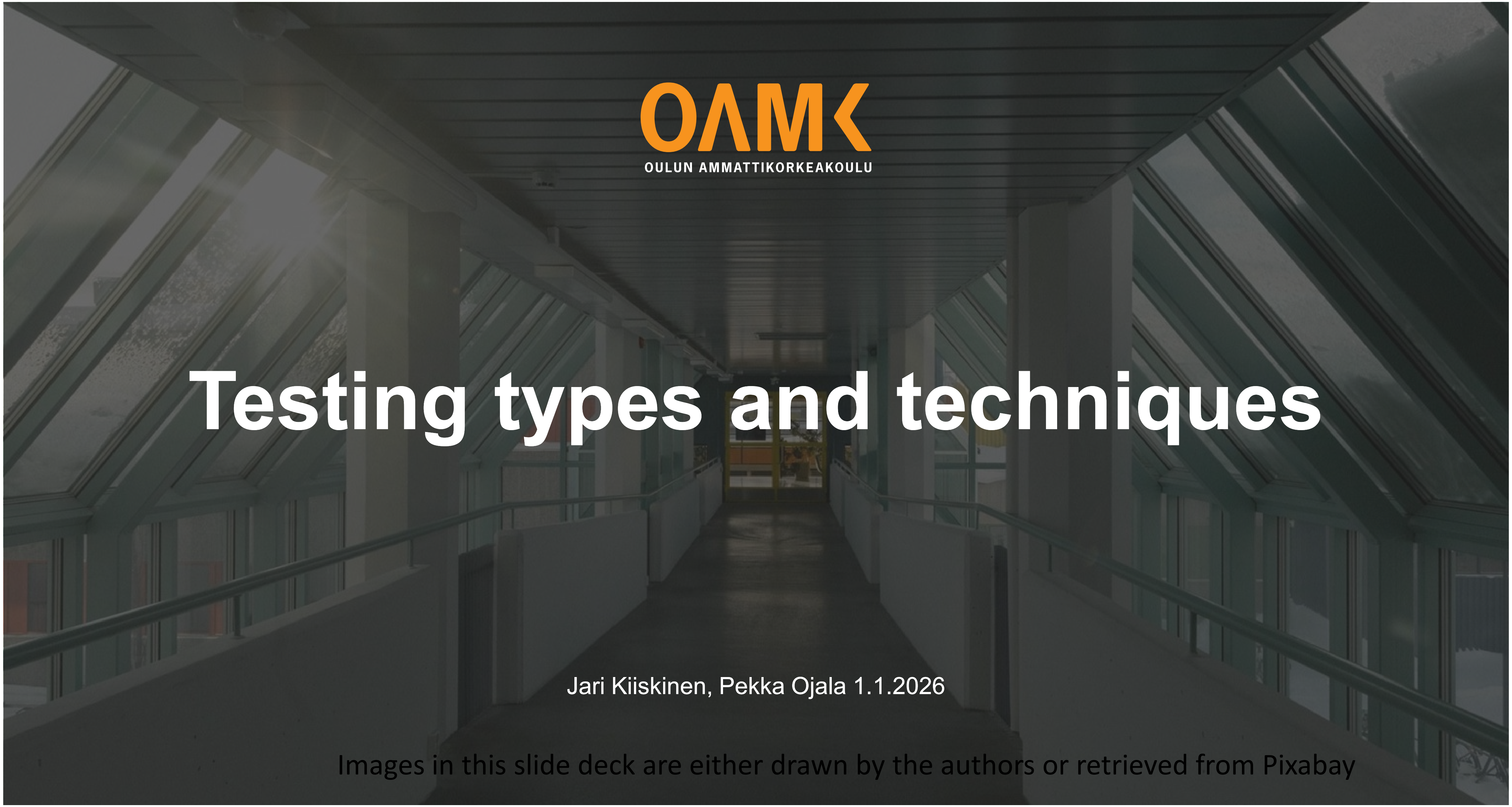


OAMK

OULUN AMMATTIKORKEAKOULU

Testing types and techniques

Jari Kiiskinen, Pekka Ojala 1.1.2026

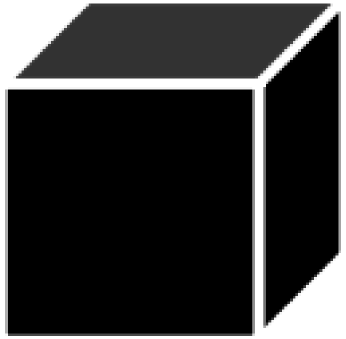
Images in this slide deck are either drawn by the authors or retrieved from Pixabay

