Technical University of Munich

School of Computation, Information and Technology
-- Informatics --

Bachelor's Thesis in Informatics

Scalable Multi-Workspace and Multi-Platform Support for Hephaestus

Skalierbare Multi-Workspace- und Multi-Plattform-Unterstützung für Hephaestus

Author:	Dmytro Dmukh
Supervisor:	Prof. Dr. Stephan Krusche
Advisors:	Felix T. J. Dietrich M.Sc.
Start Date:	15.04.2025
Submission Date:	15.08.2025

Abstract

Hephaestus is a platform to empower novice software engineers in project-based learning environments. It collects data from repository hosting services to analyze contributions such as pull requests, reviews, and issues, visualizing these through dashboards and leaderboards. By incorporating elements of game design into collaborative software engineering environments, Hephaestus promotes best practices, transparency, and sustained participation.

While the platform already offers valuable insights, its current technical design presents significant limitations. In particular, Hephaestus supports only GitHub repositories and relies on static, deployment-bound configuration, limiting adaptability and scalability, especially in educational use. This thesis proposes extending Hephaestus with dynamic, user-friendly workspace configurations, introducing multi-workspace support, and integrating GitLab as an additional repository hosting platform. These enhancements will significantly improve ease of setup, scalability, and usability, directly benefiting educational contexts such as the iPraktikum course at the Technical University of Munich (TUM), and supporting easy adoption in multiple organizations.

1 Introduction

Modern software engineering education increasingly relies on platforms like GitHub and GitLab to introduce real-world workflows ([PTR24], [Bol19]). Such tools bolster transparency and structured collaboration, key to agile development and version control.

In engineering education, Project-Based Learning (PBL) offers students structured opportunities to apply theoretical knowledge through practical, real-world projects. Research indicates that giving students such authentic, real-world challenges can significantly enhance their motivation and engagement ([SSB22], [VK17], [Amb+10]). Studies have shown that PBL significantly improves students' academic achievement, critical thinking abilities, and collaborative skills [ZZ24]. At the TUM examples such as iPraktikum and the educational platform Artemis

support PBL by enabling instructors to structure effective, hands-on software engineering experiences.

To support and enrich those environments, gamified platforms like Hephaestus can serve as valuable frameworks that guide students through software engineering practices. Hephaestus motivates developers through gamified features such as scoring, ranks, and league progression. Scores reflect review activity and pull request complexity, rewarding more involved reviews and complex contributions with higher points. Based on this, users advance through leagues, similar to ELO systems in games. In addition to gamified elements, Hephaestus includes an AI mentor component that provides tailored feedback and support, further assisting students in their software engineering learning process. Despite its significant potential, Hephaestus faces limitations, notably static configurations and GitHub-only support. Addressing these challenges is crucial to fostering broader adoption, especially in educational settings.

2 Problem

The current Hephaestus implementation faces critical limitations that impact its scalability, usability, and educational effectiveness. A central challenge is its reliance on static configuration files to define workspaces and repository settings. While some elements—such as repositories to monitor—can be adjusted dynamically, these settings are limited to a single workspace and typically still require manual updates, reducing operational flexibility.

Another major limitation is the absence of multi-workspace support, which prevents the flexible management of parallel groups, teams, or courses. To address this, Hephaestus should enable the creation of isolated workspaces that can be configured independently. Administrators—such as course instructors or company team leads—should be able to manage their own environments without relying on static deployment configurations. At the same time, developers must be able to join existing workspaces to access features like scoring, dashboards, and automated

mentoring. Without this structure, onboarding remains cumbersome and limits the platform's broader adoption.

Finally, Hephaestus currently supports only GitHub. This reliance prevents the integration of alternative platforms such as GitLab, which utilize different integration methods and structures. At TUM, GitHub use is restricted due to institutional policies. Additionally, many industrial partners involved in the iPraktikum course—such as iABG and Siemens—require GitLab for compliance reasons. As a result, projects dependent on GitLab cannot currently take advantage of Hephaestus' collaborative and motivational features, limiting its applicability in both educational and industry-linked learning environments.

