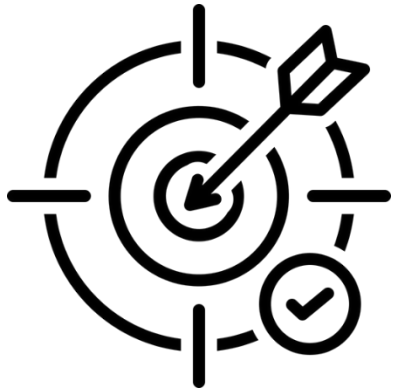
Test Generation

Web Programming and Testing

Mario Simaremare, S.Kom., M.Sc.
Program Studi Sarjana Sistem Informasi
Institut Teknologi Del

Objectives

- The objective of this session is the following:
  - The students are able to generate test cases based on the requirements.



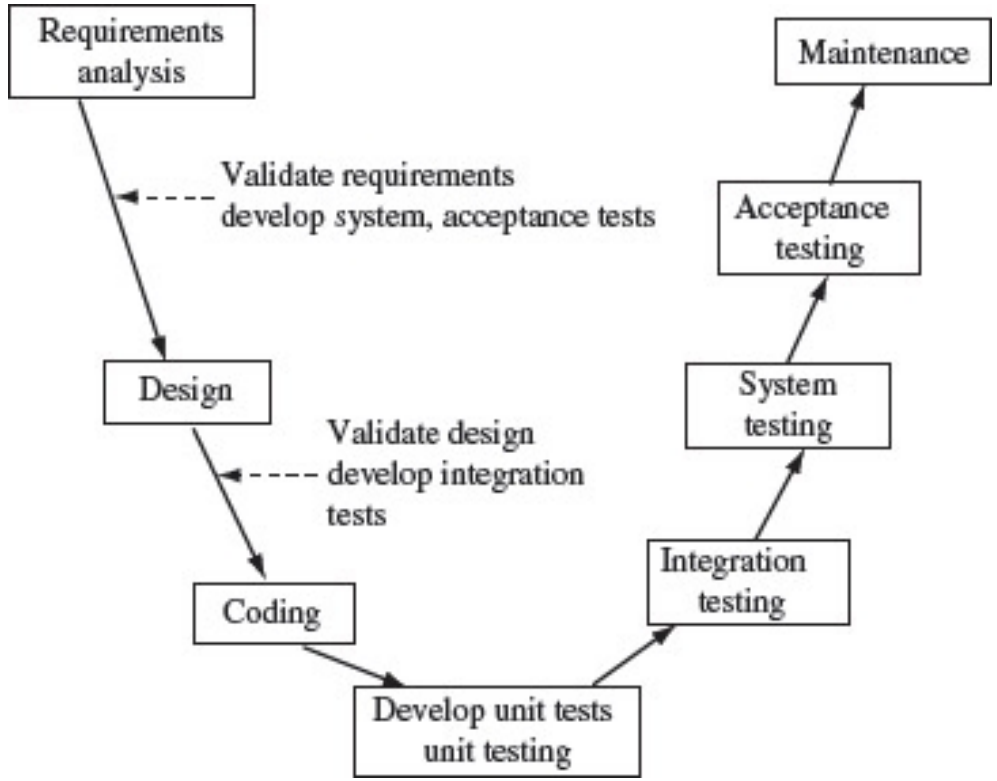
Outlines

1. Types of testing.
2. Test Generation.
3. Test Generation: Approach.
4. Get our hands dirty.

Types of testing

Testing in SDLC: V-Model

- Each step has its own testing.
  - Formal & rigorous.
  - High quality solutions.
- There are others:
  - Waterfall, spiral, agile.



Static vs. Dynamic Testing

Static Testing

- Does not require code execution.
- High labor (expensive).
- Might reveal inefficient routines.
- Ends in a high quality codes.

- **Techniques**:
  Peer review, code complexity measurement, etc.

Dynamic Testing

- Requires code executions.
- Cheaper, since it can be designed to be automated, randomized, …
- Checks the effectiveness of the solution being tested.

- **Techniques**:
  Black-box and white-box testing, model-based specification, etc.

Our Focus

- In this course, we put our focus on:
  - Achieving a correct solution (effectiveness).
  - Dynamic testing → Black-box testing.
    - Source of test generation: requirements.

- Black-box testing is an objective-based testing.
  - What this "thing" should do? Let's test it.
    - Input-output configurations are based on the requirements.
    - Detailed, clear, and unambiguous requirements are extremely helpful.

Test Generation

Some Terminologies

- Test case (TC): is a set of input and output.
  - A testing is labelled as **passed** when the actual execution with the given input configuration produces output that matches the expected output. Otherwise, it's labelled as **failed**.
  - No false positive or false negative result.
- Test suite (TS): is a set of TCs.
  - The TCs inside a TS are related one to the another.
- Test plan: describe the goal, the object, the procedures, the TCs, of a testing.

Test Generation

- Very important!
  - Enough TCs increase our confidence.
  - When is enough?
  - How to generate the required TCs?

