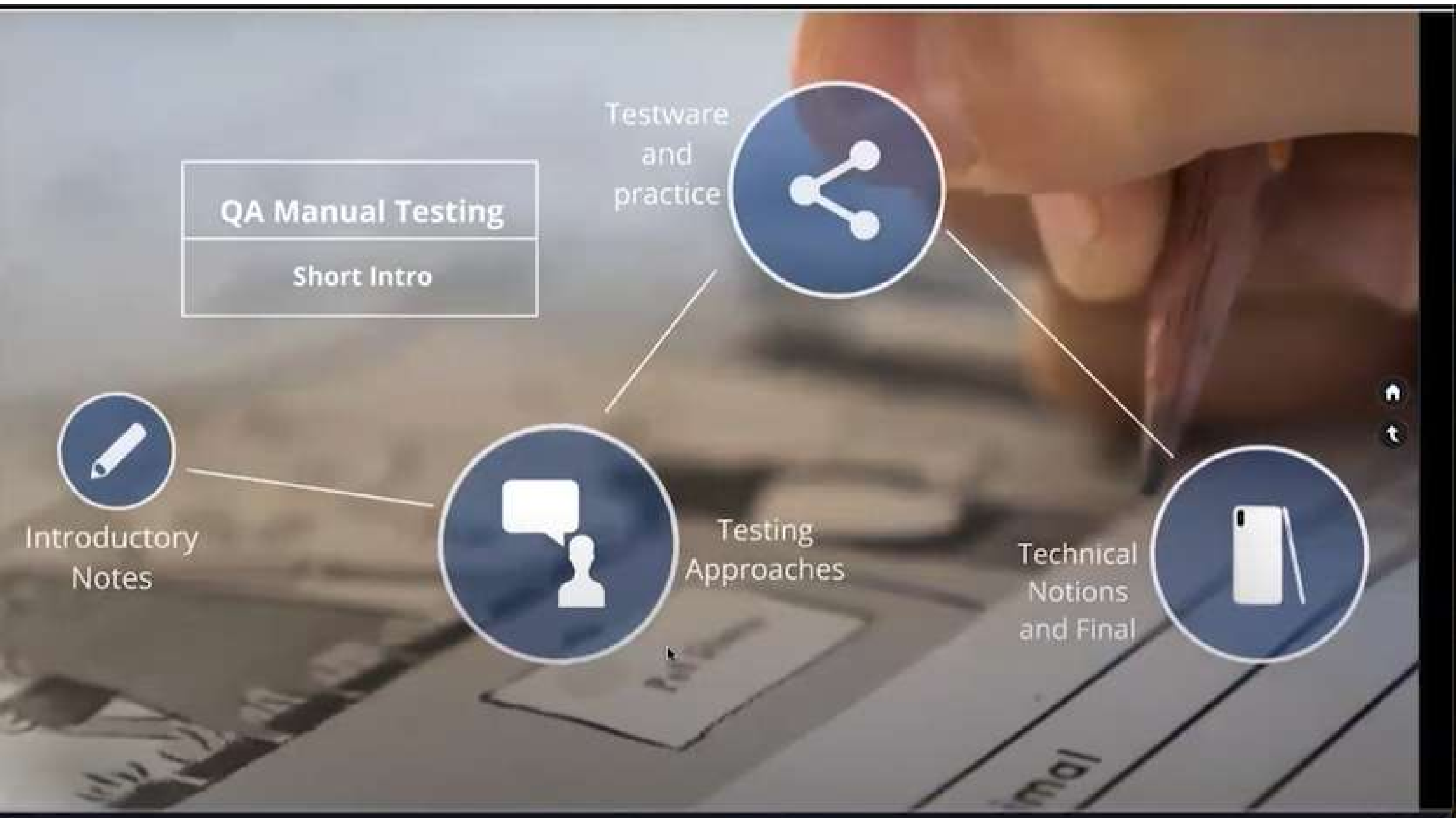


QA Manual Testing

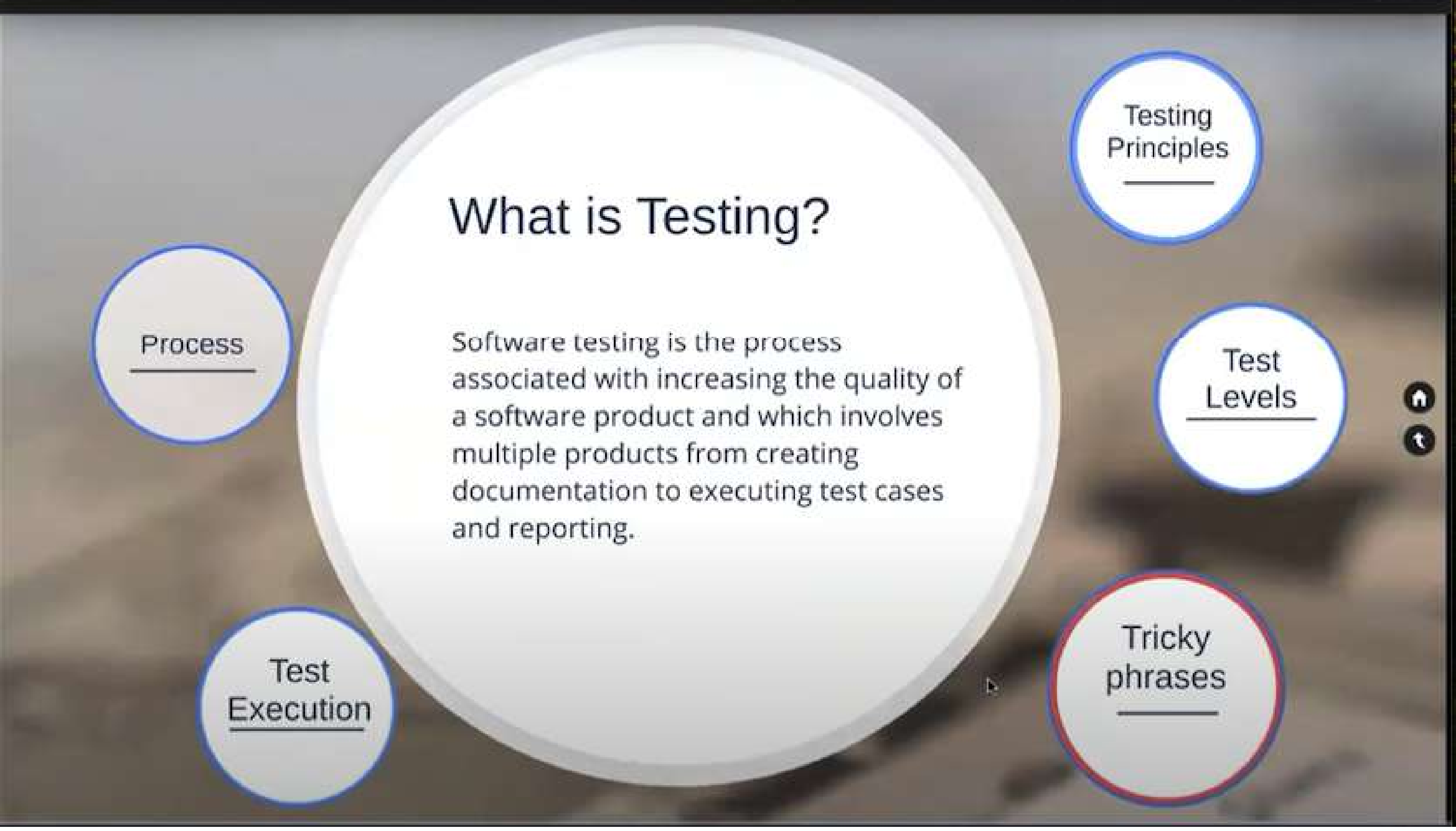
Short Intro

Testware and practice

Introductory Notes

Testing Approaches

Technical Notions and Final



What is Testing?

