## Neues Projekt-Team:

|  |  |  |
| --- | --- | --- |
| Abgabedatum : | Klasse / Semester: | Team: **(Neu zusammenstellen! Max. 2 Lernende!)** |

# Angaben zur Durchführung des Projektes

Die Leistungsbeurteilung 2 erfolgt in Form einer Projektarbeit mit 3 Meilensteinen. Dafür ist ein Zeitraum von 4+ Wochen vorgesehen.

Jedes Team wählt eine ***Automatisierungsaufgabe*** *aus*oder ***definiert eine eigene***.

***Meilenstein A*** besteht darin, anhand der Rahmenbedingungen (und evtl. nach Machbarkeitsabklärungen) *eine* ***konkrete Anforderungsdefinition*** *auszuarbeiten* und ***mit der Lehrperson zu besprechen****.* Das **Lösungsdesign** ist mit UML Aktivitätsdiagram(-men) zu dokumentieren. \*

***Meilenstein B*** beinhaltet eine **eigene Implementierung** des Lösungsdesigns (oder Teile davon). Die Abnahme des Projektes erfolgt anhand einer **Programmdemonstration** und wird mittels eines formalen **Code-Reviews** begutachtet und bewertet.

***Meilenstein C*** Alle Dokumente sind **fortlaufend auf einem GitLab-Repository** (Branch) zu speichern: Quellcode mit Inline-Kommentar gemäss TBZ-Konvention, UML AD, Installationsanleitung (Setup), Bedienungsanleitung, etc.

\*) Als Alternative kann anstelle des Lösungsdesigns, fertiggestellter Quellcode mit UML Aktivitätsdiagram(-men) dokumentiert werden.

## *Notenberechnung*

Bewertet werden nur ausgewiesene Eigenleistungen gemäss Vorgaben, welche aus selbstverfasster Dokumentation und aus der Produktion des Automations-Scriptes bestehen. Erworbene Erkenntnisse aus Ressourcen Dritter muss belegt und ausgewiesen werden.

Entsprechend des Bewertungsrasters werden Notenpunkte für Team-Aufgaben und für individuelle Einzel-Aufgaben vergeben.

## *Umgebung*

(Virtueller) Computer mit Betriebssystem (Linux) und Scriptumgebung (BASH)và Automations-Script  
Draw.io, StarUML o.ä. à Darstellung der Ablaufstruktur  
GitLAB o.ä. à Ablage Code und zus. Dokumentation.

## *Dokumente zu dieser LB*

Siehe GitLAB m122 [hier](https://gitlab.com/-/ide/project/ch-tbz-it/Stud/m122/tree/main/-/10_Projekte_LB2/)!

# Grobe Projektplanung

Integrieren Sie folgende Meilensteine in ihre Planung:

|  |  |  |  |
| --- | --- | --- | --- |
| *Meilenstein* | *Tätigkeit / Abgabe* | *Abgabetermin  (Richtwerte)* | *Bestätigung LP* |
| MS ATeam | Team Bildung, Kollaborationsplattform, GitHub Repos \*, LehrerzugangWahl / Ausarbeitung der Aufgabenstellung  * Abnahme Anforderungsdefinition  Erstellen des Repositorys\* auf GitLAB à LP hat Zugang | Woche 1 (Doppelte Wertung) |  |
| MS BEinzel | Erstellen des eigenen Automatisierungs-Scriptes Projektabnahme: Projektdemo mit mit Code-Review | Aufgebot |  |
| MS C Einzel | Abgabe kommentierter Programmcode (Branch)  * Abgabe Projektdoku und UML Aktivitätsdiagramm | Projekt Ende |  |

\*) Öffentliche GitLab-URLs LP zukommen lassen.

