



Semesterarbeit Einreichungsformular / Klasse ITCNE23 – 4. Semesterarbeit

Im Folgenden beschreibt der Studierende das geplante Thema seiner Zertifikatsarbeit. Der Studierende orientiert sich dabei an dem Bewertungsraster. Der begleitende Dozent entscheidet, ob es sich dabei um ein geeignetes Thema handelt und fügt seine Kommentare und Überlegungen hinzu.

Bitte die *kursiven Textteile* durch die konkreten Angaben ersetzen.

Name und Vorname des Studierenden

Dany Ambühl

Titel der Semesterarbeit

Continuous Code Quality check

Zu behandelnde Themenfelder / Module (bitte ankreuzen)

Pkt.	Themenfeld / Modul	
1.	Projektmanagement	Pflicht
2.	DevOps*	Pflicht
3.	CNC - Cloud-native Core*	Pflicht
Themen aus dem 3. Semester		
4.	Relationale Datenbanken	<input type="checkbox"/>
5.	NoSQL, Key/Value, Time Series	<input type="checkbox"/>
6.	Microservices mit Python	<input type="checkbox"/>
7.	Datensicherheit / Datenschutz	<input type="checkbox"/>
8.	Netzwerk	<input type="checkbox"/>

* Als Infrastruktur wird Kubernetes mit mindestens einer CI/CD Pipeline erwartet.

Kurzbeschreibung des Zertifikatsarbeit Themas (1 bis max. 2 Seiten)

Problemstellung / Ausgangslage / Potential der Semesterarbeit

Die Semesterarbeit befasst sich mit dem Problem der Qualitätskontrolle von Ansible-Playbooks und der Standardisierung von YAML-Dateien.

Das Ziel ist es, ein System zu entwickeln, das diese Prozesse automatisiert und vereinfacht, um Zeit zu sparen, Standards zu setzen und Fehler zu reduzieren.

Zielsetzung der Semesterarbeit

Implementierung einer Open-Source-Lösung für Codequalitätsprüfungen:

Innerhalb des ersten Sprints soll eine Open-Source-Lösung auf einem Kubernetes-Cluster installiert und eingerichtet werden, um eine Überprüfung und Analyse der Codequalität von Ansible-Playbooks auf einem Git Repository zu ermöglichen.

Anbindung an das Git-Repository und Einrichtung einer CI/CD-Pipeline:

Bis zum Ende des zweiten Sprints wird eine Pipeline entwickelt, welche die Ansible-Playbooks im Git-Repository überprüft. Bei jedem neuen Merge Request in den «Main» Branch wird diese Pipeline automatisierte Qualitätschecks durchführen und die Ergebnisse dokumentieren.

Automatisierte Syntax-Überprüfung und Ergebnispräsentation:

Das System muss bis zum letzten Sprint eine Funktion integrieren, die die YAML- und Ansible-Syntax der Playbooks automatisiert überprüft und die Ergebnisse in einer Benutzeroberfläche darstellt.

Integration eines Formatierungstools für YAML-Dateien:

Bis zum letzten Sprint, soll ein Tool zur Automatisierung der Formatierung von YAML-Dateien implementiert und in die Entwicklungsumgebung integriert werden, um den Entwicklern die Einhaltung festgelegter Formatierungsstandards zu erleichtern.

Terminplan mit den wesentlichen Arbeitsschritten

Datum	Aktivität
21.10.2024	Abgabe und Besprechung Einreichungsformular Semesterarbeit
08.11.2024	Ergebnis 1. Sprint
02.12.2024	Ergebnis 2. Sprint
06.01.2025	Ergebnis 3. Sprint
29.01.2025	Abgabe der Arbeit / Abnahme

Sachmittel / Rahmenbedingungen

Hier werden jegliche benötigten Sachmittel aufgeführt, welche für das Erstellen der Ergebnisse des Projektes benötigt werden.

- CI/CD Pipeline
- Kubernetes Distributor
- Ansible
- Ansible-lint

Vorgaben, Methoden und Werkzeuge

Hier wird festgelegt, welche Methoden und Werkzeuge im Projekt eingesetzt werden und welche Vorgaben innerhalb des Projektes einzuhalten sind.

- Lean
- Kanban
- Gitlab/GitHub
- Visual Studio Code
- Draw.io

Risiken

Welche Risiken sind zu erwarten. Wie hoch sind diese und wie können diese Pro-aktiv verhindert werden.

Stärken:

- *Einsatz von Open-Source-Plattformen und Kubernetes, die eine robuste und erprobte Basis für die Durchführung von Codequalitätsprüfungen bieten.*
- *Automatisierung der Überprüfung der Codequalität fördert die Einhaltung der Qualitätsstandards.*
- *Integration von Tools zur Formatierung und Standardisierung von YAML-Dateien, um die Qualität und Lesbarkeit des Codes zu verbessern.*

Schwächen:

- *Risiko von Integrationsproblemen zwischen der CI/CD-Pipeline, der Codequalitätsplattform und dem Git-Repository.*
- *Mangelnde Erfahrung bei der Handhabung von spezifischen Code-Qualitätsanalysen für Ansible Playbooks und YAML-Syntax könnte zu Verzögerungen führen.*

Chancen:

- *Erhöhung der Qualität und Zuverlässigkeit von Ansible Playbooks Komponenten.*
- *Möglichkeit, durch die erfolgreiche Implementierung Referenzen für zukünftige berufliche Chancen zu schaffen.*

Bedrohungen:

- *Konstante Aktualisierung und Wartung der Open-Source-Tools können zu erhöhtem Aufwand führen.*
- *Widerstand gegen Änderungen und neue Prozesse bei den Entwicklern.*

Strategien, um die Schwächen der SWOT-Analyse zu adressieren:

- *Pflege von guten Beziehungen zur Entwickler-gemeinschaft, um Support und Ratschläge zu geben.*
- *Erarbeitung und Verfeinerung der Qualitätsprüfungsmethoden in einem Pilotprojekt, bevor es vollständig eingeführt wird.*

Entscheid des begleitenden Dozenten

Bitte ankreuzen

- Genehmigt ☐
- Zu verbessern ☐
- Abgelehnt ☐

Begleitender Dozent

Ort und Datum: 18.10.2024 _____

Name & Unterschrift: Dany Ambühl _____

Beurteilung des Antrages

Kriterium	Kommentare	Erfüllt
Anforderungen an die Form (Strukturierung) der Semesterarbeit		
Problemstellung		<input type="checkbox"/>
Ziele (mindestens drei!)		<input type="checkbox"/>
Risiken bezogen auf Zertifikatsarbeit		<input type="checkbox"/>
Themenfelder (mindestens zwei!)		<input type="checkbox"/>
Anforderungen an Qualität der Semesterarbeit		
Machbar		<input type="checkbox"/>
Praxisnah		<input type="checkbox"/>
Herausfordernd		<input type="checkbox"/>
Lehrgangsbezug		<input type="checkbox"/>

Damit die Semesterarbeit angenommen wird, müssen alle Kriterien erfüllt sein.

Die Kommunikation wird über den jeweiligen Teams-Kanal geführt.