3 Motivation

Solving the identified problems will significantly boost Hephaestus' educational impact, growth potential, and user-friendliness. Introducing dynamic, intuitive configuration tools simplifies administration, enabling educators to effortlessly create and manage repositories and workspaces tailored to their specific needs.

Multi-workspace support, inspired by platforms like Slack, ensures clear separation and organizational efficiency, promoting scalability across numerous concurrent projects or educational settings. Furthermore, incorporating GitLab alongside GitHub expands Hephaestus' applicability and adaptability, positioning the platform to easily accommodate additional repository services in the future.

Studies such as those by Calvo et. al [CIF20] underline gamification's ability to significantly enhance student engagement and motivation, particularly in software engineering education. Beyond academic settings, Hephaestus can also play a valuable role in professional training and lifelong learning. Companies may use the platform to support skill development, track peer collaboration, and sustain motivation among their software engineers. By supporting both educational and industrial contexts, Hephaestus positions itself as a flexible and scalable tool for empowering developers across different stages of their learning and career journeys.

4 Objective

The overarching aim of this thesis is to significantly enhance Hephaestus' usability and flexibility, facilitating its wider adoption, particularly within educational settings. Specifically, the thesis will achieve two primary objectives:

1. **Manage Multiple Workspaces**:

The goal is to design and implement a flexible architecture that allows for the configuration and isolation of multiple workspaces within Hephaestus. This will replace the current static setup and make the platform more adaptable for different organizational or project-based contexts.

2. **Integrate GitLab**:

The second goal is to enable Hephaestus' seamless use in educational environments by integrating GitLab and simplifying repository onboarding. This makes the platform compliant with institutional requirements and suitable for large-scale course deployments like iPraktikum.

4.1 Manage Multiple Workspaces

A central objective of this thesis is the introduction of a flexible and dynamic workspace management system to replace the current static, deployment-bound configuration. In the existing setup, workspace and repository settings are configured using a combination of hardcoded values and runtime environment variables, limiting scalability, adaptability, and overall user experience. This complicates onboarding and limits adaptability for diverse project needs.

The updated structure addresses these shortcomings by allowing any workspace owner—such as an instructor, project lead, or organization administrator—to create and configure their workspace directly through a user interface. Each workspace is an isolated environment with configurable repositories, leaderboards, and integrations. Key configuration options, such as leaderboard update intervals, reset times, and Slack channel bindings, will no longer be statically defined but instead offered as editable parameters within each workspace. This enables granular control and greater adaptability.