Contents

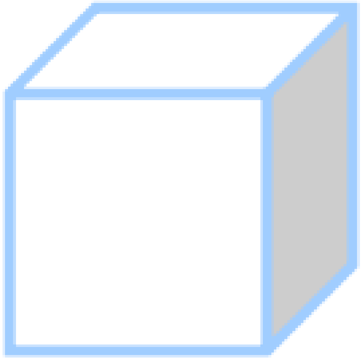
- Black-box and white-box testing
- Functional testing
- Non-functional testing
- Regression testing
- Confirmation testing
- Equivalence partitioning
- Boundary value analysis
- Decision table
- Risk-based testing
- Exploratory testing

Black-box and White-box Testing

Two best-known testing strategies, this slide deck focuses on black-box testing



Black Box Testing examines program behavior from external perspective - inputs and outputs. Anyone with domain knowledge can perform it. Types of testing: **functional, non-functional, regression**. Example: Testing online store product search.



White Box Testing requires understanding the internal operation. Only those familiar with architecture or code can do it. Testing type can be **integration** or **unit tests** both requiring different knowledge pool for setup and relevant inputs and outputs. Example: Automated tests for components within a web store.

After refactoring, are tests still valid?
Especially valid difference in security testing

Black-box and White-box Testing

White-box testing allows running large numbers of specific tests faster. Black-box testing results in fewer unnecessary tests.

o Example: Suppose a programmer has made a rounding error in a code component. Instead of rounding up first, decimals are simply truncated

o If a tester tests this program from a customer's perspective, they may not notice the rounding, or the rounding error may not appear with the inputs used

o To reveal the error with black-box testing, more inputs are needed (or white-box testing that compares large input and output quantities to reference values)

Functional Testing

- A type of black-box testing where software is **tested against its specified requirements** from a functionality perspective
- Examples of functional testing for an online store:
  - Registration
  - Login
  - Product search
  - Purchase transaction
  - etc...

Non-functional Testing

- A type of black-box testing where software is tested against non-functional requirements
- Examples of non-functional requirements and their testing for an online store:
  - Usability (e.g., how easy is registration in the store)
  - Security (e.g., are there security vulnerabilities in the store)
  - Performance (e.g., how fast do product searches work)
  - Compatibility (e.g., are there differences in the store's use between browsers)

Regression Testing

- If changes made to software result in new bugs or other problems, this is called a Side Effect
- The goal of regression testing is to ensure that fixes made to software don't cause new problems in functionalities that have already been tested and verified as working
- Regression testing should be automated because the number of tests can be very large and grows as the software size increases

Confirmation Testing

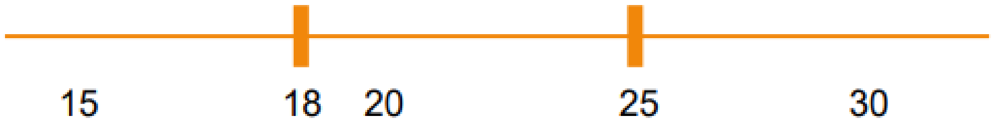
- The concept of regression testing is widely used in software development
- Confirmation Testing is a less common concept than regression testing. If it is to be distinguished from regression testing, it means the following:
  - Verify that fixed bugs **are actually fixed**
  - Repeat tests that don't pass
  - May reveal the need to write new tests

Equivalence Partitioning

- Equivalence Partitioning is one of the best-known black-box testing methods
- Test data is divided into equivalence classes (equivalence partitions) and testing is done based on test cases representing the classes
- If a test case from a certain class passes the test, it can be assumed that all other test cases from the same class would also pass, and vice versa
- Makes testing more efficient because redundant test cases (representatives of the same class) don't need to be unnecessarily repeated

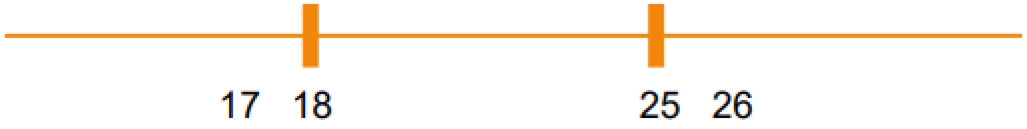
Equivalence Partitioning

- Example: Suppose an input field in software must accept an age between 18-25. Different equivalence classes:
  - Valid values (e.g., 20)
  - Values too small (e.g., 15)
  - Values too large (e.g., 30)
  - Invalid inputs (e.g., negative number, decimal number, or non-numeric input)
  - Missing input