Software testing is the process associated with increasing the quality of a software product and which involves multiple products from creating documentation to executing test cases and reporting.

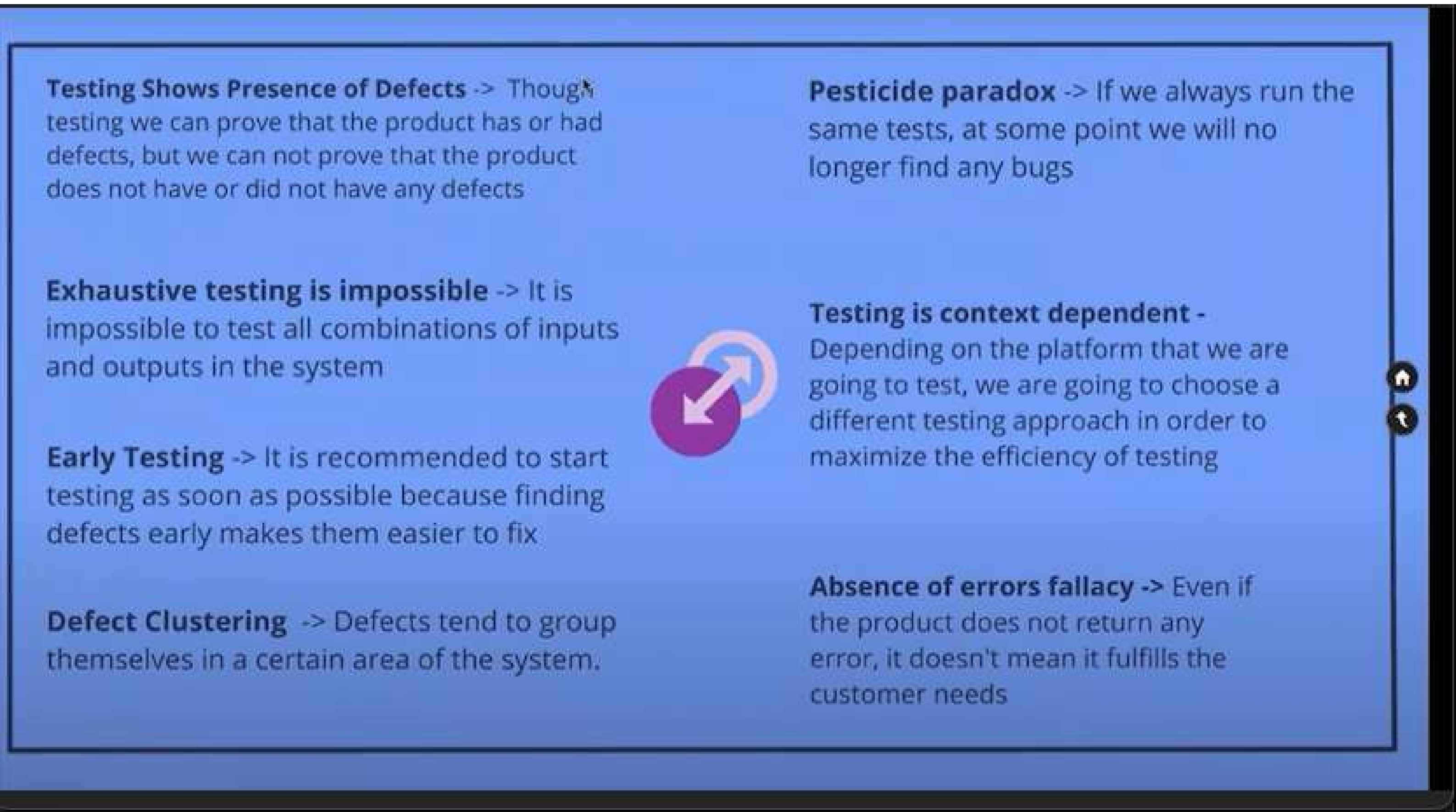
Process

Test Execution

Testing Principles

Test Levels

Tricky phrases



Testing Shows Presence of Defects -> Though testing we can prove that the product has or had defects, but we can not prove that the product does not have or did not have any defects

Exhaustive testing is impossible -> It is impossible to test all combinations of inputs and outputs in the system

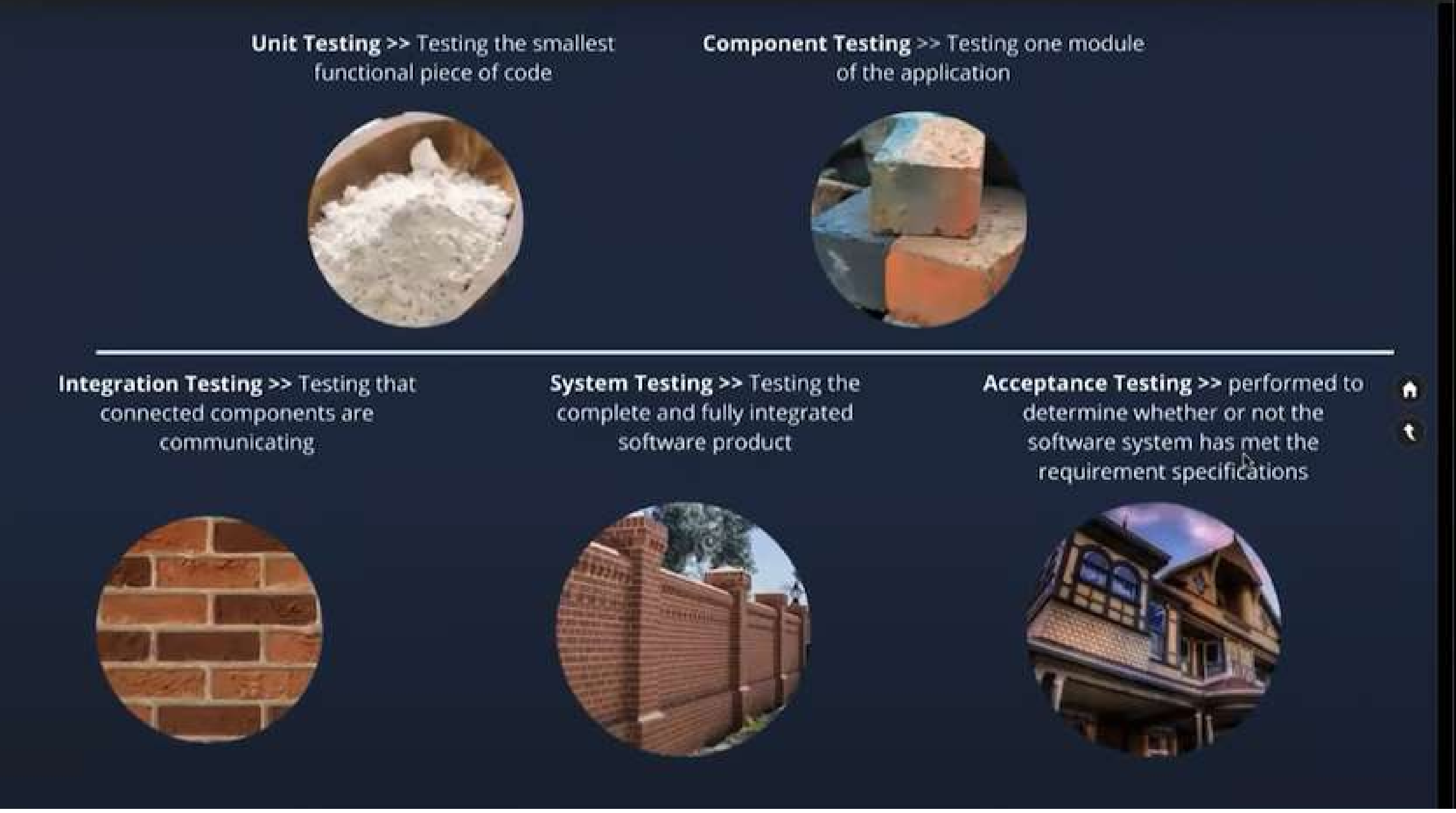
Early Testing -> It is recommended to start testing as soon as possible because finding defects early makes them easier to fix

Defect Clustering -> Defects tend to group themselves in a certain area of the system.