Namenskonvention URL: **M122\_Klasse\_Thema\_Name\_Name**

## Kompetenzraster: Namen: \_1)\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_2)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Werden die mittig beschriebenen Kompetenzindikatoren erreicht, so ergibt dies die Note 4. Pro Kriterium kann je nach Erfüllungsgrad entweder die maximal 0.2 Notenpunkte abgezogen bzw. dazu gezählt werden.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Kompetenzfeld Einzelaufgaben** | **max. - 0.2** | **4** | **max. + 0.2** | **N 1)** | **N 2)** |
| **MS A:**  Anforderungs-definition   *(Zählt doppelt)* | c Keine oder unklare Formulierung vorhanden  c Aufgabe ist simpel | c Die Anforderungsdefinition beschreibt eine **Automatisierungsaufgabe** des Betriebssystems. (MUSS + KANN) | c Die Anforderungsdefinition ist **ausführlich** ausformuliert (**Prosa**)  c Eingabe- und Ausgabedaten, sowie deren Formate sind beschrieben. | **2 x  \_** |  |
| **MS A (**oder **C):**  UML Aktivitätsdiagramm | cc UML Aktivitätsdiagrammhat formale Fehler oder fehlt.  cc Ablauf ist unklar oder falsch. | cc Der zu erwartende **Ablauf für Benutzer** ist graphisch korrekt dargestellt *(oder*  cc Ein sinnvoller **Ablauf im Code** ist graphisch korrekt dargestellt.) | cc Das **UML Aktivitätsdiagramm** ist graphisch korrekt und **grossumfänglich** dargestellt | \_ | \_ |
| **MS B:** Implementation [IhrSystem]  *(Quellcode)* | cc **Verarbeitungsteil** unlogisch oder fehlerhaft  cc ungewollte oder fehlende Fehlermeldungen | cc **Verarbeitungsteil** gemäss Anforderungsdefinition **funktionsfähig**  cc und **robust** | cc Cronjob eingerichtet und funktionstüchtig.  cc Code mit **Funktionen** zusammengefasst / abstrahiert | \_ | \_ |
| **MS B:** Input/Output [IhrSystem]  *(Quellcode)* | cc Rudimentäre Funktionen | cc **Anfrage** an ein **Kundendienst** funktionsfähig  cc **Weitergabe** an ein **Kundendienst** funktionsfähig | cc **Anfrage** an ein **Kundenserver** funktionsfähig  cc **Weitergabe** an ein **Kundenserver** funktionsfähig | \_ | \_ |
| **MS B:** Konfiguration/Logging [IhrSystem]  *(Quellcode)* | **c**cKeine Variablenliste  **c Abzug Teamarbeit \*  (-1.0)** | cc Konfiguration im Script (via Variablen)  cc Logging / Fehlermeldung durch Konsolenausgabe | cc **Konfiguration via ext. Datei**  cc **Logging** in Datei  cc **Mailing** an Drittperson | \_   - | \_ |
| **MS B:**  Abgabe max. 10min & Komplexität | cc Keine, abgebrochene oder fehlerhafte Demo ohne Ursachenerkenntnis  cc Programm unfertig, keine Entwicklungs-History | cc **Vorbereitete Demo erfolgt**  cc **Formaler Testbeschrieb** für Code-Review vorbereitet und (nach Code-Review) korrekt ausgefüllt. | cc Lernender kann sein Programm mit **weiterführender Erkenntnis verbinden** / kommentieren.  (Integration / Sicherheit)  cc **Komplexes** Projekt erfolgreich umgesetzt. (Mehrere Scripte, WEB-API, mehrstufig, ...) | \_ | \_ |
| **MS C:**  Inline-Dokumentation | cc Quellcode unklar kommentiert, Variablennamen nicht sprechend  cc Codestruktur uneinheitlich / nicht eingerückt | cc Quellcode **kommentiert** (Wichtige Funktionen / Aufrufe von CL-Tools)  cc Struktur korrekt **eingerückt** | cc Quellcode **ausführlich** und verständlich kommentiert  (Header, Abschnitte, spez. Funktionen) | \_ | \_ |
| **MS C:**  Dokumentation der Implementation | cc Keine oder unklare Doku vorhanden  cc Kein(e) oder unklare(r) Testbericht / Testdaten | cc **Einfache Beschreibung** des Projekts(README.md) (Setup & Bedienung) | **Ausführliche Beschreibung**  cc der internen Funktionen (für Entwickler)  cc und der **Installation** (Admin). | **\_** | **\_** |
| **Formale Aspekte** | cc MS A zu spät ..  cc MS B/C zu spät .. cc Kein GitLab-Link ... abgegeben | cc Abgabetermine **eingehalten**  cc **Regelmässiges Update** auf GitLab | cc **Bonus** für: - Projektablauf protokolliert - Zeitbonus  - Engagement - ... | \_ | \_ |
| ***Total Einzel Note*** *(auf Zehntel gerundet)* | | | **=** | | **=** | |
| ***Bemerkungen Expert/in:***  *Bei der Beurteilung ist das Nichterreichen der minimalen Anforderungen (Note < 4) zu begründen.*  *\*) Wenn nur ein gemeinsames Programm abgegeben wird!* | | | | |  | |