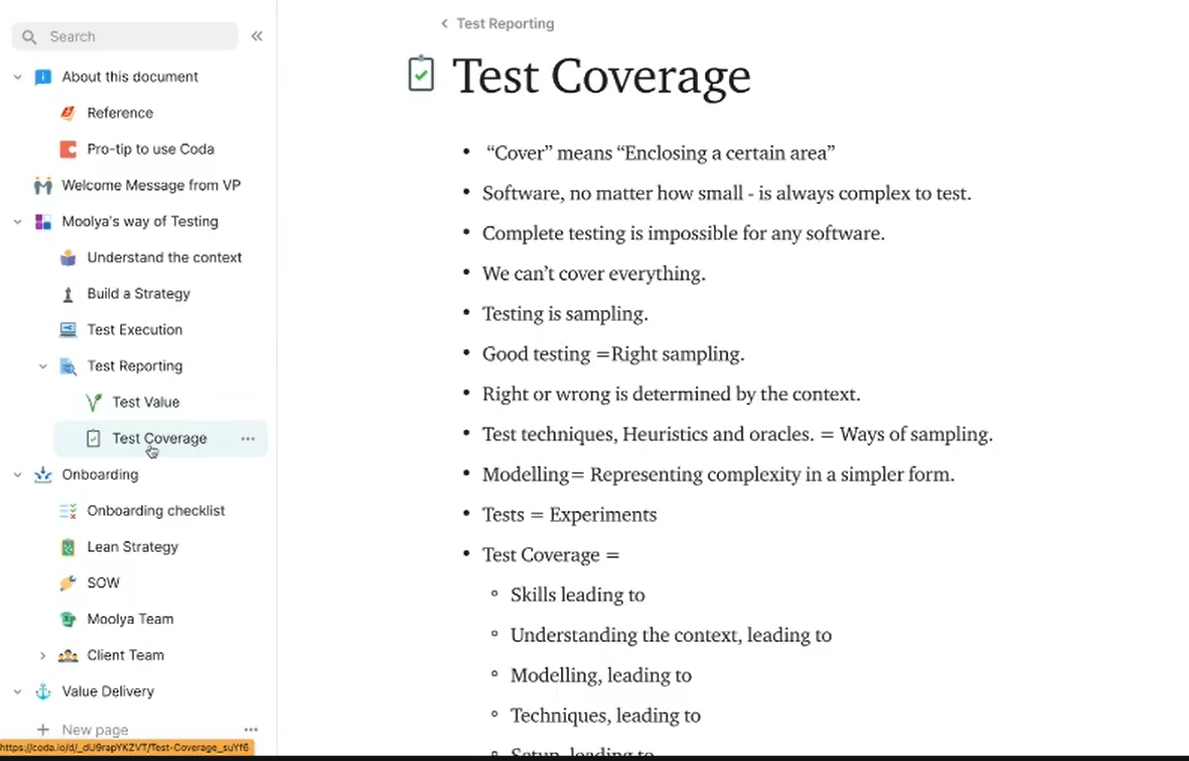












Test value & test coverage is the deliverables in Moolya