Pesticide paradox -> If we always run the same tests, at some point we will no longer find any bugs

Testing is context dependent - Depending on the platform that we are going to test, we are going to choose a different testing approach in order to maximize the efficiency of testing

Absence of errors fallacy -> Even if the product does not return any error, it doesn't mean it fulfills the customer needs



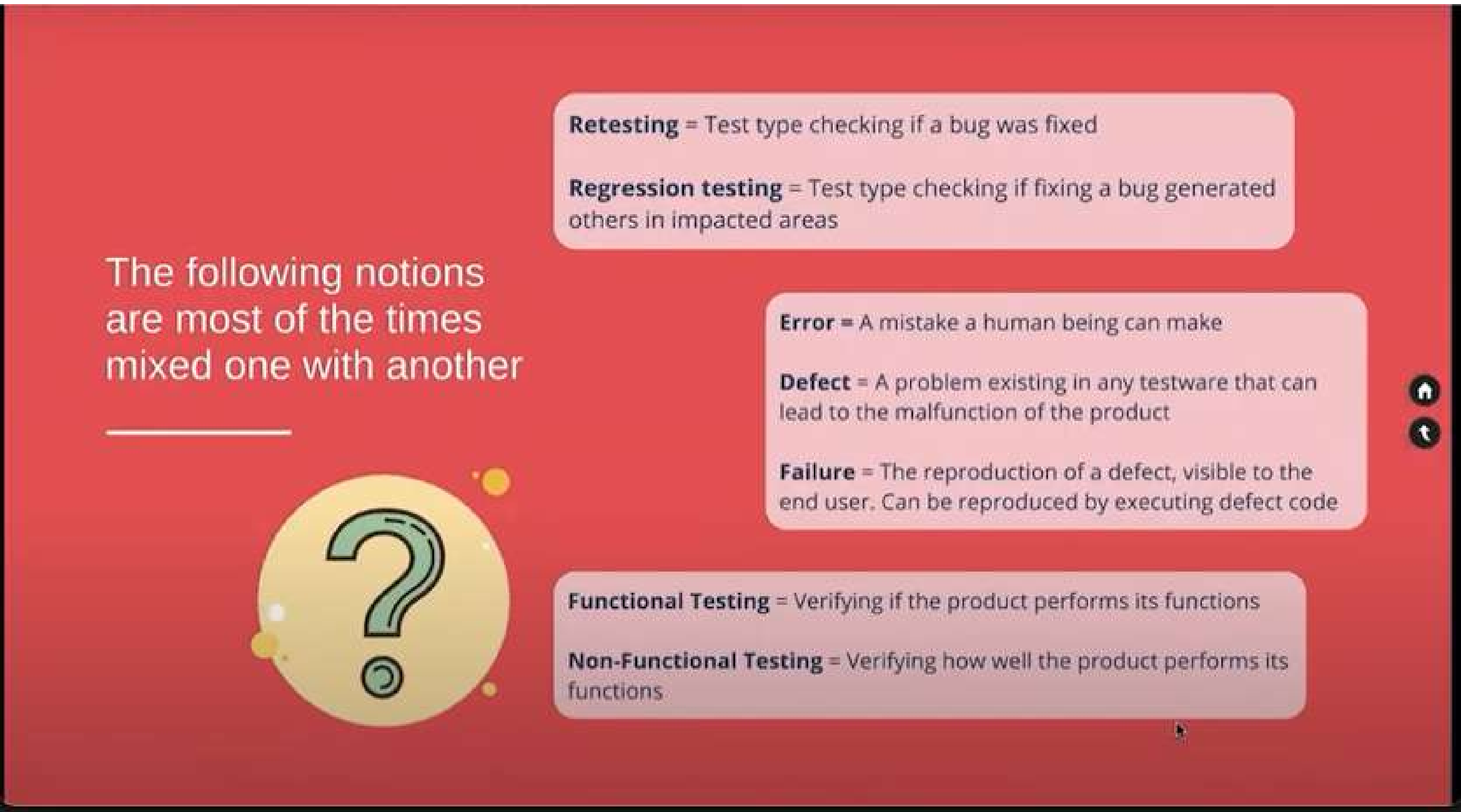
Unit Testing >> Testing the smallest functional piece of code

Component Testing >> Testing one module of the application

Integration Testing >> Testing that connected components are communicating

System Testing >> Testing the complete and fully integrated software product

Acceptance Testing >> performed to determine whether or not the software system has met the requirement specifications



The following notions are most of the times mixed one with another

Retesting = Test type checking if a bug was fixed

Regression testing = Test type checking if fixing a bug generated others in impacted areas

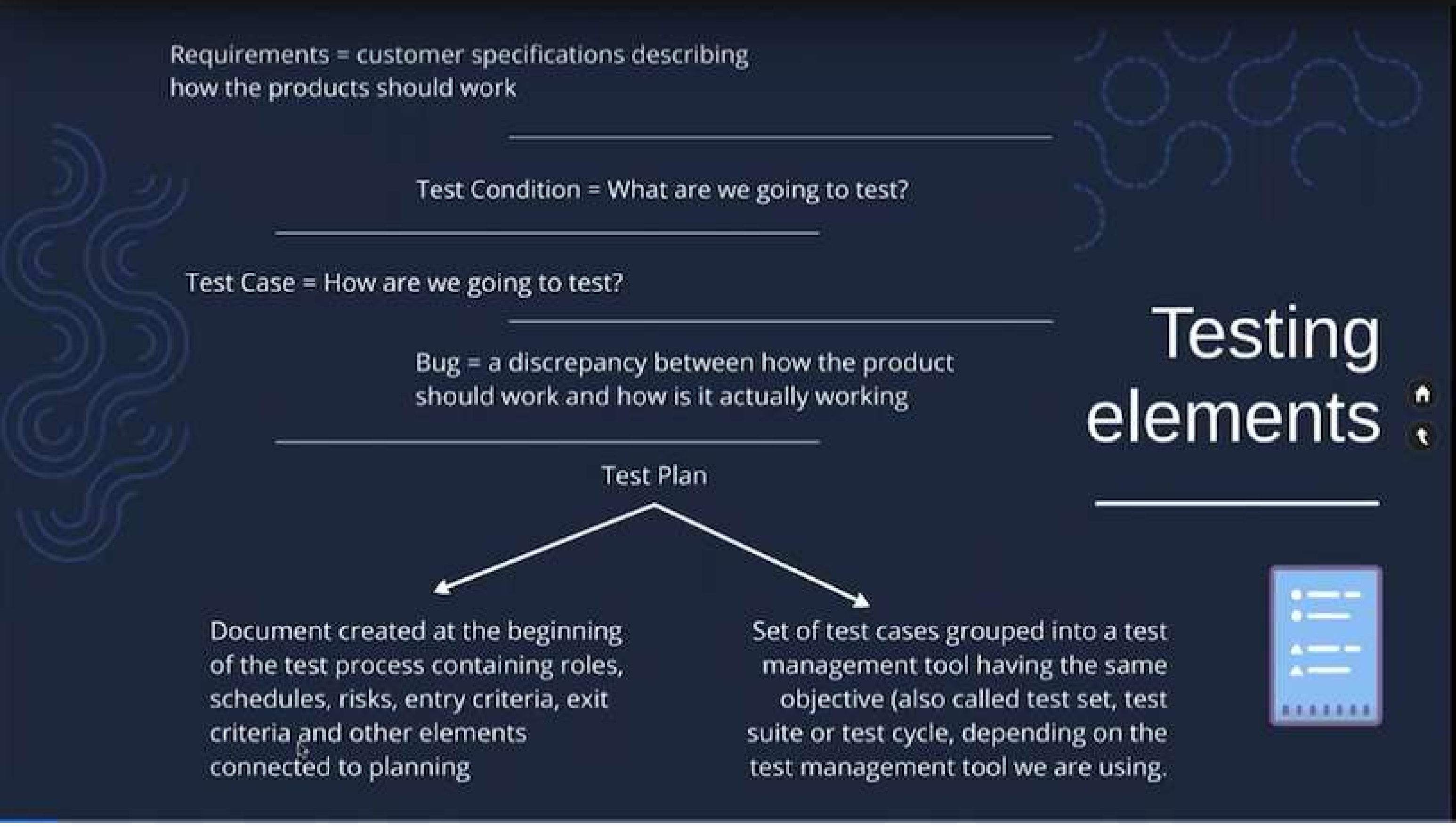
Error = A mistake a human being can make

