University Name

Doctoral Thesis

Thesis Title

Author:
John Smith

Supervisor:
Dr. James Smith

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

Research Group Name
Department or School Name

17 sierpnia 2015 roku

Declaration of Authorship

I, John Smith, declare that this thesis titled, 'Thesis Title' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Spis treści

Spis rysunków

Spis tabel

Rozdział 1

Testowanie oprogramowania

1.1 Rodzaje testów

Testowanie oprogramowania jest procesem oceny oprogramowania w celu wykrycia różnic pomiędzy oczekiwanym i rzeczywistym rezultatem działania. Jest niezbędny w celu osiągnięcia wysokiej jakości. Powinien być wykonywany już podczas rozwoju aplikacji. Można go podzielić na dwa procesy:

- Weryfikacja jest procesem, który ma za zadanie zagwarantować spełnienie warunków ustalonych na początku fazy rozwoju oprogramowania.
- Walidacja jest procesem, który ma za zadanie spełnienie określonych wymagań na końcu fazy rozwoju oprogramowania (by upewnić się, że produkt jest zbudowany zgodnie z wymaganiami klienta).

Możemy wyróżnić dwa rodzaje testów:

- Testy czarnej skrzynki: technika, w której ignorowane są wewnętrzne mechanizmy aplikacji. Osoba projektująca testy tego typu nie musi mieć wiedzy na temat budowy programu, pracować przy rozwoju systemu, ani posiadać wiedzy z zakresu programowania. Ocenie podlega wynik działania systemu w zależności od określonych danych wejściowych. Opierają się na założeniach funkcjonalnych (testy czarnej skrzynki zwane są również testami funkcjonalnymi).
- Testy białej skrzynki: technika, która wymaga od projektanta znajomości wewnętrznych mechanizmów systemy. Znane również jako testy szklanej skrzynki lub testy strukturalne.

- Testy jednostkowe

  Testy jednostkowe (ang. unit testing) to testy, które dotyczą pojedynczego modułu systemu. Najczęściej są pisane przez programistę by zapewnić, że zaimplementowany przez niego moduł (np. klasa) produkuje oczekiwany rezultat gdy przyjmuje określone dane wejściowe.

- Testy integracyjne

  Testy integracyjne (ang. integration testing) to testy, w których grupa komponentów są łączone by wyprodukować rezultat. W tym rodzaju badań potwierdzana jest również poprawność interakcja pomiędzy warstwą oprogramowania i warstwą sprzętową (jeśli taka interakcja zachodzi). W zależności od implementacji mogą być klasyfikowane jako testy białej skrzynki lub testy czarnej skrzynki.

- Testy funkcjonalne

  Testy funkcjonalne (ang. functional testing) to testy, które mają zapewnić, że sprecyzowana funkcjonalność określona w wymaganiach działa. Klasyfikowane jako testy czarnej skrzynki.

- Testy systemowe

  Testy systemowe (ang. system testing) to testy, które mają zapewnić, że integracja systemu w docelowym środowisku (np. systemie operacyjnym) dalej działa. Testy systemowe są robione podczas gdy system i środowisko są w pełni zaimplementowane. Są to testy czarnej skrzynki.

- Testy obciążeniowe (ang. stress testing) to testy, których celem jest ocena jak system się zachowuje w trakcie panowania niekorzystnych warunków. Testy przeprowadzane są w warunkach przekroczenia limitów ustalonych w specyfikacji. Reprezentują testy czarnej skrzynki.

- Testy wydajnościowe

  Testy wydajnościowe (ang. performance testing) to testy określające czy szybkość i efektywność systemu jest zadowalająca np. czy system wygenerował rezultaty poniżej limitu czasu, określonego w wymaganiach wydajnościowych. Klasyfikowane jako testy czarnej skrzynki.

- Testy użyteczności

  Testy użyteczności (ang. usability testing) są wykonywane z perspektywy użytkownika systemu by ocenić czy środowisko graficzne jest przyjazne użytkownikowi. Sprawdzane jest z jaką łatwością klient jest w stanie nauczyć się posługiwać zaprojektowanym systemem, jak wydajnie jest w stanie pracować z aplikacją oraz czy wygląd programu jest przyjemny. Są to testy czarnej skrzynki.

- Testy akceptacyjne

  Testy akceptacyjne (ang. acceptance testing) są najczęściej wykonywane przez klienta by zapewnić, że dostarczony produkt spełnia wymagania i wykonuje zdefiniowaną przez niego pracę. Należą do kategorii testów czarnej skrzynki.

- Testy regresyjne

  Testy regresyjne (ang. regression testing) to testy wykonywane po modyfikacji systemu, komponentu lub grupy powiązanych komponentów by zapewnić, że zmodyfikowany system pracuje poprawnie, nie jest uszkodzony oraz nie powoduje, że inne moduły produkują nieoczekiwane rezultaty. Są to testy czarnej skrzynki.

- Testy beta

  Testy beta (ang. beta testing) są wykonywane przez użytkowników końcowych, zespół złożony z osób nie biorących udziału w rozwoju danego oprogramowania. Często w celu przeprowadzenia testów beta opublikowana zostaje wersja produktu zwana wersją beta. Klasyfikowane jako testy czarnej skrzynki.

1.2 Learning LaTeX

section

1.2.1 A (not so short) Introduction to LaTeX

If you are familiar with LaTeX, then you can familiarise yourself with the contents of the Zip file and the directory structure and then place your own information into the '`Thesis.cls`' file. Section **??** on page **??** tells you how to do this. Make sure you read section **??** about thesis conventions to get the most out of this template and then get started with the '`Thesis.tex`' file straightaway.

1.2.2 A Short Math Guide for LaTeX

If you are writing a technical or mathematical thesis, then you may want to read the document by the AMS (American Mathematical Society) called, "A Short Math Guide for LaTeX". It can be found online here:
`http://www.ams.org/tex/amslatex.html`
under the "Additional Documentation" section towards the bottom of the page.

1.2.3 About this Template

This LaTeX Thesis Template is originally based and created around a LaTeX style file created by Steve R. Gunn from the University of Southampton (UK), department of Electronics and Computer Science. You can find his original thesis style file at his site, here:
http://www.ecs.soton.ac.uk/~srg/softwaretools/document/templates/

My thesis originally used the '`ecsthesis.cls`' from his list of styles. However, I knew LaTeX could still format better. To get the look I wanted, I modified his style and also created a skeleton framework and folder structure to place the thesis files in.

This Thesis Template consists of that modified style, the framework and the folder structure. All the work that has gone into the preparation and groundwork means that all you have to bother about is the writing.

Before you begin using this template you should ensure that its style complies with the thesis style guidelines imposed by your institution. In most cases this template style and layout will be suitable. If it is not, it may only require a small change to bring the template in line with your institution's recommendations.

Dodatek A

Appendix Title Here

Write your Appendix content here.