

Session 2

Evaluation and Testing

By the end of this session participants will have:

- Examined the importance of testing in the software development cycle.
- Experienced real-life testing scenarios.
- Developed test plans and test cases for a variety of situations.
- Investigated and experienced test-driven development.

LCCS Learning Outcomes

Evaluation and Testing:

L.O.s 2.19, 2.20, 2.21, 2.22.

But also

Design and Developing

L.O. 1.19

Computational Thinking

L.O.s 1.1 - 1.19

Computers and Society

L.O.s 1.11

Computer Science in Practice

L.O.s 3.2 – 3.14

Talking about Testing



Software Testing ?

In groups of three, discuss the importance of software testing?

Each group to come up with three to five points, which are collated and saved.



Debrief From Each Group

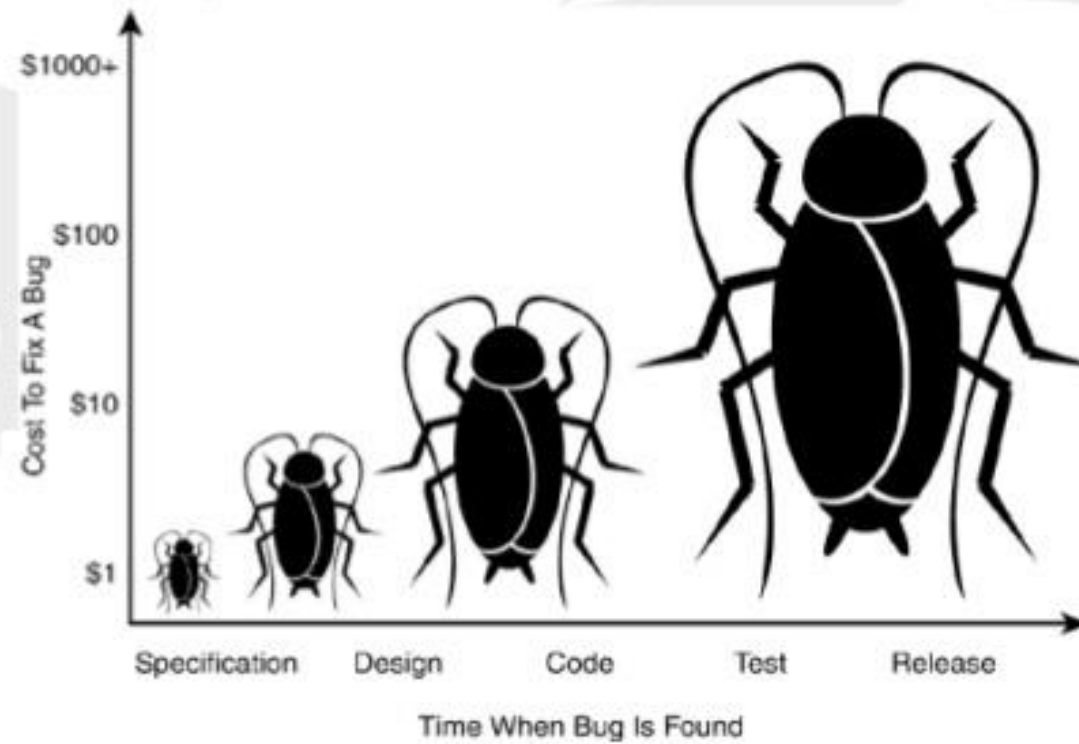
Software testing: What were your thoughts



Some interesting scenarios

- Windows XP – how many bugs?
- MS Word 97 – French version – what happened?
- Boeing
- Costs
- Norman Whiteside

