Informatics Institute of Technology

In collaboration with

University of Westminster, UK.

Utilizing Contextualized Source Code to Plan Emerging Software Change Requests

A dissertation by

G.T. Savindra Perera (2013042)

Supervised by

Mr. Sudharshana Welihinda

Submitted in partial fulfillment of the requirements for the

BSc (Hons) Software Engineering degree

Department of Computing

Abstract

Maintenance phase of a software project starts usually after the delivery of its first artifact or feature. Maintenance of a software is distinguished by the high cost and time consuming development of changes. Software change requests (CR) are an unavoidable part of software maintenance and evolution, it is not uncommon to observe software projects with large code bases receive thousands of CRs per day. Finding the appropriate developer to undertake CRs while maintaining the most economic and shortest fixing time is an issue which got attention and causes CR handling to be much complicated and challenging. However a developer can start implementing changes or fixes after CR is assigned. At the beginning, most software developers spend a significant time on searching and traversing through the source code to locate relevant features in source code that has an impact from CR. Despite of the nature of change such as modification, addition or deletion of software artifacts, developer cannot proceed further without first locating features in the source code. These problems are very frequent and common in handling CRs and have caused software maintenance to be high costly.

As an attempt to solve these problems, this research project presents a tool named CR-Assistant to recommend feature locations in source code which may have an impact from CR and developers who authored source code in recommended feature locations. CR-Assistant employs textual feature location technique using information retrieval that consider developer's Git commit descriptions/messages and issue descriptions in issue trackers as a data source which describes artifacts in the source code. Unlike in most of the previous researches, textual feature location technique that relies on Git commits and issue descriptions makes the application of CR-Assistant programming language independent. The effectiveness of recommendations provided by the tool was assessed using change requests on open source projects RoundCube, Visual Studio Code and Angular.

Subject Descriptors:

- Information Systems ~ **Information Retrieval** ~ Document representation ~ Content analysis and feature selection
- Software and its Engineering ~ Software notations and tools ~ Software maintenance tools

Keywords:

Software maintenance, Change Request, Information Retrieval, Feature Location, Triage