- Key success factor: **requirements**.

Test Generation

- Correct software → when it does exactly what it is expected to do without any failure. Perfecto!

- How many TCs are required?
  - Depends on the requirment's complexity.
  - When every possible input-output configurations are tested flawlessly.
    - Sometimes it is possible, sometimes it is **not**.
    - When it is not possible:
      - Could the testers find some configurations that represent the while possibility? **Yes**.

Test Generation: Approach

Test Generation Approach

- Randomly generated (Random Testing).
  - The inputs are generated randomly (naïve approach).
- Boundary value analysis → our focus.
  - Testing stuffs around the boundaries.
- Equivalence partitioning.
  - When there is a set of classification, or category or partition dividing the input domain.
    - E.g. kids, teenager, adult, etc.
- … and many more.

Boundary Value Analysis

- "*Bugs lurk in corners and congregate at boundaries*"– Boris Beizer
- The fact is most bugs were simply faulty boundary values.
- Rules:
  - Say there is an input variable, x.
  - The domain value is $\{x_{min}, \ldots, x_{max}\}$

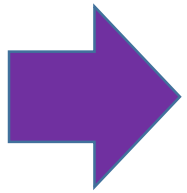
$$x_{min} - 1,\ x_{min},\ x_{min} + 1$$
$$x_{max} - 1,\ x_{max},\ x_{max} + 1$$
$$(x_{min} + x_{max})/2$$

Boundary Value Analysis

- Example, say a variable hh is aimed to represent a 24-hour.
  - $hh_{min} = 0$
  - $hh_{max} = 23$

- Our test data would be:

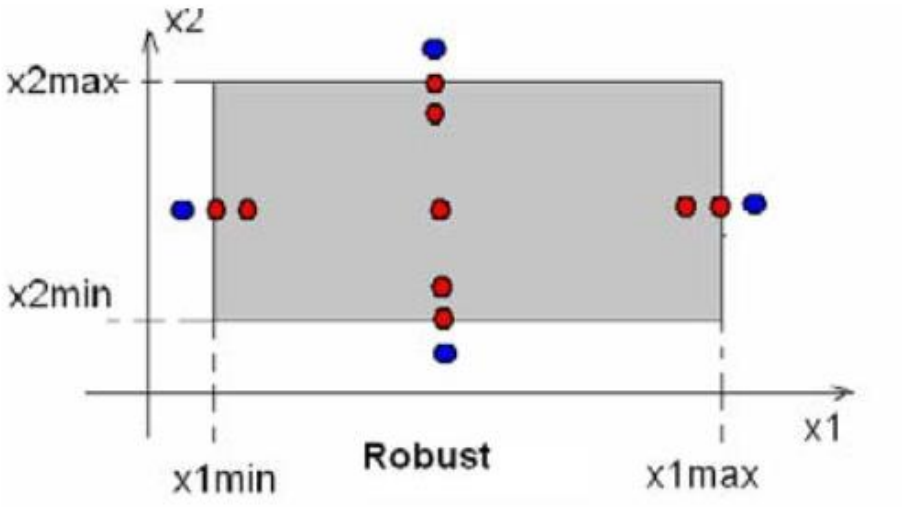
$hh_{min} - 1$, hh_{min}, $hh_{min} + 1$
$hh_{max} - 1$, hh_{max}, $hh_{max} + 1$
$(hh_{min} + hh_{max})/2$

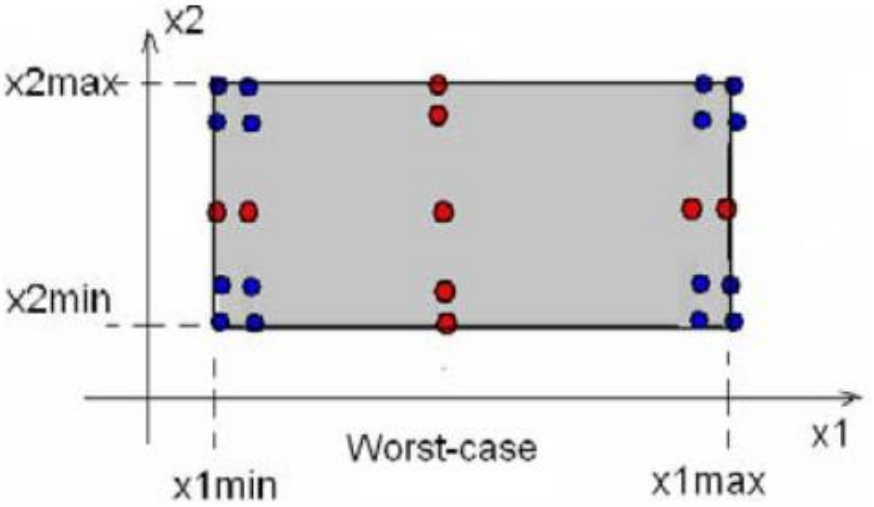


-1, 0, 1
22, 23, 24
12

Boundary Value Analysis

- Rules for multiple variables.
  - Say there are input variables, x1 and x2.
  - The domain values are {$x1_{min}$, …, $x1_{max}$} and {$x2_{min}$, …, $x2_{max}$}.





