A MONITORING FRAMEWORK FOR SOFTWARE ARCHITECTURE DEGRADATION IN DEVOPS PRACTICE

Gonagala Withanage Piyumi Sampath Gunarathna

158216D

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

March 2019

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 thesis, 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:

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:

Acknowledgement

I am deeply grateful to my supervisor Dr. Indika Perera from the University of Moratuwa Department of Computer Science and Engineering whose help, motivation, suggestions and encouragement helped me in all the time to complete this work. Dr. Indika Perera inspired me greatly to work in this project.

I would like to thank my family and friends for their understanding during the period of the project and gave me support. Without their help I would face many difficulties in the time of the project. And finally my grateful thanks to all the people who help me during the project.

Abstract

This is a research report that desired to carry out to find the software architecture issues and possible monitoring techniques to overcome those issues in DevOps practice**.** DevOps is a new philosophy that helps software organizations to innovate faster and to be more responsive to business needs, it promotes collaboration between developers and operations which improves quality of software development and more frequent software releases.

Continues delivery in shorter development iterations and deploy a software system faster are the key practice of DevOps. And also because of that faster practice, there is huge risk of occurring a software failure unless the continuous monitoring. Therefore the DevOps teams use automated tools for monitoring functional criteria and the performance. But still, most of the teams are not monitoring the architecture of the application.

Unless there need to be done a change intently, the architecture of the system need to pertain its initial designed and finalized architecture in DevOps culture, in shorter development iterations. Therefore it is more important to monitor the software architecture without leading to a software drift or erosion. Therefore this research objective is to build a Monitoring framework for architectural degradation in DevOps practice.

The end goal is Continuous Testing and Continuous Monitoring. Testing and Monitoring are what will prove that the new built is the right required application, that functions and performs as designed and desired.

As the main research objective it identified a missing area of software architecture monitoring methodologies and analyzed and identified a way to prevent software architecture erosion using that. This research is more focused on unconventional usability of the solution and project file contents and how it can be leveraged to capture the architecture of the application and how it can be used as an effective architecture design monitoring framework.

This research states a methodology which uses project file and solution file content to detect the architecture specific information from 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.

Keywords

DevOps, Continuous integration, Continuous deployment, Defect warnings, Continuous monitoring, Software development lifecycle, Quality assurance, SDM

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES