CHAPTER

4

AN EXPERIENCE OF SYSTEM DEVELOPERS USING GITLAB

Lina Fatini Azmi, Wan Lokman Wan Bedurdin, Farizie Abdul Razak, and Nazmona Mat Ali

4.1 INTRODUCTION

This chapter aims to describe the best practise of GitLab implementation in a research university, in the context of the experiences of the developer team. The contribution of this chapter is to provide lessons learned to others about GitLab implementation in the system development team.

The development of innovative electronic projects for research universities is known for its complexity and requires a large number of resources. They are complex because of the combination of innovation, information, and communication technologies, politics, and social impacts (Wen et al., 2020). Handling such collaborative initiatives can be difficult. Poor project management is one of the most common reasons for project failure, owing to issues such as a lack of creativity and vision, poor communication and organisational skills, a hazy work breakdown, poor scope definition, ineffective workload management, and change management, as well as shifting requirements, technical complexity and poor delegation and tracking (Wen et al., 2020). Other particular issues emerge, such as coordinating the project's teamwork, identifying goals, and synchronising the pace of work among developers (Goldfinch, 2007). As a

result, proper management of the project should be a key focus in order to develop effective project solutions.

The significant success of the GitLab project in delivering complicated and high-quality products demonstrates the benefit of incorporating its principles and methods into collaborative situations (Wen et al., 2020). The benefits of this strategy are reduced software and business development expenses, faster time-to-market, improved performance, and a focus also economic profit (Kourtesis et al., 2012). Complex and large-scale projects in research universities cause delays or even failures in the delivery of digital services. One way to tackle these challenges is to use agile methodologies in system development to update large-scale existing infrastructure and adjust to environmental changes and stakeholder requirements more quickly (Mergel, 2016).

The adoption of agile practises in a research university is an important topic to cover in this chapter. For a successful GitLab implementation, the authors will highlight managerial, organisational, process, and technology concerns that must be rethought and reconfigured in a research university. According to the researchers, agile methodologies are better suited to projects with a wide range of requirements, technical capabilities, and technologies. In terms of development activities, agile practises place a premium on increased communication and collaboration (Bheri & Vummenthala, 2019). The researchers emphasised the importance of GitLab values and practises in addition to the well-known agile methodologies.

Moreover, this chapter suggests approaches to solve issues that occurred in the system development section of a research university. The researchers also propose a framework of process flow to implement GitLab to manage our teams and resources. To do so, the researchers need to gather the ideas and experiences of system development teams in a research university. Based on those experiences, it helps the researchers to create the base of the framework and support all the processes in the framework.

This chapter is organised by firstly highlighting the purpose of this paper and describing the problems in system development teams in Section 4.1. Then, the next section describes the introduction of GitLab

and presents related work in Section 4.2. Section 4.3 describes the GitLab framework in a research university, while Section 4.4 explains the issues and challenges faced by developers in a research university. Section 4.5 discusses the results and advantages of implementing GitLab as a lifecycle tool, and finally, Section 4.6 of this chapter provides the conclusion for this chapter.

4.2 REVIEW OF GITLAB IMPLEMENTATION IN SYSTEM DEVELOPMENT LIFECYCLE

A comprehensive review of GitLab and the implementation approaches by other organisations can be found in this section. The brief history of GitLab and description of its features are also included in this section.

4.2.1 DevOps Framework for Agile Development

DevOps is a diverse and collaborative endeavor within a company that automates the continuous delivery (CD) of new software versions while ensuring their correctness and performance (Leite et al., 2019). DevOps culture is a method of facilitating communication amongst IT teams while also speeding up and improving the maintenance cycle. DevOps encourages rapid and continuous feedback from operations to development in order to identify problems before they may affect users (Lopez-Pena et al., 2020).

Moreover, DevOps is an advancement of the Agile movement, and it combines Agile methods with an automated toolchain infrastructure to allow continuous integration (CI), testing, quality assurance, and deployment. Small release iterations with user feedback are encouraged in agile system development. It is expected that the team will be able to release the system regularly in a production-like environment (Leite et al., 2019). Therefore, in this chapter, an explanation of a DevOps infrastructure for agile development is provided based on the GitLab code repository.