Chapter 4

Artefact Consistency Management in DevOps Practice: A Survey

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

Iresha Rubasinghe
University of Moratuwa, Sri Lanka

Indika Perera
University of Moratuwa, Sri Lanka

ABSTRACT

DevOps practices preserve the continuous innovation in software development. The collaborative nature and stakeholder communication are keys in DevOps that lead to highly effective and quality software outcomes with customer satisfaction. The software artefacts involved in a DevOps practice must adapt to frequent changes due to continuous stakeholder feedback. Hence, it is challenging to artefact consistency throughout the software life cycle. Although artefact traceability preserves the consistency management with theoretical support, there are practical limitations in traceability visualisation, change impact analysis, and change propagation aspects. This chapter presents an analysis of existing studies focused on software artefact traceability for the suitability in DevOps. It also identifies leading limitations and possible future research directions to resolve for the benefit of researchers and software practitioners.

DOI: 10.4018/978-1-7998-1863-2.ch004

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

Software System

Overview of a Software System

A software system is a combination of several software elements that evolves through a particular software development process model. It is an interface that connects the user with application software and computer hardware. It is a combination of a set of design decisions that lead to system architecture, which is a blueprint for any software system (Arora, & Arora, 2016). The study of theoretical concepts related to software system development, technical aspects, budgeting, management and maintenance is known as software engineering (Sommerville, 2010). With the rapid improvements in technology and resources, the importance of software systems has become vital in everyday activities. For instance, different domains such as finance, transportation, agriculture, military, academics, healthcare, business rely on software systems (Chang, 2005) (Sommerville, 2010). In practice, the aim is to maximise the use of automated software systems to minimise manual workforce and to improve quality. Thus, several well-defined software process models and technologies have been used in software system development.

Software Artefacts

Software artefacts refer to the intermediate by-products used in different phases of software development. These elements include System Requirement Specification (SRS), 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 and risk assessments among many (Sommerville, 2010). Each artefact has its life cycle during software evolution. The types of artefacts in a software project may vary depending on the adopted software process model and technologies. Thus, a software system is a result of a collection of elements that goes through changes affecting each other at different levels. There are relationships and dependencies between these software artefacts, and it is essential to manage these software artefacts to maintain adequate consistency during changes. The improper management and outdated artefacts can lead to inconsistencies, synchronisation issues and lack of trust by stakeholders (Cleland-Huang, Zisman, & Gotel, 2012).
The Compute Infrastructures for Big Data Analytics
www.igi-global.com/chapter/compute-infrastructures-big-data-analytics/103211?camid=4v1a

Overview of Big Data-Intensive Storage and its Technologies for Cloud and Fog Computing
www.igi-global.com/article/overview-of-big-data-intensive-storage-and-its-technologies-for-cloud-and-fog-computing/219362?camid=4v1a