Bug fixing cost



⏸

Pause

00:00:00

⊞

Select Area

🎤

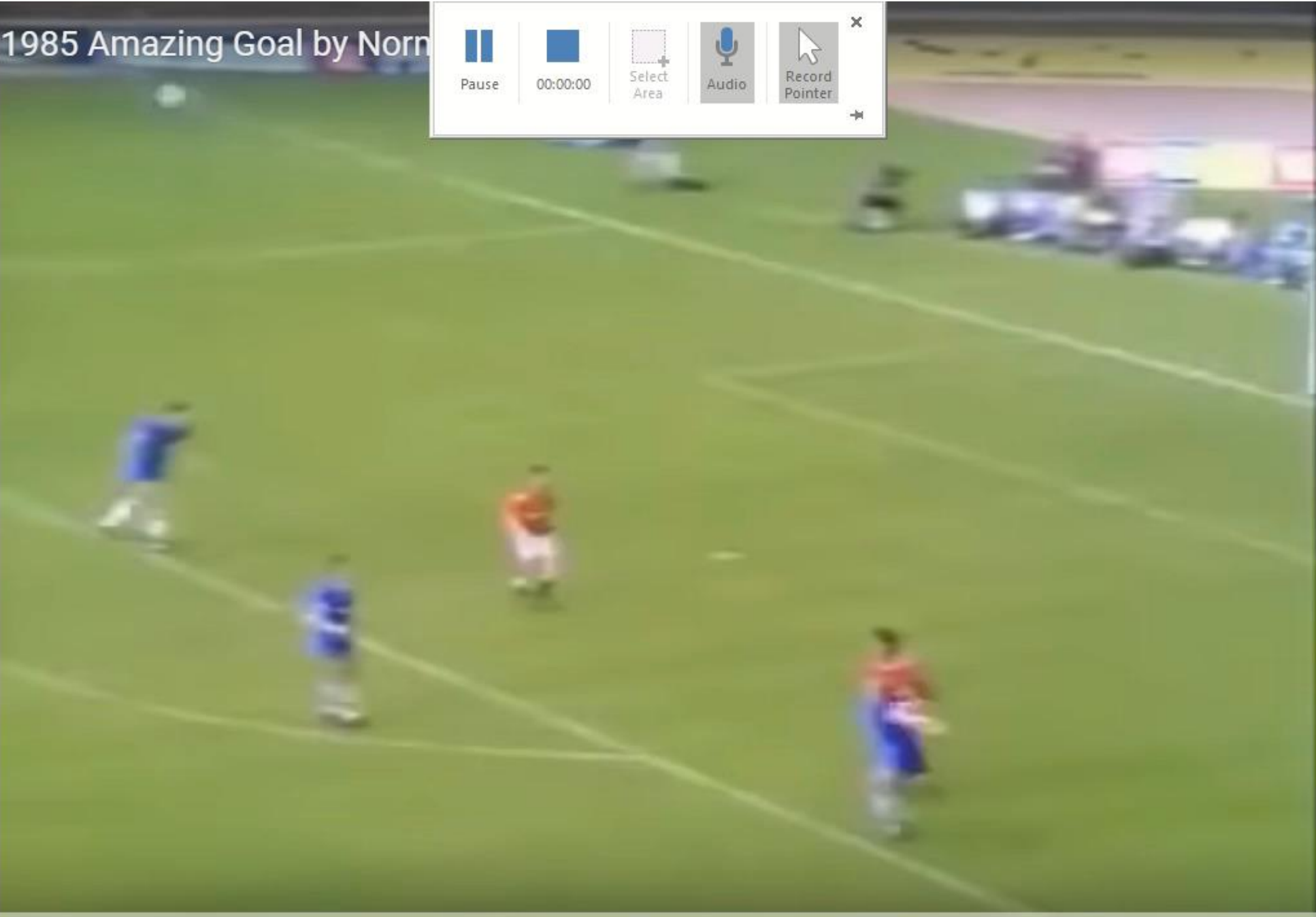
Audio

🖱

Record Pointer

✕

↕



Play (k)

