

EXAMENSARBETE Using DevOps to Improve Feedback and Traceability for Performance
in Software Development**STUDENTER** Hanna Höjbert, Elias Vernersson**HANDLEDARE** Lars Bendix (LTH)**EXAMINATOR** Emelie Engström (LTH)

How to find performance issues before production?

POPULAR SCIENCE PAPER **Hanna Höjbert, Elias Vernersson**

The change of software performance can sometimes go unnoticed if proper feedback system are not implemented. This research therefore intends to investigate the means of using an automated tool to increase feedback of performance issues.

In today's rapidly changing environment timely feedback is of the essence, not only for functionality but also for performance. Testing performance early, before releasing systems to the customer, could therefore provide great value for companies, in terms of feedback. But if there is no defined practice in the development process for how to test performance, it is easy for performance issues to end up in production. If performance issues end up in production, it can ultimately have a negative impact on the user experience and revenue can be lost. Then a tool that senses and informs about performance issues may be appropriate.

In our research, we have examined what requirements developers place on a tool and how a prototype, based on these requirements, affected the development process in terms of feedback and traceability. Our results were a requirements specification, a proof of concept and problem validation. Through the prototype, developers were able to get faster feedback and issue identification. Furthermore, traceability increased in the development process and in the code base.

The requirements were developed through an iterative process where the first iteration involved interviews with developers and inspiration from literature to produce an initial requirements specification. The second iteration involved developing different designs based on the requirements

and eliciting additional requirements with the help of evaluation with developers. In the last iteration, a prototype was implemented to validate the requirements and elicit additional requirements. The prototype was then evaluated by developers and can be seen in figure 1.

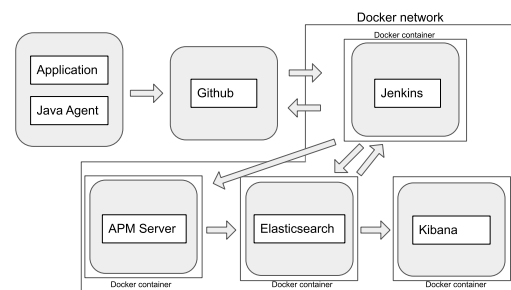


Figure 1: Architecture of prototype when running on build server

Overall, our research shows that our prototype could increase the traceability between performance issues and the code base and allow for the developers to receive more timely feedback about performance issues. This while also providing a stronger sense of closure, along with a requirements specification of the prototype that is able to be implemented in a general software development environment.