Specification-based Techniques

# Introduction

The software development process is a comprehensive process initiated from the project's idealization to the actualization of the idea to solve a given business need in the business domain. Software development processes can be approached methodologies such as Secure Feature has driven development (Durelli, et.al 2019). The feature development methodology is tailored to focus on software features to achieve the functionalities as intended in the development process. This is to enhance the relevancy of the software under development.

# Specification-based Techniques

Development techniques in the software development process can be diversely deployed depending on the scenario to be used. Specification-based techniques are some of the development techniques used for an efficient software development process. Specification-based techniques are founded on software specifics in terms of requirements, system modules, and objectives as detailed in the respective scenario. Specification-based testing can be anchored on software requirements to achieve robust testing.

## Requirements as Specification

Requirements modeling is a vital aspect of any software development process designed to fine-tune a given case scenario into the specifications in terms of requirements to address the business solution. Specification-based techniques include software objectives, requirements, and testing. Software requirements are specifications defining the business needs for the respective software under development. This is mostly the commonly used specification-based approach in the software development process, in such a case, the development process is specifically based on the requirements specified to address the business needs in a given market niche.

## Specification-based Testing

Testing is an important part of the software’s life cycle designed to evaluate the software’s efficiency and effectiveness in the respective field of operation. The specification-based testing technique is an enhanced testing technique that is majorly focused on intra-class and inter-class testing levels. Intra-class testing in this perspective tests the inter-relationships between different modules in a class in a given software under development. Inter-class testing on the other hand tests the interrelation between different classes in the program. Specification-based testing is an efficient testing approach in the problems emanating from object-oriented attributes such as aggregation, inheritance, and polymorphism.

**Fault testing**

Software faults are programming flaws that may arise in the process of development. A test case from fault specification techniques gives a mitigation idea. As a technique in specification-based testing, the tester works with a hypothesis before coding and a real scenario after the completion of the program.

**Decision table/ Cause and effect graphing**

This technique is an "if and else" testing technique in specification-based testing. The conditions specified in this case are the if cases covered in the response given in the effect feature of testing. Decision tables and the cause and effect graphs are considered shallow techniques of testing concerning robustness in testing for bug identification in a given software program. This testing approach in the specification-based testing perspective can be categorized under black-box testing since it is drafted without the actual code for its conditions.

**Use case testing**

As a part of the specification-based testing, the use case testing approach can be considered a form of functionality testing designed to evaluate the use cases of a given software solution. This examines the individual functionality. Since the output from this test approach is already known, it can be viewed as white-box form testing to identify software bugs.

**State transition tables**

State transitions are much alike to the decision tables, a slight disparity emanates from the fact the state transitions seek to examine the several states of the software application. This test evaluates the stability of the software program from all the valid states as the coverage area of the test cases.

**Boundary Value Analysis (BVA)**

BVA analysis evaluates the respective boundaries of the software's partitions, this is a common testing technique in specification-based testing, experience-based, and structure-based testing.

# Conclusion

The importance of software in business necessitates an efficient software development process designed to achieve ultimate efficiency in software to fill the identified business gap. This has necessitated various design and development techniques and methodologies to enhance the ultimate efficiency of the software. Specification-based is one such technique that specifically addresses the respective software specifics. The software specifications confine the design and development process to requirements, test cases, and objectives as part of the specifications in the SDLC process (Garousi, et.al 2017). Usually, the objective specification is an aspect in specification-based testing that specifies what is to be achieved at the end of the entire development life cycle. This specifies the quality assurance of the software under development.

# Reference List

Durelli, V.H., Durelli, R.S., Borges, S.S., Endo, A.T., Eler, M.M., Dias, D.R. and Guimarães, M.P., 2019. Machine learning applied to software testing: A systematic mapping study. *IEEE Transactions on Reliability*, *68*(3), pp.1189-1212.

Garousi, V., Felderer, M., Kuhrmann, M. and Herkiloğlu, K., 2017, June. What industry wants from academia in software testing? Hearing practitioners' opinions. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering* (pp. 65-69).