Defect = A problem existing in any testware that can lead to the malfunction of the product

Failure = The reproduction of a defect, visible to the end user. Can be reproduced by executing defect code

Functional Testing = Verifying if the product performs its functions

Non-Functional Testing = Verifying how well the product performs its functions



Testing elements

Requirements = customer specifications describing how the products should work

Test Condition = What are we going to test?

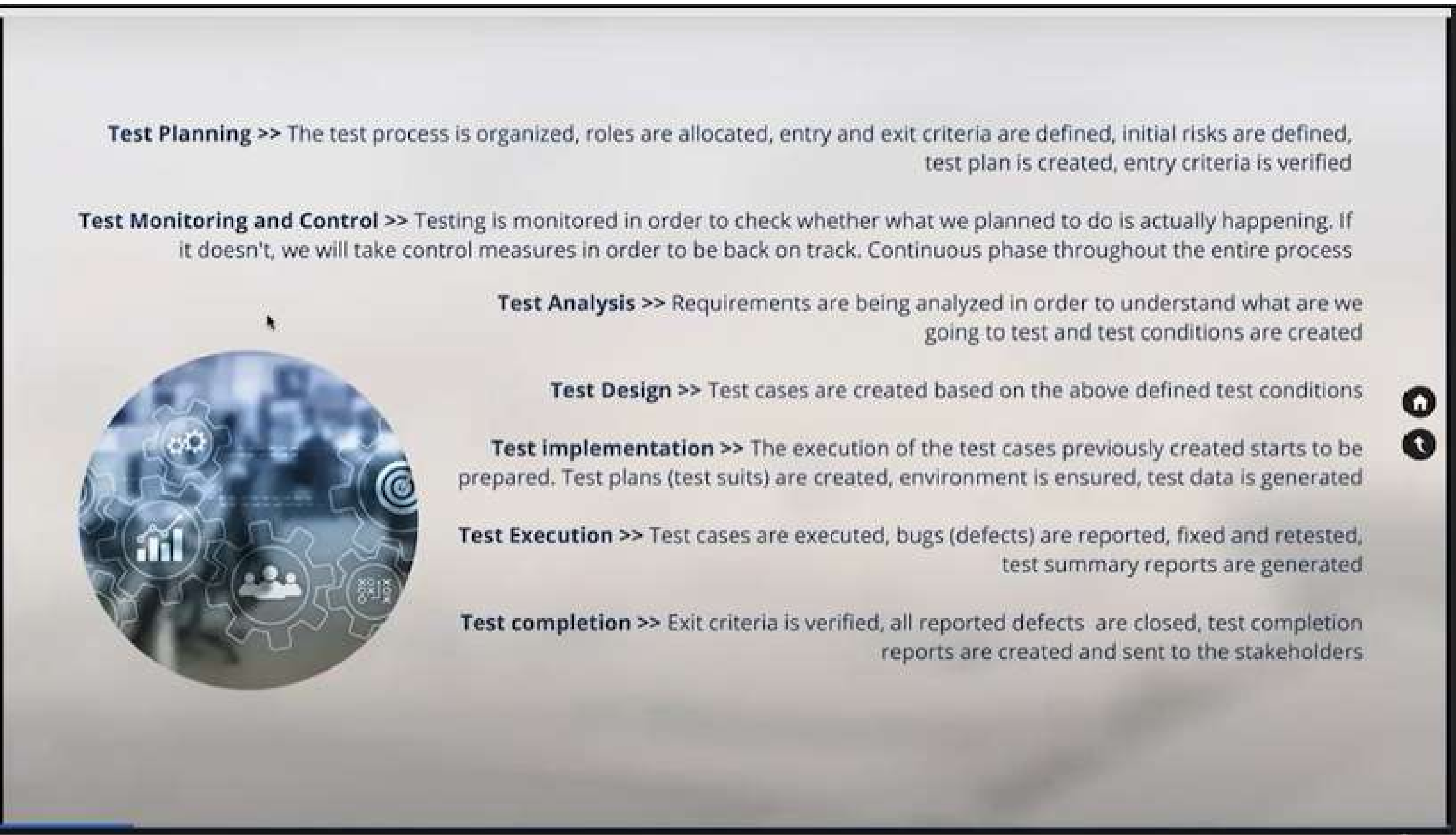
Test Case = How are we going to test?

Bug = a discrepancy between how the product should work and how is it actually working

Test Plan

Document created at the beginning of the test process containing roles, schedules, risks, entry criteria, exit criteria and other elements connected to planning

Set of test cases grouped into a test management tool having the same objective (also called test set, test suite or test cycle, depending on the test management tool we are using.



Test Planning >> The test process is organized, roles are allocated, entry and exit criteria are defined, initial risks are defined, test plan is created, entry criteria is verified

Test Monitoring and Control >> Testing is monitored in order to check whether what we planned to do is actually happening. If it doesn't, we will take control measures in order to be back on track. Continuous phase throughout the entire process

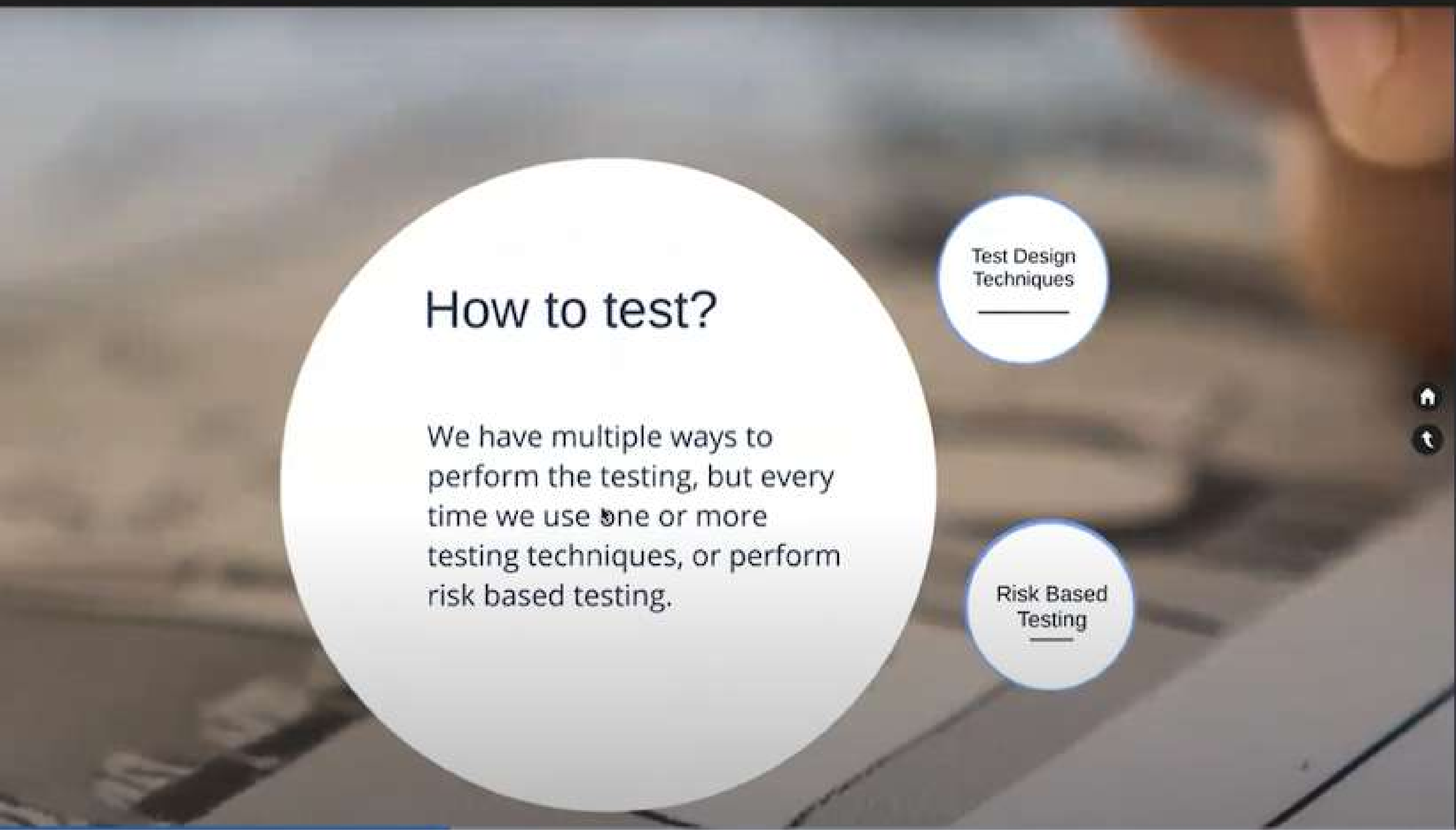
Test Analysis >> Requirements are being analyzed in order to understand what are we going to test and test conditions are created

Test Design >> Test cases are created based on the above defined test conditions

Test implementation >> The execution of the test cases previously created starts to be prepared. Test plans (test suits) are created, environment is ensured, test data is generated

Test Execution >> Test cases are executed, bugs (defects) are reported, fixed and retested, test summary reports are generated

Test completion >> Exit criteria is verified, all reported defects are closed, test completion reports are created and sent to the stakeholders

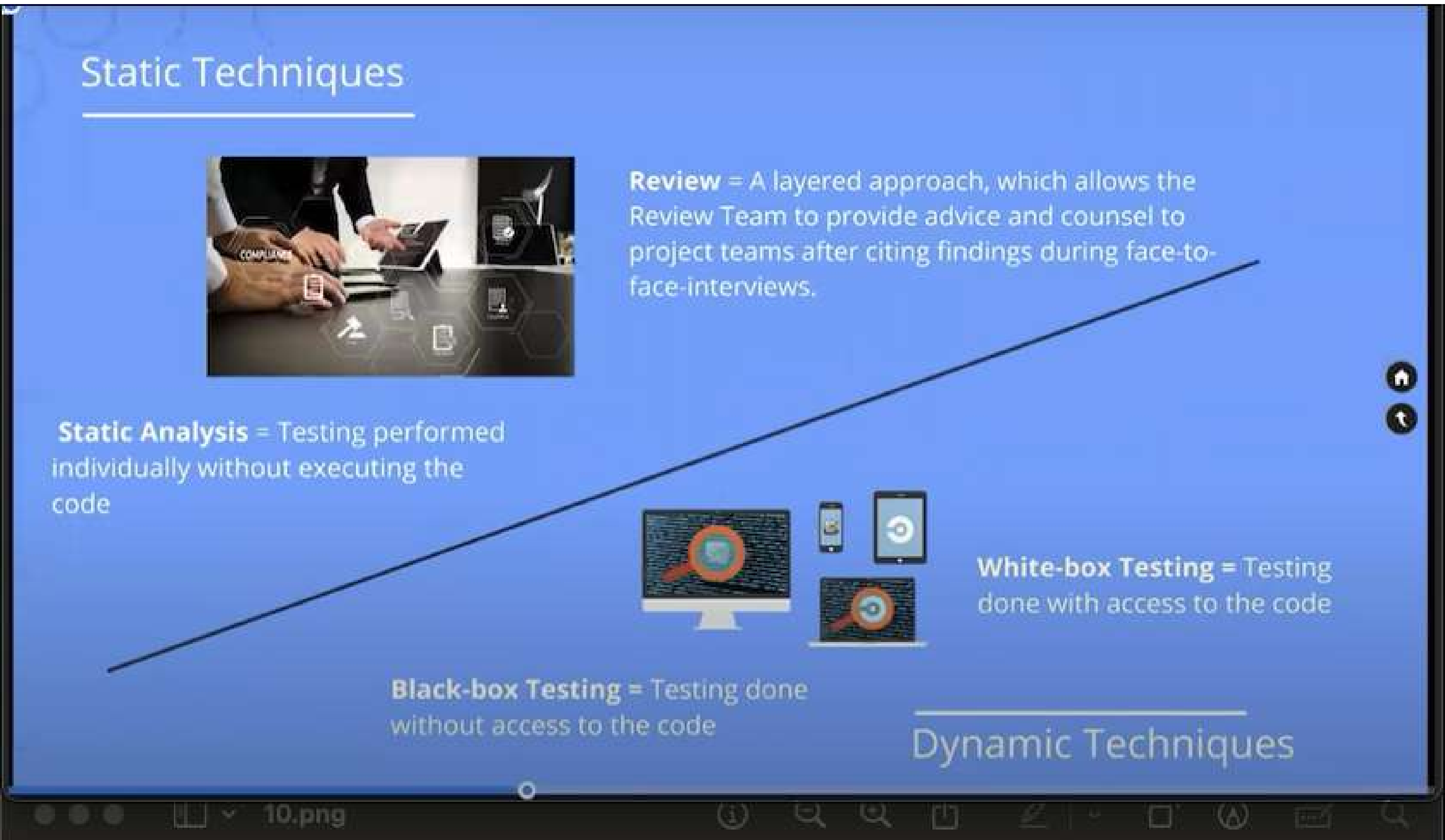


How to test?

We have multiple ways to perform the testing, but every time we use one or more testing techniques, or perform risk based testing.

