IDENTIFYING SOFTWARE ARCHITECTURE EROSION THROUGH CODE COMMENTS

Vidudaya Neranjan Bandara

(168206G)

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

June 2018

IDENTIFYING SOFTWARE ARCHITECTURE EROSION THROUGH CODE COMMENTS

Herath Mudiyanselage Vidudaya Neranjan Bandara

(168206G)

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

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

June 2018

DECLARATION

I declare that this is my own work and this thesis 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.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books)

Signature: ..... Date: ...….....

H.M.Vidudaya Neranjan Bandara

The above candidate has carried out research for the Masters thesis under my supervision.

Name of the supervisor: Dr. Indika Perera

Signature of the supervisor: ..... Date: .....

Abstract

Software architecture erosion or the as-implemented architecture is not complying with the as-intended architecture is one of the major problems faced by many organizations. There is no easy way to trace design decisions or tracking back or reconstructing those decisions by looking at the source code level elements is one of the major reasons for software architecture erosion. Other than that the mistakes or carelessness of the programmer may lead the system to an eroded status eventually. Lack of domain knowledge, lack of knowledge about intended architecture and unable to identify possible violations of as-intended architecture (by identifying architectural degradation) are some other reasons for software architecture erosion.

There are various methodologies and tools for architecture conformance checking and analyzing the static architecture and provide comparison results which can be used to determine whether the architecture of a system is altered or not [10]. Most of them require high end tool support and providing the implemented architecture and the intended architecture each time the analysis needs to done.

As the main research objective it identified a missing area of software architecture conformance checking methodologies and analyzed and identified a way to prevent software architecture erosion using that. This research is more focused on unconventional usability of the code comments and how it can be leveraged to capture the architecture of the application and how it can be used as an effective architecture conformance checking mechanism.

This research states a methodology which uses Java Doc comments to inject architecture specific information into the code base and a mechanism to capture them and compare them with a pre-defined architecture rule set. An empirical and theoretical evaluation has been done to prove this concept actually works in real life scenarios. It opened up a new area of architecture conformance checking to the future researchers of the field of software architecture.

ACKNOWLEDGEMENT

My sincere appreciation goes to my family for the continuous support and motivation given to make this thesis a success. I also express my heartfelt gratitude to the research supervisor Dr. Indika Perera, for the supervision, advice and continues guidance given throughout to make this research a success. Also I am grateful for the support and advice given by Dr. Malaka Walpola, by encouraging continuing this research.

My special thanks go to Mr. Umesh Indith Liyanage and Dr. Rasika Withanawasam for sharing their knowledge regarding software architecture and software architecture conformance.

Finally I wish to thank the academic and nonacademic staff of Department of Computer Science and Engineering and colleagues of MSC'16 for the support and encouragement given.

TABLE OF CONTENTS