Boundary Value Analysis

- Challenge:
  - There are three variables, `hh`, `mm`, `ss`.
  - `hh` represents a 24-hours;
  - `mm` represents a 60-minutes; and
  - `ss` represents a 60-seconds.

- What is the input combinations?

Get our hands dirty

Test Cases

- Your test case should mention:
  - The test case identifier.
  - The input configuration and the expected output.
  - The step-by-step procedures.
  - Some description about the test.
  - Also, if there is a assumption, say it.

- During the test, note:
  - The actual output and the verdict (passed or failed).
  - Also, mention who did the test, and when he/she did it.

Test Cases

- Say we have a requirement:
  - The customer should be able to login with their credentials (combination of email and password).
  - Only customers who have confirmed their emails are allowed to login.

- Based on the requirement, how should your test cases look like?
  - Or what should be tested, where to start?

Test ID	NFR02-01
FR/NFR ID	NFR02 – Security: authentication
Test Name	Login (authenticating) into the system with credential (email and password)
Objective	To ensure that only appropriate customers allowed to login
Description	This mechanism is part of security measure to elevate the system's security by authenticating its users. This test is intended to test the given credentials to the existing customer data. The authentication must reject any credential that is not matched with the existing data, or those who haven't confirmed their email.
Precondition	The user is still not logged in
Date	23 August 2020 18:40WIB
Tester	Jaka Sembung

Testing Scenario

1. Navigates to the login page (/user/login).
2. Enter the input configuration and send the form.

Evaluation Criteria

1. The given credentials are matched with the stored data; and
2. The customer has confirmed his/her email address.

Test Cases

ID	Input	Expected Behavior	Actual Behavior	Verdict
NFR02-01-01	A registered customer with confirmed email address. email: jaksem@examplecom pwd: kvo2d001m-	A successful authentication message is shown and the profile page is opened	An unsuccessful authentication message is shown and the login page is opened	[x] passed [ ] failed
NFR02-01-02	A registered customer without confirmed email address. email: saras008@examplecom pwd: ussox&661	An unsuccessful authentication message is shown and the login page is opened	An unsuccessful authentication message is shown and the login page is opened	[ ] passed [x] failed
NFR02-01-03	An unregistered customer. email: wirsab@examplecom pwd: kw0212axJ	An unsuccessful authentication message is shown and the login page is opened	An unsuccessful authentication message is shown and the login page is opened	[x] passed [ ] failed

Notes

Double checked it and the second case still didn't pass.

Test ID	NFR02-01-API
FR/NFR ID	NFR02 – Security: authentication
Test Name	Login (authenticating) into the system with credential (email and password)
Objective	To ensure that only appropriate customers allowed to login
Description	This mechanism is part of security measure to elevate the system's security by authenticating its users. This test is intended to test the given credentials to the existing customer data. The authentication must reject any credential that is not matched with the existing data, or those who haven't confirmed their email.
Precondition	The user is still not logged in
Date	23 August 2020 18:40WIB
Tester	Jaka Sembung

Testing Scenario

1. Set the target URL (/users/authenticate).
2. Set the request method to POST.
3. Set a JSON-formatted data with the given configuration and send it.

Evaluation Criteria

1. The given credentials are matched with the stored data; and
2. The customer has confirmed his/her email address.

Test Cases

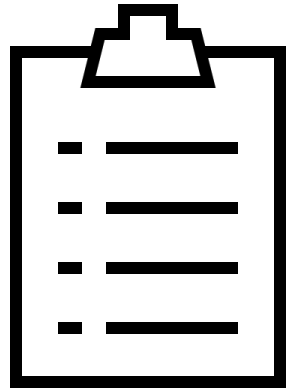
ID	Input	Expected Behavior	Actual Behavior	Verdict
NFR02-01-API-01	A registered customer with confirmed email address. email: jaksem@examplecom pwd: kvo2d001m-	A 200-based response with a JSON-formatted payload contains the authenticated user.	A 200-based response with a JSON-formatted payload contains the authenticated user.	[x] passed [ ] failed
NFR02-01-API-02	A registered customer without confirmed email address. email: saras008@examplecom pwd: ussox&661	A 404-based response with a JSON-formatted payload contains "incorrect credential" message.	A 200-based response.	[ ] passed [x] failed
NFR02-01-API-03	An unregistered customer. email: wirsab@examplecom pwd: kw0212axJ	A 404-based response with a JSON-formatted payload contains "incorrect credential" message.	A 404-based response with a JSON-formatted payload contains "incorrect credential" message.	[x] passed [ ] failed

Notes

Double checked it and the second case still didn't pass.

To-dos

1. Design some test cases for your APIs.
2. Conduct the testing.



References

Srinivasan, M. (2012). Web Technology: Theory and Practice. Pearson.

Mathur, A. P. (2013). Foundations of Software Testing. Pearson

Thank you