Chapter 5

Tool Support for Software Artefact Traceability in DevOps Practice:
SAT-Analyser

Iresha Rubasinghe
University of Moratuwa, Sri Lanka

Dulani Meedeniya
https://orcid.org/0000-0002-4520-3819
University of Moratuwa, Sri Lanka

Indika Perera
University of Moratuwa, Sri Lanka

ABSTRACT

Software development in DevOps practice is a widely used approach to cope with the demand for frequent artefact changes. These changes require a well-defined method to manage artefact consistency to ease the continuous integration process. This chapter proposes a traceability management approach for the artefact types in the main phases of the software process including requirements, design, source code, testing, and configuration. This chapter addresses traceability management, including trace link creation, change detection, impact analysis, change propagation, validation, and visualisation. This chapter presents a tool named SAT-Analyser that is applicable for any software development method and designed for continuous integration, multi-user collaboration, and DevOps tool stack compatibility. The SAT-Analyser is assessed using case studies and shown an impact analysis accuracy of 0.93 of F-measure. Further, the feedback by DevOps practitioners has shown the suitability and innovativeness of the proposed approach.

DOI: 10.4018/978-1-7998-1863-2.ch005

Copyright © 2020, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
INTRODUCTION

A software system is a combination of several software artefacts that evolves through a software development process model. Software artefacts refer to the intermediate by-products used in different phases of the SDLC such as SRS documents, design diagrams, architectural documents and quality attributes or the non-functional design reports, source code, test scripts, walkthroughs, inspections, bug reports, build logs, test reports, project plans, risk assessments (Sommerville, 2010). It is essential to manage the relationships and dependencies among artefacts to maintain adequate consistency towards the completion of the software product. The improper management and outdated elements can lead to inconsistency among artefacts, synchronisation issues and lack of trust for the system by stakeholders. Therefore, software artefact traceability is required to follow the artefact life cycle during the software development process.

DevOps is a recently emerged software development practice that increases the collaboration among developers and operations teams. It is required to manage the consistency among the software artefacts throughout the SDLC phases and project teams, with the nature of frequent artefact changes. Traceability supports to track the artefact changes, their transformations and relationships in both forward and backwards directions. Traceability management is a multi-step process and should ensure the correctness and performance (Maro, Anjorin, Wohlrab, & Steghöfer, 2016) (Mäder & Gotel, 2012). However, traceability in practice is popular due to the high cost and effort required to manage the artefacts relationships and maintain consistency during changes. Also, there is a lack of automated and platform independent tool support in traceability management. Thus, automated traceability management and consistency maintenance that covers a variety of artefacts in software development are essential.

This chapter addresses traceability and artefact consistency management in DevOps practice. We propose an approach for software artefact traceability management and a prototype tool ‘Software Artefact Traceability Analyser’ (SAT-Analyser) as the proof-of-work. This study considers different software artefact types representing the main activities in a software process. These artefacts include requirements in natural language text, UML class design diagram, Java source code,JUnit test scripts and build-scripts configuration files. The methodology consists of several modules to manage traceability through the trace-link establishment, change detection, change impact analysis (CIA), change propagation, visualisation, validation and integration with the DevOps tool stack. The applicability of the tool is evaluated using case-study based analyses and a usability study among DevOps practitioners. Thus, the proposed approach attempts to fulfil the research hindrance in traceability support in DevOps practice.
Big Data and Its Visualization With Fog Computing
www.igi-global.com/article/big-data-and-its-visualization-with-fog-computing/210566?camid=4v1a