▶

⏮

🔊

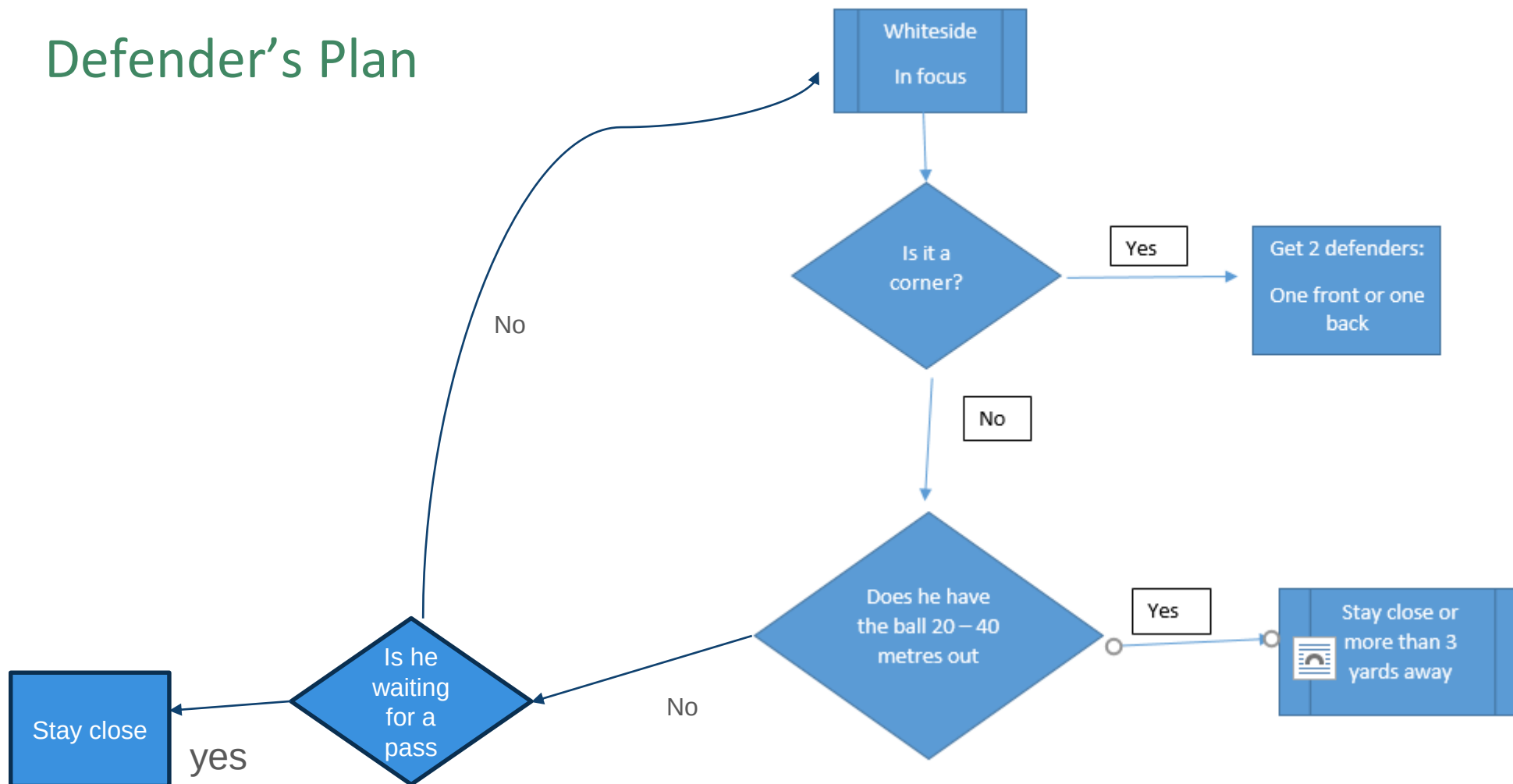
0:00 / 1:45

Scroll for details

⚙

⛶

Defender's Plan

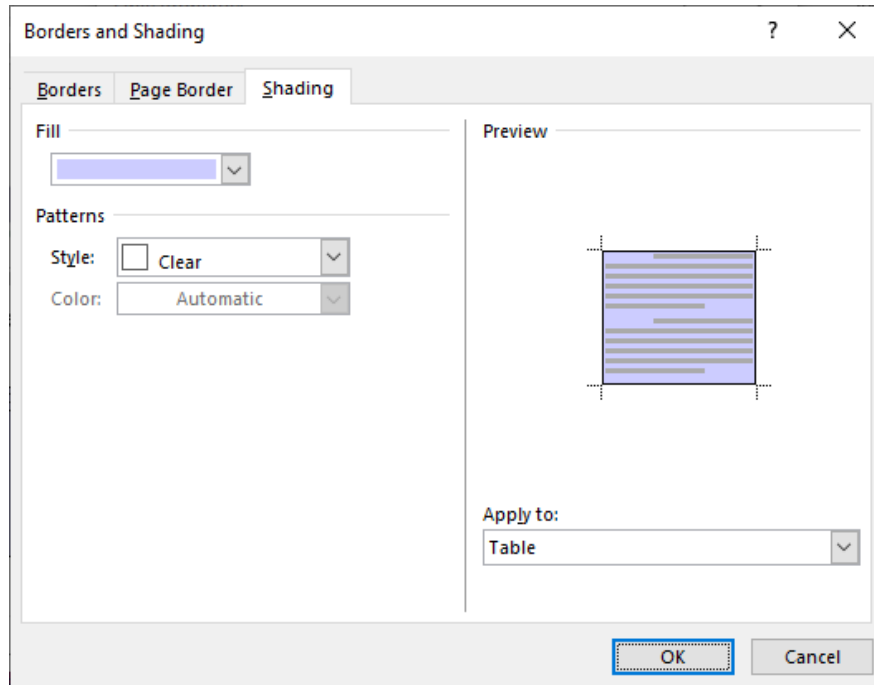