Test Design Techniques

Risk Based Testing



Static Techniques

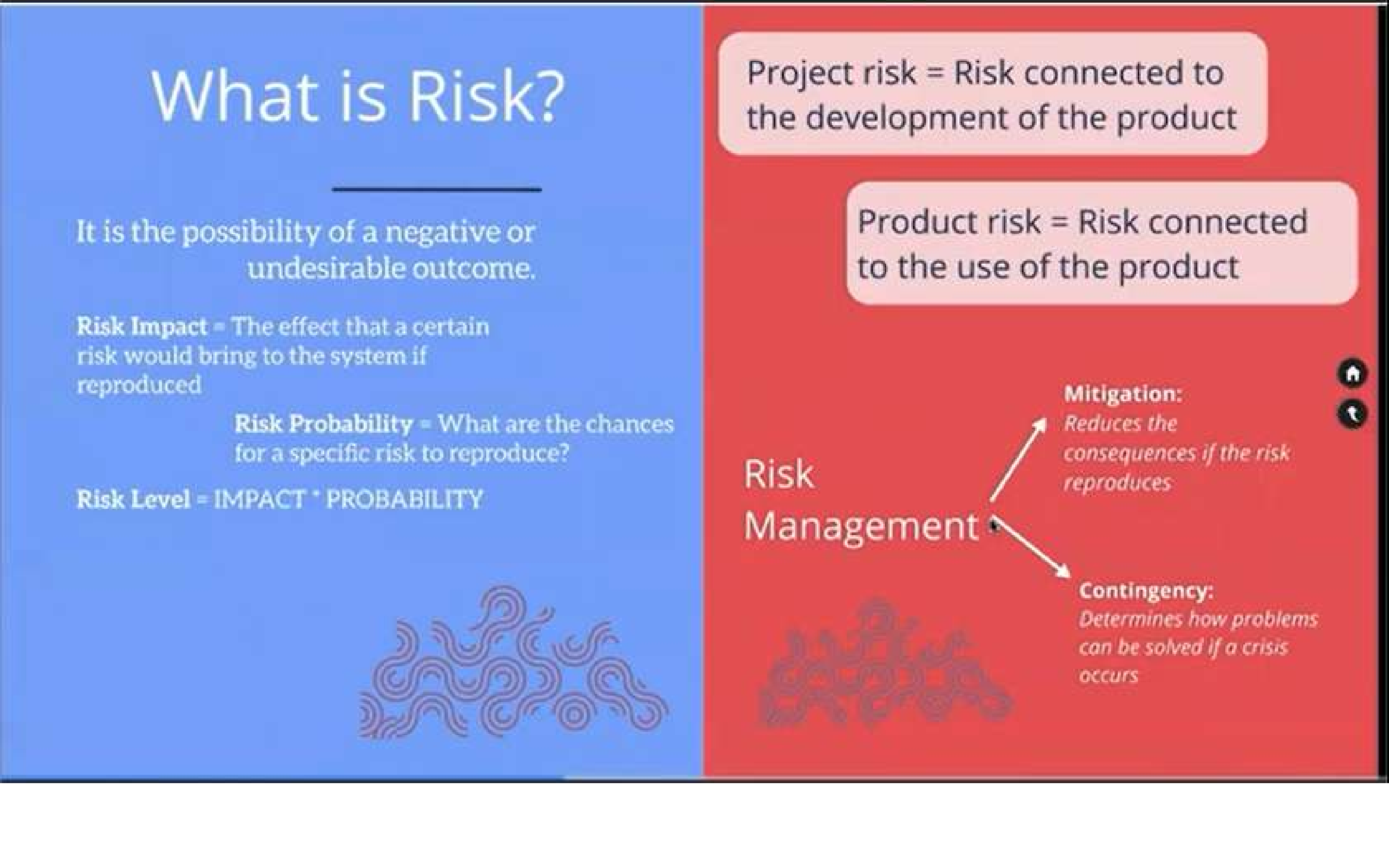
Review = A layered approach, which allows the Review Team to provide advice and counsel to project teams after citing findings during face-to-face-interviews.

Static Analysis = Testing performed individually without executing the code

White-box Testing = Testing done with access to the code

Black-box Testing = Testing done without access to the code

Dynamic Techniques



What is Risk?

It is the possibility of a negative or undesirable outcome.

Risk Impact = The effect that a certain risk would bring to the system if reproduced

Risk Probability = What are the chances for a specific risk to reproduce?

Risk Level = IMPACT * PROBABILITY

Project risk = Risk connected to the development of the product

Product risk = Risk connected to the use of the product

Risk Management

Mitigation: *Reduces the consequences if the risk reproduces*

Contingency: *Determines how problems can be solved if a crisis occurs*

Testing Tools

Testing process will require the use of a series of tools that will make the life of a tester easier and the process more effective and efficient.

Testing Classes

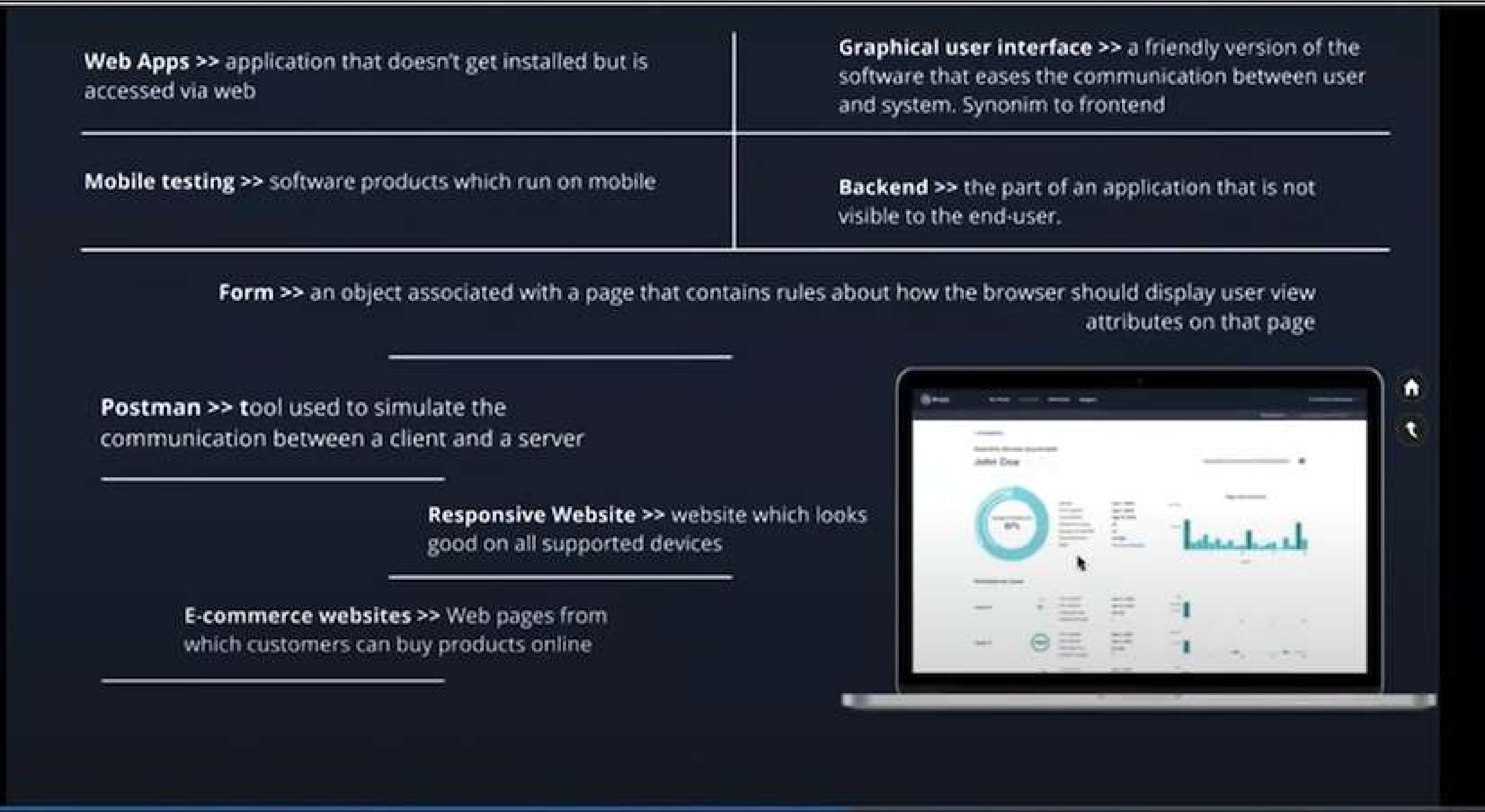
Testing tools are mostly represented by the test management tools, but they include also tools that are not necessarily related to test cases like postman, swagger, mysql workbench.

