Università degli Studi di Salerno
Dipartimento di Informatica

Tesi di Laurea di I livello in
Informatica

Test Driven Development for Embedded Systems

Relatore
Giuseppe Scanniello
Correlatore
Dott. Nome Cognome

Candidato
Michelangelo Esposito

Academic Year 2021-2022

Abstract

...

The purpose of this thesis is to analyze the benefits and/or drawbacks derived from the application of Test Driven Development (TDD) as part of the software development lifecycle of Embedded Systems.

Contents

List of Figures

List of Acronyms and Abbreviations

Chapter 1

Introduction

Chapter 2

Problem formulation

2.1 Test Driven Development

At its core, TDD is made up of three iterative phases: "Red", "Green" and "Blue" (or "Refactor"):

- In the "**Red**" phase, a test case is written for the chunk of functionality to be implemented; since the corresponding logic does not exist yet, the test will obviously fail, often not even compiling.
- In the "**Green**" phase, only the code that is strictly required to make the test pass is written.
- Finally, in the "**Blue**" phase, the implemented code, as well as the respective test cases, is refactored and improved. It is important to perform regression testing after the refactoring to ensure that the changes didn't result in any unexpected behaviors in other components.

Each new unit of code requires a repetition of this cycle [1].

The figure below provides a representation of the TDD cycle:

The general mantra of TDD revolves around the "Make it green, then make it clean" motto.

2.2 Testing Embedded Systems

Embedded Systems (ES) can be defined as a combination of hardware components and software systems that seamlessly work together to achieve a specific purpose. Such systems can be dynamically programmed or have a fixed functionality set, and are often engineered to achieve a domain-specific, often

critical, goal. In recent years, such system have seen a surge in popularity, and have driven innovation forward in their respective areas of deployment: everywhere, spanning from the agricultural field, to the medical and energy ones, ES of various size and complexity are employed.
Due to their high specialization, ES often deal with time and resource constraints, both in hardware and in software; often these systems are battery-powered and therefore the hardware they are equipped with, often purpose built, must be highly efficient in its operations. Furthermore, from the software point-of-view, it is essential that the system operates deterministically and with real-time constrains.
Failures in ES should always be evident and identifiable quickly (a heart monitor should not fail quietly [2]). Given the high criticality of such systems, ensuring their dependability over the course of their lifespan is essential; ES can be deployed in extreme conditions (i.e., weather monitoring in extreme locations of the planet, devices inside the human body, or ...), where maintenance operations cannot be performed regularly, and high availability is expected.

Furthermore, given the absence of a user interface in most cases, testing such systems can be particularly challenging, given the lack of immediate feedback.

2.3 Test Driven Development for Embedded Systems

Generally, the testing process of ES follows the X-in-the-loop paradigm [3]; subcategories in this area include model-in-the-loop, software-in-the-loop, processor-in-the-loop, hardware-in-the-loop, and system-in-the-loop. Reference the old survey papers (i.e. X in the loop) that provide a summary of the main techniques

Chapter 3

Literature

Chapter 4

Conclusions

Bibliography

[1] *Guidelines for Test-Driven Development*. URL: https://learn.microsoft.com/en-us/previous-versions/aa730844(v=vs.80)?redirectedfrom=MSDN.

[2] White E. *Making Embedded Systems: Design Patterns for Great Software*. O'Reilly, 2011.

[3] Vahid Garousi et al. "What We Know about Testing Embedded Software". In: *IEEE Softw.* 35.4 (2018), pp. 62–69. DOI: 10.1109/MS.2018.2801541. URL: https://doi.org/10.1109/MS.2018.2801541.

[4] Ross B. Girshick et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, 2014, pp. 580–587. DOI: 10.1109/CVPR.2014.81. URL: https://doi.org/10.1109/CVPR.2014.81.

[5] Jindi Zhang et al. "Evaluating Adversarial Attacks on Driving Safety in Vision-Based Autonomous Vehicles". In: *IEEE Internet Things J.* 9.5 (2022), pp. 3443–3456. DOI: 10.1109/JIOT.2021.3099164. URL: https://doi.org/10.1109/JIOT.2021.3099164.

[6] Edmund K. Burke and Graham Kendall. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer US, 2014, pp. 403–449.

[7] *Check: framework for unit testing in C*. URL: https://libcheck.github.io/check/.

[8] A. Author and A. Author. *Book reference example*. Publisher, 2099.

[9] A. Author. "Article title". In: *Journal name* (2099).

[10] *Example*. URL: https://www.isislab.it.

[11] A. Author. "Tesi di esempio ISISLab". 2099.