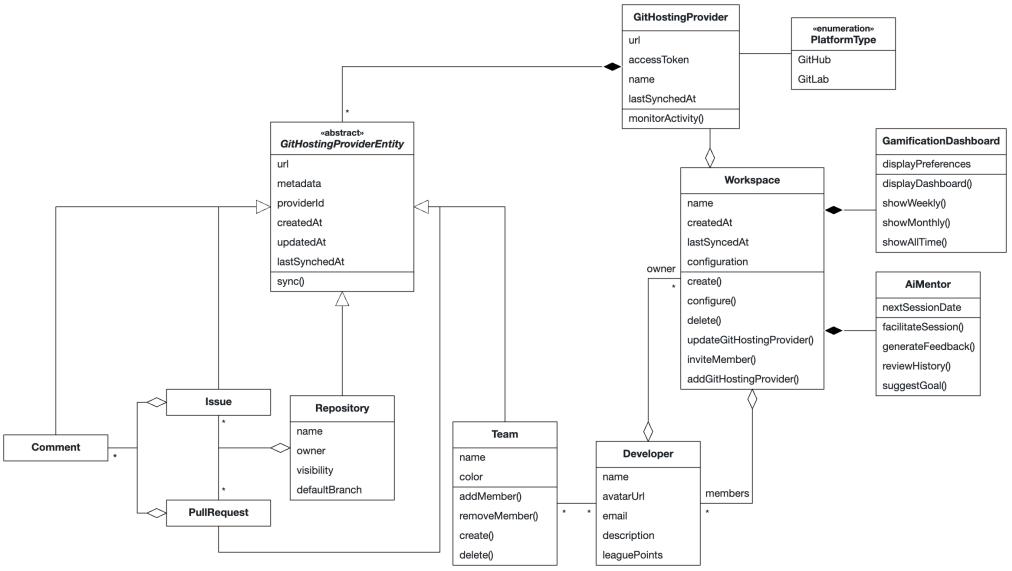

Figure 1: Enhanced Architectural UML Diagram

Figure 1 presents the updated Analysis Object Model (AOM), which captures the redesigned backend structure of the platform. At its center is the Workspace entity, which organizes its components—including developers, leaderboards, and integration endpoints—through associations to GamificationDashboard, Developer, and a set of GitHostingProvider instances. The AI mentor is also associated with the workspace, supporting users by providing guidance and feedback. Through the common GitHostingProviderEntity base class, platform-specific logic (GitHub, GitLab) is isolated while shared entities—Repository, PullRequest, Issue, Comment—remain available via one unified interface. This decoupling enables extensibility (e.g., Bitbucket, Azure DevOps) and eases maintenance. Furthermore, each platform entity includes standardized metadata fields (e.g., providerId, url, lastSynchedAt) to support synchronization workflows and allow flexible reporting and data handling across workspaces.

By combining configurable workspace options with a modular structure, Hephaestus will support multiple, parallel environments tailored to specific course phases, organizations, or student groups. This restructured model promotes autonomy, clarity, and scalability—all crucial for adoption within project-based learning ecosystems and enterprise-scale deployments.

4.2 Integrate GitLab

Given TUM's legal restrictions regarding GitHub, an essential objective of this thesis is the integration of GitLab as an alternative repository provider within Hephaestus. The absence of GitLab integration currently excludes Hephaestus from being utilized in certain educational contexts at TUM, limiting its full potential.

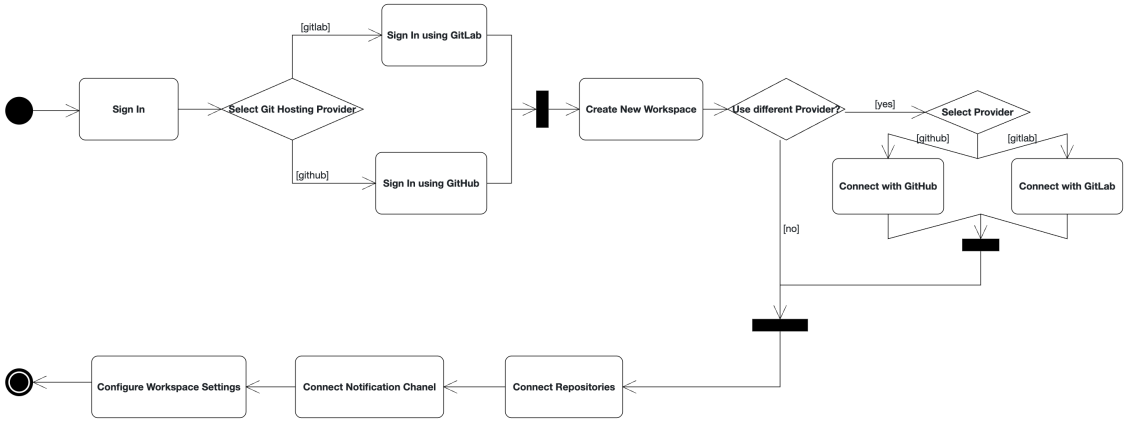

Figure 2: High-level activity diagram

This thesis addresses this issue by introducing GitLab integration, utilizing OAuth for secure authentication, and providing a streamlined and user-friendly repository selection process. The GitLab support is structured around an abstract provider entity (GitHostingProvider), enabling educational institutions to dynamically select repositories from either GitLab or GitHub, depending on institutional policies or preferences. This integration will significantly broaden Hephaestus' applicability within educational scenarios and allow educators to easily configure the platform to their teaching requirements.