TestRail

Jira

Model: Practitest

TestLink



Web Apps >> application that doesn't get installed but is accessed via web

Graphical user interface >> a friendly version of the software that eases the communication between user and system. Synonim to frontend

Mobile testing >> software products which run on mobile

Backend >> the part of an application that is not visible to the end-user.

Form >> an object associated with a page that contains rules about how the browser should display user view attributes on that page

Postman >> tool used to simulate the communication between a client and a server

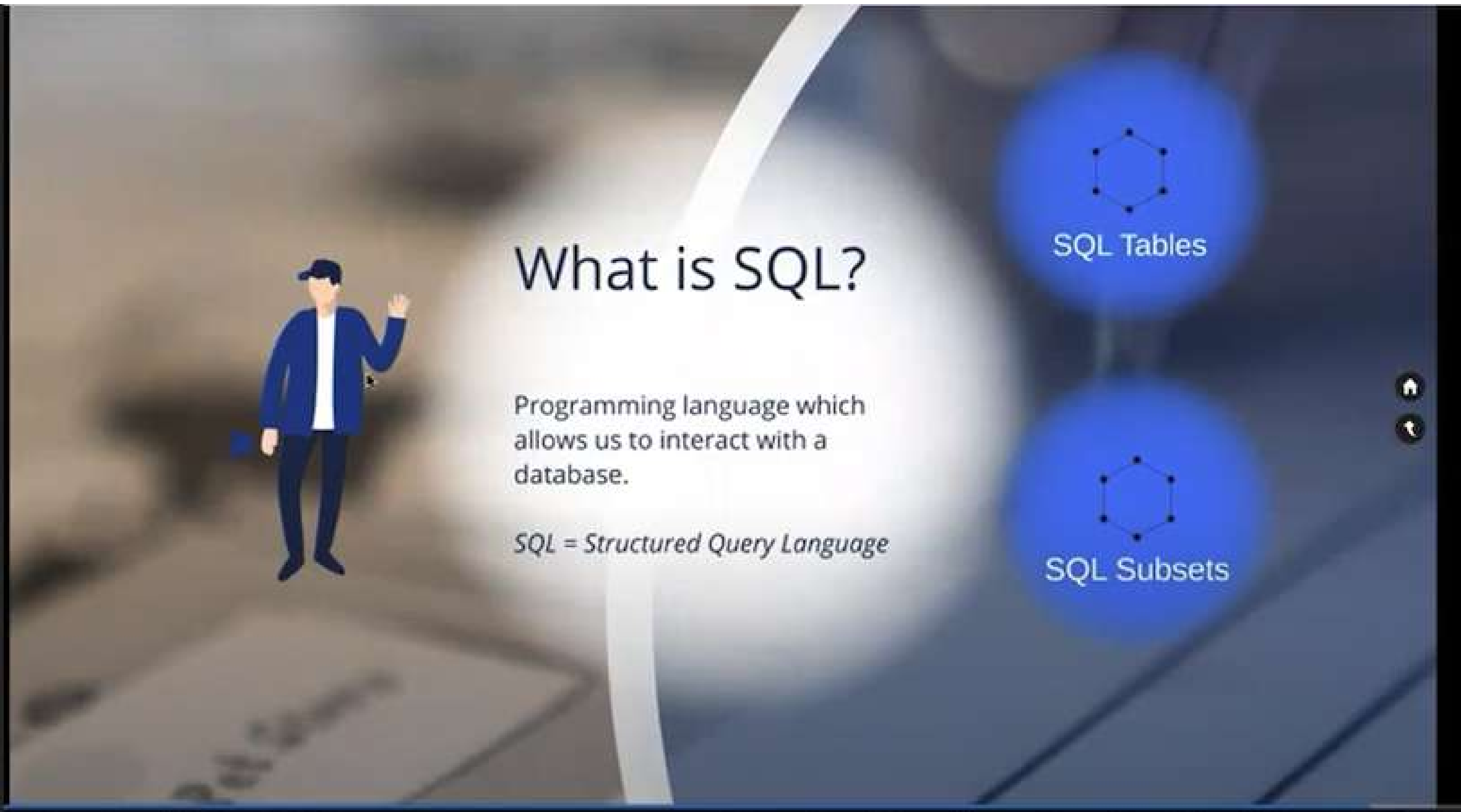
Responsive Website >> website which looks good on all supported devices