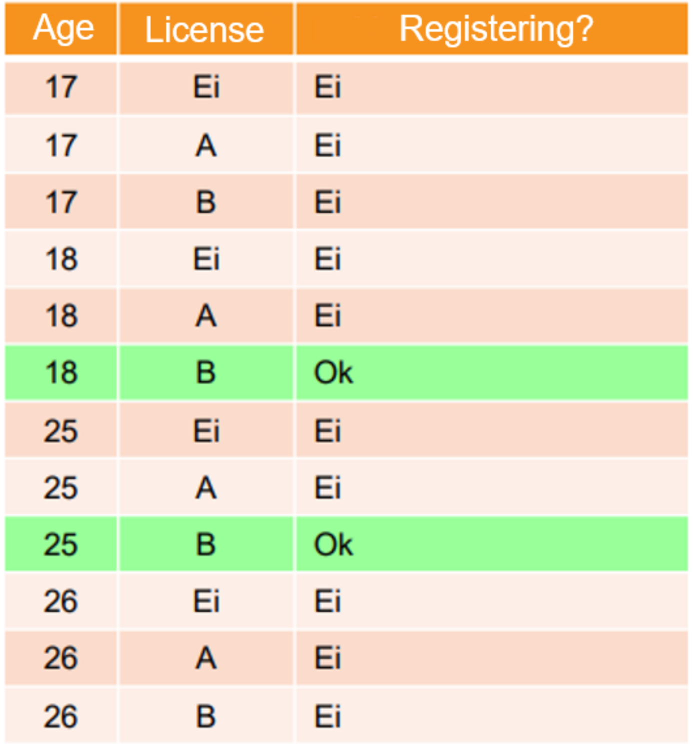
Boundary Value Analysis

- Boundary Value Analysis is another example of a common black-box testing method
- Testing is done with values at the boundaries of equivalence classes. For the previous slide's example (age must be between 18-25), minimal boundary value analysis would be done with the following test cases:
  - 17 (one below the accepted lower limit)
  - 18 (accepted lower limit)
  - 25 (accepted upper limit)
  - 26 (one above the accepted upper limit)



Decision Table

- A Decision Table is a third example of a black-box testing method. It can be used to describe the effects of different functions or actions in various combinations
- Example with boundary values: Registration in software succeeds if the registrant's age is between 18-25 and they have at least a Class B driver's license
- Test all combinations for 100% coverage or less?
- N/A (not applicable) describes an impossible parameter, - describes an irrelevant parameter



Age	License	Registering?
17	Ei	Ei
17	A	Ei
17	B	Ei
18	Ei	Ei
18	A	Ei
18	B	Ok
25	Ei	Ei
25	A	Ei
25	B	Ok
26	Ei	Ei
26	A	Ei
26	B	Ei

Ikäpoikkeusluvalla ajokortin voi saada 17-vuotiaana
https://www.traficom.fi/fi/asioi-kanssamme/hae-ikapoikkeuslupaa-ajokorttia-varten

Risk-based Testing

- Risk Based Testing is one of the Agile Testing strategies, where risks related to software are mapped and test cases are selected accordingly
- Software testing can be directed where it is most needed (so-called prioritized testing)
- For each feature, the following can be analyzed:
  - How important is the feature to software users?
  - How often is the feature used?
  - How much damage does the feature's failure cause?

Risk-based Testing

Risk-based testing also aims to assess implementation uncertainty by considering the following:

o How new is the implementation technology?

o How dependent is the implementation on other components?

o Has the function been tested previously, e.g., in other products?

o What is the skill level of the developers?

o Other factors such as code history (how many authors) and code complexity, how many bugs it has had previously

Smoke Testing

When software is available in some way, a pre-test can be performed to verify whether the delivery should be taken forward or seriously tested:

o Does the program do anything?

o Is the program at all stable?

o Do the most common user interfaces open?

o Does the code compile/link, do unit tests pass, and can a release even be made from it?

(or does smoke come out of the device when starting)
Many things in this course applies to SUT/DUT/HW/SW/Mech…
HW can be smoking and it's very relevant to smoke test new HW (**lot** of time saved!)

Exploratory Testing

- Exploratory Testing is another example of agile testing. The starting point for testing is that you continuously learn more about the test object during testing
  - As you learn more about the test object, you can create meaningful test cases that can be repeated later
- Exploratory testing should not be understood as a strategy where testing doesn't need to be planned and documented, because it's important that test cases created during testing are maintained
  - The current tester is not the only one who needs test cases

Sources and Further Reading

- Atlassian. Exploratory Testing. https://www.atlassian.com/continuousdelivery/software-testing/exploratory-testing
- GURU99. What is BLACK Box Testing? Techniques, Types & Example. https://www.guru99.com/black-box-testing.html
- GURU99. Risk Based Testing. https://www.guru99.com/risk-basedtesting.html
- Imperva. Black Box Testing. https://www.imperva.com/learn/applicationsecurity/black-box-testing/
- VALA. What is regression testing. https://www.valagroup.com/fi/blogi/mita-onregressiotestaus/