AN ANALYSIS OF GOF SOFTWARE DESIGN PATTERNS ON SOFTWARE MAINTAINABILITY IN MICRO SERVICES

R. M. H. S. B. Rathnayake
209370G

Degree of Master of Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

June 2023

AN ANALYSIS OF GOF SOFTWARE DESIGN PATTERNS ON SOFTWARE MAINTAINABILITY IN MICRO SERVICES

R. M. H. S. B. Rathnayake
209370G

Thesis submitted in partial fulfillment of the requirements for the degree
Master of Science in Computer Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

June 2023

DECLARATION OF THE CANDIDATE AND SUPERVISOR

I declare that this is my own work, and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another. person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: …… Date: 26/06/2023

Name: R.M.H.S.B.Rathnayake.

The supervisor/s should certify the thesis/dissertation with the following declaration.

The above candidate has carried out research for the Masters thesis/dissertation under my supervision. I confirm the declaration made above by the student is true and correct.

Signature of the supervisor: …… Date: 26/06/2023

Name of the supervisor: Eng. Prof. Indika Perara.
Head - Dept. of Computer Science & Engineering, Faculty of Engineering
University of Moratuwa, Moratuwa 10400, Sri Lanka.

ACKNOWLEDGEMENT

I express my sincere gratitude to my supervisor Eng. Prof. Indika Perara for the guidance he provided me throughout this research. Their knowledge, expertise and encouragement helped me to complete this research.

I would also like to thank the companies that helped to gather data for this research. Without their help this research wouldn't have been possible.

Also, I would like to thank my colleagues and friends who helped me with gathering data and for providing me with technical advice regarding the tools and technologies used in this project.

Also, I would like to thank my family for their support in completing this research.

Abstract

The purpose of this paper is to identify how gang of four design patterns impact the maintainability of micro services-based systems. Design patterns were introduced as solutions to common problems that occur in programming. These are supposed to improve the maintainability of a system by improving the code quality. But with modern programming languages, frameworks, and integrated development environments, whether these patterns serve their purpose is a question that is not fully addressed. This paper proposes a tool that can be used to identify whether use of a specific design pattern by a specific developer for a particular micro service-based project can improve its maintainability or not. To do this a model has been created by analyzing enterprise micro service-based applications and gathering data from developers who were involved in the development of those projects. This data is used to create models for maintainability metrices coupling, lack of cohesion, duplication and cyclomatic complexity. This tool will help to decide whether the system is more maintainable with or without the use of selected design patterns. This helps better decision making in deciding how to write new code or refactoring existing code. Results of this research have shown that lack of cohesion is not affected by developer experience, design patterns or the language used in enterprise micro service-based applications. Cyclomatic complexity was only affected by the language used. Also, use of certain design patterns decreased the coupling in the system. But some of the design patterns caused duplications to be increased. So, the results showed that use of design patterns can have a negative and positive impact on the maintainability of a microservice depending on the design pattern used. This research also emphasizes the importance of code review process and code quality analysis automation.

Keywords: GOF design patterns, micro services, software maintainability

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
AHEF	Attributed hiding effectiveness factor
AIF	Attribute inheritance factor
ALT	Alternative version without patterns
API	Application programming interface
Avg	Average
CAM	Cohesion among methods
CBO	Coupling between objects
CC	Cyclomatic complexity
CF	Coupling factor
COM	Percentage of comments per module
GOF	Gang of four
HTML	Hypertext markup language
HTTP	Hypertext transfer protocol
HV	Halstead volume
IDE	Integrated development environment
IIF	Internal inheritance factor
LCOM	Lack of cohesion methods
LOC	Average number of lines per module
LOC	Lines of code
MI	Maintainability index
MIF	Method inheritance factor
MOOD	Matrices for object-oriented design
OHEF	Operation hiding effectiveness factor
PAT	Pattern version
PF	Polymorphism factor
PPF	Parametric polymorphism factor
P-value	Probability value
RFC	Response for a class
UI	User interface

LIST OF APPENDICES