E-commerce websites >> Web pages from which customers can buy products online

SQL

Preparation For ISTQB

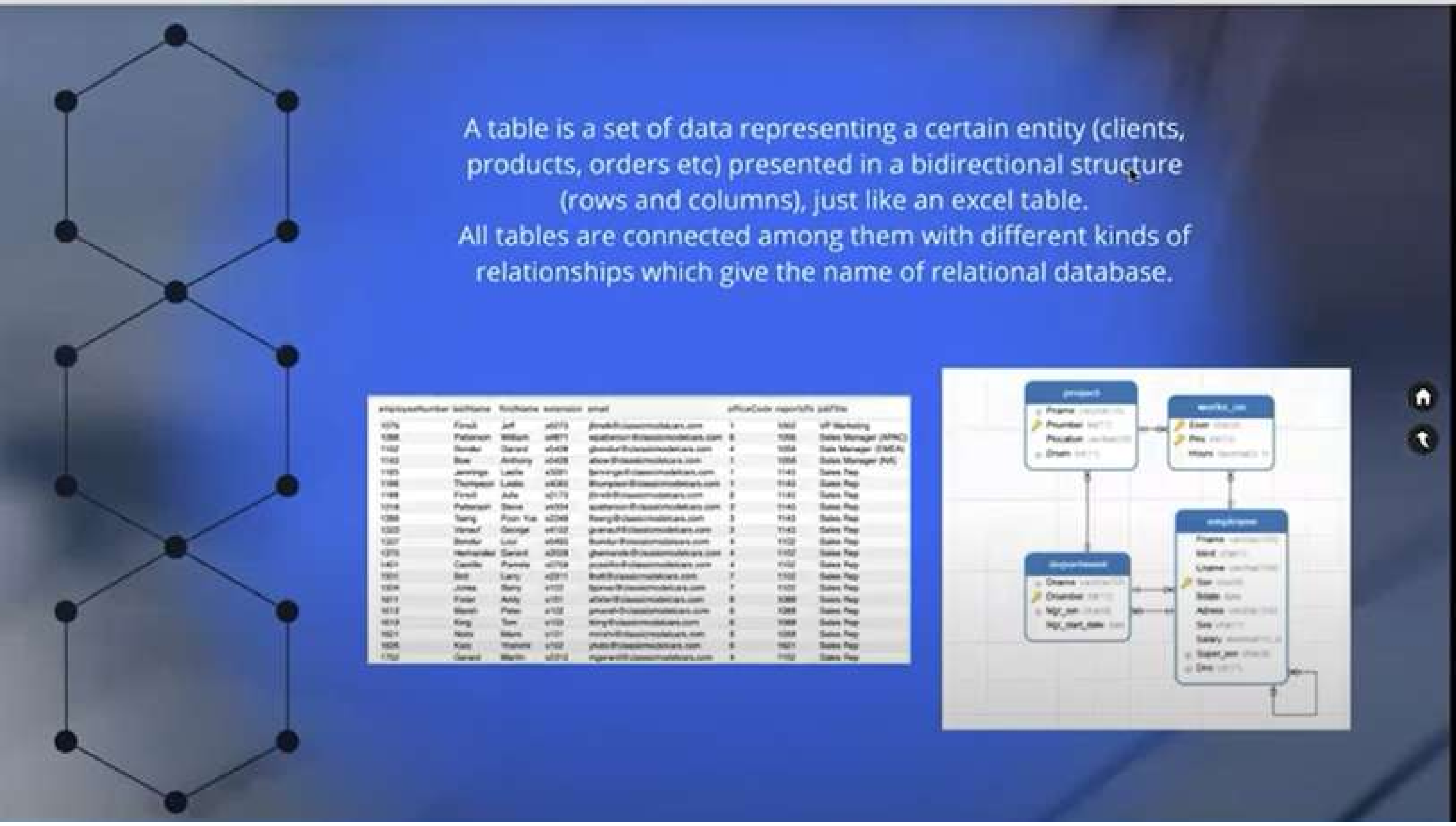
Final Project



What is SQL?

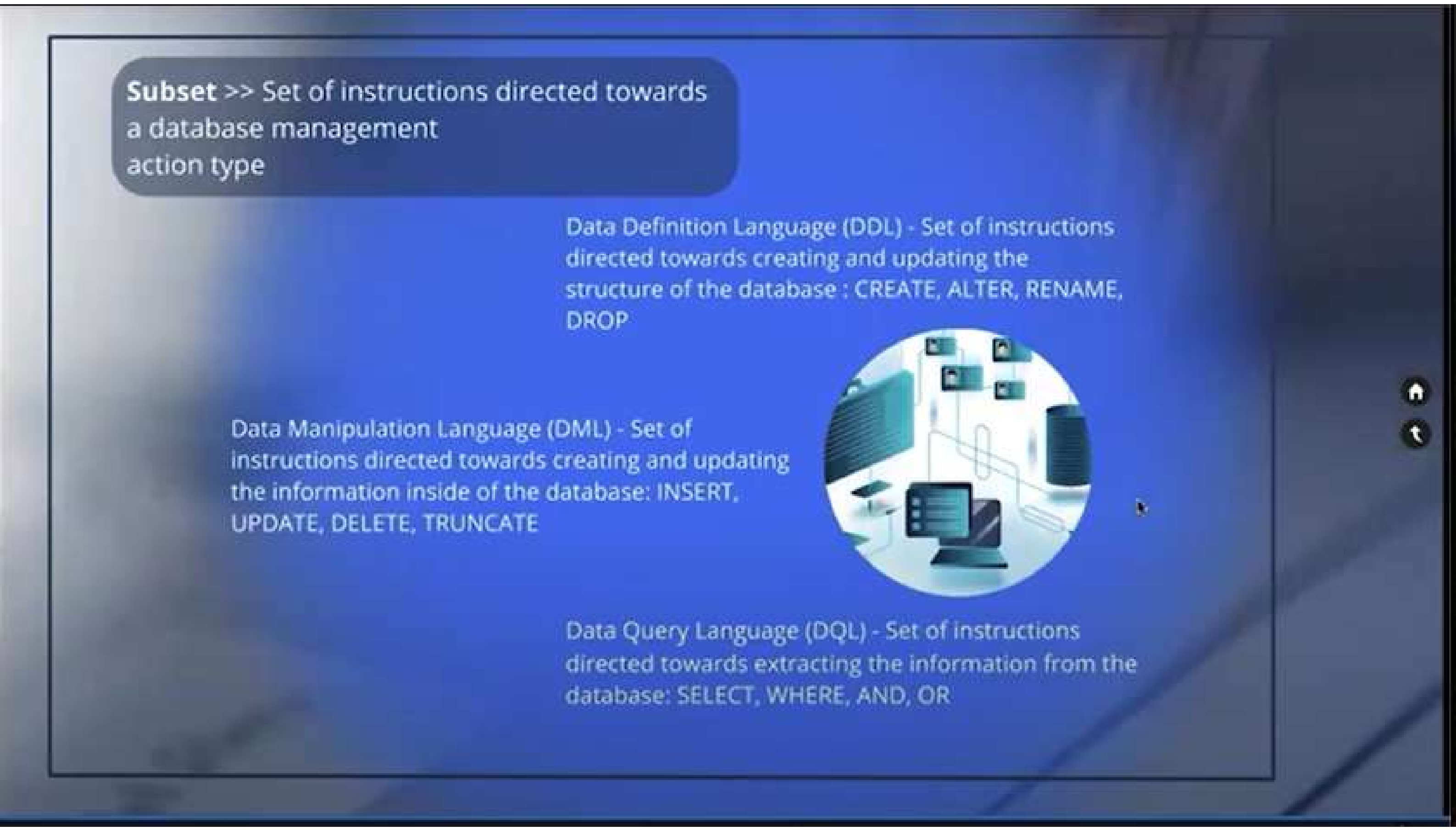
Programming language which allows us to interact with a database.

SQL = Structured Query Language



A table is a set of data representing a certain entity (clients, products, orders etc) presented in a bidirectional structure (rows and columns), just like an excel table.
All tables are connected among them with different kinds of relationships which give the name of relational database.

employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle
1076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com	1	1002	VP Marketing
1088	Patterson	William	x4871	wpatterson@classicmodelcars.com	6	1056	Sales Manager (APAC)
1102	Bondur	Gerard	x5408	gbondur@classicmodelcars.com	4	1056	Sale Manager (EMEA)
1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)
1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep
1166	Thompson	Leslie	x4065	lthompson@classicmodelcars.com	1	1143	Sales Rep
1188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep
1216	Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales Rep
1286	Tseng	Foon Yue	x2248	ftseng@classicmodelcars.com	3	1143	Sales Rep
1323	Vanauf	George	x4102	gvanauf@classicmodelcars.com	3	1143	Sales Rep
1337	Bondur	Loui	x6493	lbondur@classicmodelcars.com	4	1102	Sales Rep
1370	Hernandez	Gerard	x2028	ghernande@classicmodelcars.com	4	1102	Sales Rep
1401	Castillo	Pamela	x2759	pcastillo@classicmodelcars.com	4	1102	Sales Rep
1501	Bott	Larry	x2311	lbott@classicmodelcars.com	7	1102	Sales Rep
1504	Jones	Barry	x102	bjones@classicmodelcars.com	7	1102	Sales Rep
1611	Fixter	Andy	x101	afixter@classicmodelcars.com	6	1088	Sales Rep
1612	Marsh	Peter	x102	pmarsh@classicmodelcars.com	6	1088	Sales Rep
1619	King	Tom	x103	tking@classicmodelcars.com	6	1088	Sales Rep
1621	Nishi	Mami	x101	mnishi@classicmodelcars.com	5	1056	Sales Rep
1625	Kato	Yoshimi	x102	ykato@classicmodelcars.com	5	1621	Sales Rep
1702	Gerard	Martin	x2312	mgerard@classicmodelcars.com	4	1102	Sales Rep



Subset >> Set of instructions directed towards
a database management
action type

Data Definition Language (DDL) - Set of instructions directed towards creating and updating the structure of the database : CREATE, ALTER, RENAME, DROP

Data Manipulation Language (DML) - Set of instructions directed towards creating and updating the information inside of the database: INSERT, UPDATE, DELETE, TRUNCATE

Data Query Language (DQL) - Set of instructions directed towards extracting the information from the database: SELECT, WHERE, AND, OR