Let's test 1

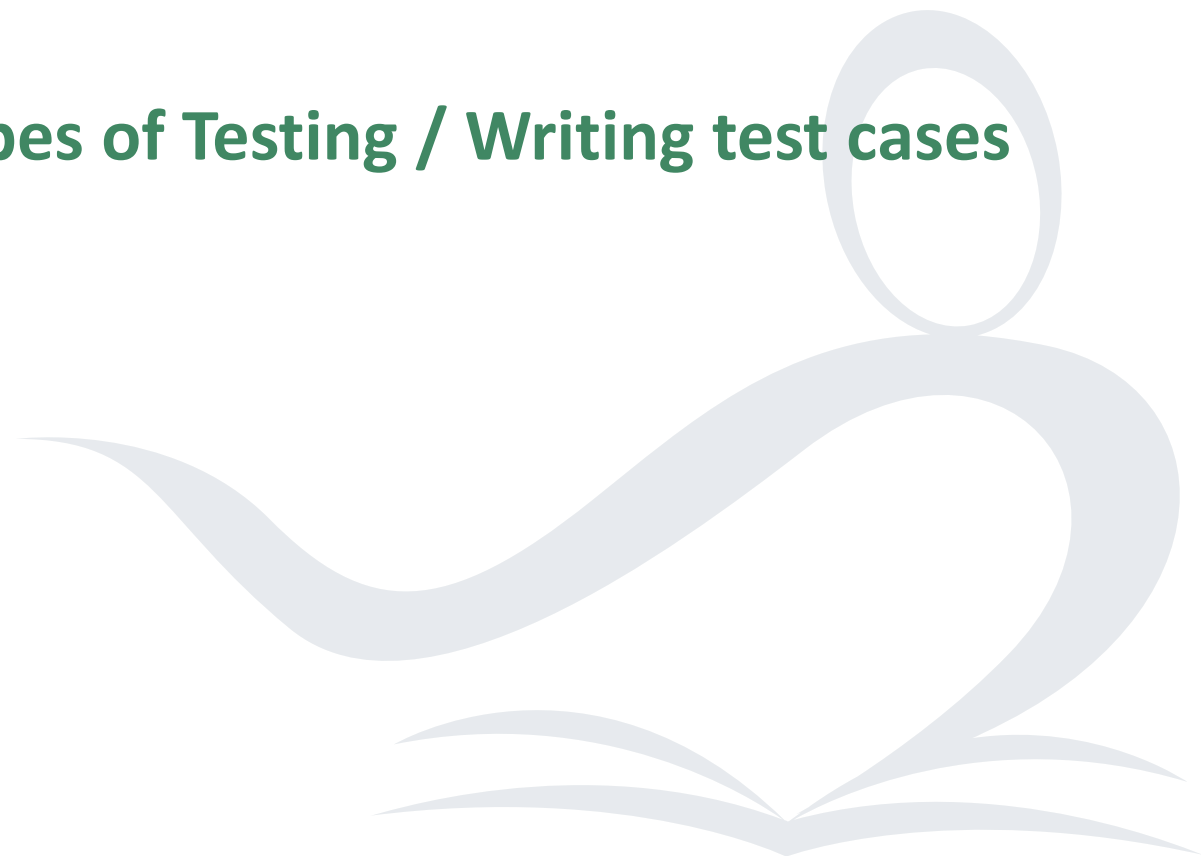


Let's Test

Plan and implement a test plan for a real-life test scenario:



Types of Testing / Writing test cases



Teachers research and report:

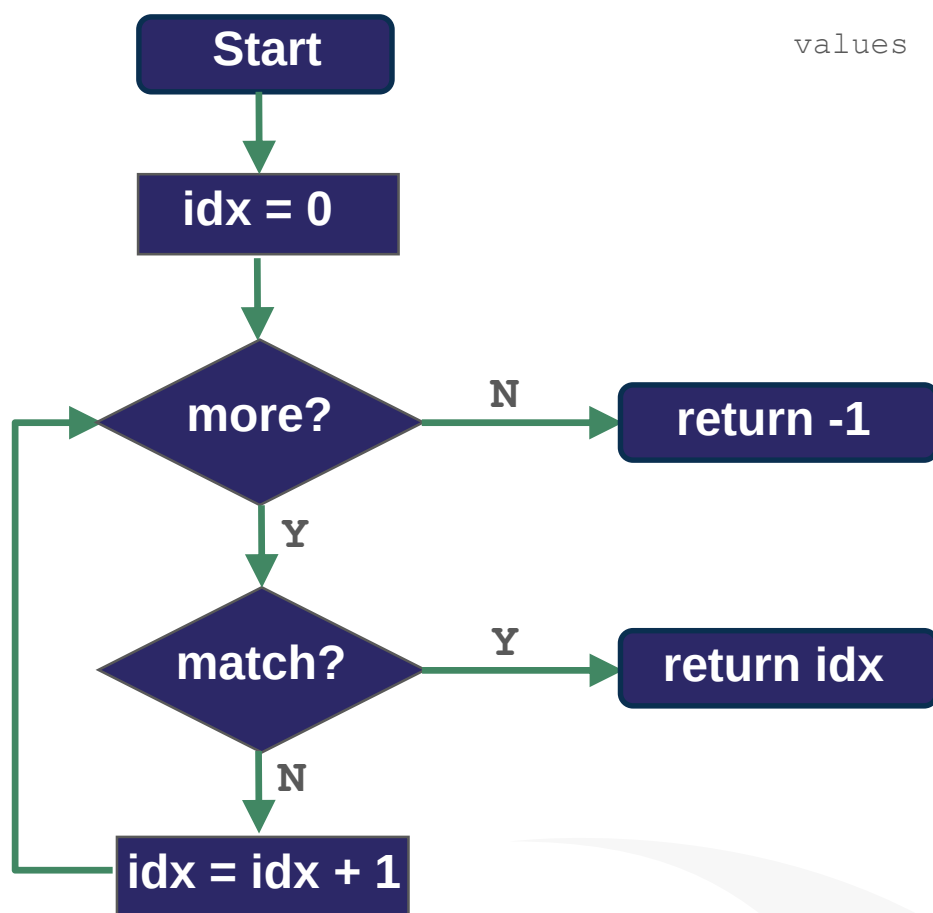
1. Functional v non-functional testing
2. Unit vs Integration (System Testing)
3. White v black box testing.
4. Designing a test plan.
5. Writing test cases.
6. Test-driven development
7. Testing in Waterfall / Agile / V-shaped model development

Let's test 2



Linear Search

idx	0	1	2	3	4	5	6
values	21	17	-1	26	22	-5	24



Testing Cycle



Testing Cycle:

Requirements Analysis

Test

Planning

Development

Execution

Reporting

Test-driven development



Test-driven development:

1. Run a test
2. Run all tests and see if new test fails
3. Write the code
4. Run tests
5. Refractor code
6. Repeat

KISS: Keep it simple stupid.

YAGNI: You aren't gonna need it.

“Fake it till you make it” (Kent Blake): Write only code needed to pass tests.

TDD / Agile / Waterfall / V-model Development:

Does TDD fit in the Agile framework?

Is TDD the same as Agile development?

Is Waterfall development relevant?

What is the V-model in testing?

How does the role of testing change, according to these methodologies?

Let's test 3



Submit

www.cupannua.com