

Creating Tomorrow's Software Development Environment at Swiss Post

Degree programme : MAS Information Technology

This thesis explores the feasibility of standardising development environments at Swiss Post using Dev Containers and GitHub Codespaces. The research provides insights into improving developer productivity and reducing costs by addressing inefficiencies in onboarding and environment consistency.

Context and Problems

Since 2024, Swiss Post has adopted GitHub as the primary platform for source code management (SCM) and continuous integration/continuous deployment (CI/CD). Developers work on managed Windows or macOS devices, but the current local development environments suffer from inconsistencies. This leads to inefficiencies in onboarding new developers and discrepancies between local and CI environments. These issues contribute to the well-known „it works on my machine“ problem, where builds fail in CI despite working locally. The onboarding process for new developers is inefficient due to manual setup procedures, which lead to inconsistencies in development environments. Over time, tool versions and configuration variations cause discrepancies between local and CI/CD environments, resulting in unexpected build failures. These challenges slow down development, increase maintenance efforts, and reduce overall productivity.

Objectives and Goals

The aim is to standardise and streamline the development environment for Swiss Post's Java teams. To achieve this, a Dev Containers prototype was developed, and a proof of concept (PoC) was conducted with GitHub Codespaces. Additionally, to improve developer flexibility, the feasibility of a bring-your-own-device (BYOD) policy was evaluated. The solution automates tool management, ensures version consistency, and simplifies setup procedures. The key objectives include reducing onboarding time for new developers, improving consistency across development environments, and analysing the cost-effectiveness of various solutions.

Methods

A SWOT analysis was made to show the risks and chances of using GitHub Codespaces. The requirements for the development environment were collected through an internal survey to all Java devel-

opers, with quantitative and qualitative questions to capture a broad range of requirements. Prototyping and testing were used to build a prototype that fulfils the goals. A self-experiment measured and compared the onboarding time between the old and the new prototype environments.

Results

The internal survey, completed by 28 Java developers, indicated higher satisfaction among macOS users (7.4 out of 10) than Windows users (5.2 out of 10). Furthermore, 75% of developers would like to have a BYOD option. 86% of Swiss Post Java developers use the JetBrains IntelliJ IDEA IDE as their primary IDE. GitHub Codespaces was initially considered as a possible solution, but in January 2025, GitHub silently removed the mention of the JetBrains IDE from the GitHub Codespaces docs. Further internal networking constraints at Swiss Post and the corporate proxy made using GitHub Codespaces for daily work impossible. The survey also revealed that most (56%) Java developers are not interested in using a cloud development environment. To ensure consistency across environments, the prototype replaces manual configurations (currently used by 57% of developers) with Dev Containers. This reduces configuration drift and ensures alignment between local development and CI pipelines. Additionally, cost savings were achieved by eliminating Docker Desktop licenses in favor of a setup that uses WSL2 on Windows and Lima on macOS, preserving compatibility with IntelliJ and Swiss Post's corporate proxy. The prototype significantly improved onboarding efficiency. The total setup time was reduced by 55% (from 1 hour 32 minutes to 41 minutes), while the active developer interaction required during setup was decreased by 78% (from 1 hour 19 minutes to 17 minutes). This allows developers to start productive work faster, improving overall efficiency.



David Ziswiler
david.ziswiler@fhncloud.ch