Figure 2 illustrates the improved workspace setup flow, from selecting a Git provider to configuring repositories, notifications, and leaderboard settings. This streamlined process ensures ease of use and adaptability, crucial for effective educational integration.

5 Schedule

The thesis will begin on April 28th, 2025, and is expected to be completed by August 15, 2025. lasting 16 weeks, divided into four agile iterations:

Iteration 1 (Weeks 1–3)

- Implement dynamic workspace creation through user interface
- Integrate backend persistence of workspace entities

Iteration 2 (Weeks 4–6)

- Add UI for Git hosting provider selection and connection
- Refactor configuration handling (replacing static YAML config)

Iteration 3 (Weeks 7–8)

- Add leaderboard configuration settings (intervals, resets, channel bindings)
- Implement Slack notification channel connection
- Add support for multi-workspace distinction and isolation

Iteration 4 (Weeks 9–10)

- Add GitLab OAuth integration and repository retrieval
- Abstract platform logic via `GitHostingProviderEntity`
- Synchronize GitLab repositories and integrate webhook registration

Iteration 5 (Weeks 11–13)

- Perform full-stack validation of GitHub & GitLab integrations
- Extend error handling and validation mechanisms in workspace flows
- Enable dynamic Git provider addition to existing workspaces

Iteration 6 (Weeks 14–16)

- Finalize Slack leaderboard notification flow with mapping verification
- Conduct comprehensive integration tests with real-world setups
- Incorporate supervisor feedback and polish configuration UX

Bibliography

[PTR24] P. Patani, S. Tiwari, and S. S. Rathore, "The Impact of GitHub on Students' Learning and Engagement in a Software Engineering Course," *Computer Applications in Engineering Education*, vol. 32, no. 4, p. e22775, 2024, doi: 10.1002/cae.22775.

[Bol19] R. G. Bolivar-Valdivia, "Collaborative Learning Using Git with GitLab in Students of the Engineering Programming Course," in *Proceedings of the 2019 International Conference on Education and New Learning Technologies (EDULEARN19)*, 2019, pp. 2555–2564. [Online]. Available: http://ceur-ws.org/Vol-2555/paper8.pdf

[SSB22] D. Ståhl, K. Sandahl, and L. Buffoni, "An Eco-system Approach to Project-based Learning in Software Engineering Education," *IEEE Transactions on Education*, vol. 65, no. 4, pp. 514–523, 2022.

[VK17] J. Van Merriënboer and P. Kirschner, *Ten Steps to Complex Learning: A Systematic Approach to Four-component Instructional Design*. Routledge, 2017.

[Amb+10] S. A. Ambrose, M. W. Bridges, M. DiPietro, M. Lovett, and M. K. Norman, *How Learning Works: Seven Research-based Principles for Smart Teaching*. John Wiley & Sons, 2010.

[ZZ24] W. Zhang and L. Zhu, "A study of the impact of project-based learning on student learning outcomes," *Journal of Educational Research*, vol. 115, no. 3, pp. 248–258, 2024, [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC10411581/

[CIF20] R. A. Calvo, M. B. Ibáñez, and M. Freire, "Gamification in Software Engineering Education: A Systematic Literature Review," *Education and Information Technologies*, vol. 25, pp. 4615–4637, 2020, doi: 10.1007/s10639-020-10118-0.

Transparency in the use of AI tools

Throughout the development of this proposal, I used ChatGPT to expand ideas, search for relevant sources, and assist with grammar and style refinement. I thoroughly reviewed and edited all AI-generated content to ensure its accuracy, clarity, and relevance to the thesis topic. Additionally, I used tools such as Grammarly to check grammar and improve readability.