

Hochschule Bremerhaven

Fachbereich II – Management und Informationssysteme

**Informatik**

Modul Rechnerarchitektur

## **Protokoll zu Aufgabenblatt Nr.01**

von

**Fabian Tober** Matrikel-Nr. 40607  
**Benjamin Schill** Matrikel-Nr. 40671

05.04.2024

# Inhaltsverzeichnis

|          |                                                                 |          |
|----------|-----------------------------------------------------------------|----------|
| <b>1</b> | <b>Einleitung</b>                                               | <b>3</b> |
| 1.1      | Material und Methode . . . . .                                  | 3        |
| 1.1.1    | Gegeben . . . . .                                               | 3        |
| 1.1.2    | Gesucht . . . . .                                               | 3        |
| 1.1.3    | Geplante Durchführung und Erwartung . . . . .                   | 3        |
| <b>2</b> | <b>Aufgabe 01 - Sicherheitseinweisung Labor</b>                 | <b>3</b> |
| <b>3</b> | <b>Aufgabe 06 - Text Zusammenfassung</b>                        | <b>4</b> |
| 3.1      | Fassen Sie den Text mit Ihren eigenen Worten zusammen . . . . . | 4        |
| 3.2      | Schreiben Sie drei Beispiel-Commit-Nachrichten . . . . .        | 4        |
| <b>4</b> | <b>Aufgabe 07 - git-Befehle</b>                                 | <b>4</b> |
| <b>5</b> | <b>Aufgabe 08 - weiterer git-Befehl</b>                         | <b>5</b> |
| <b>6</b> | <b>Aufgabe 09 - Video Notizen</b>                               | <b>5</b> |
| <b>7</b> | <b>Ergebnis</b>                                                 | <b>6</b> |
| <b>8</b> | <b>Aufgaben Zusammenfassung</b>                                 | <b>6</b> |
| 8.1      | Aufgabe 02 - Projektarchiv . . . . .                            | 6        |
| 8.2      | Aufgabe 03 - Arbeiten mit git . . . . .                         | 6        |
| 8.3      | Aufgabe 04 - Musterprotokoll . . . . .                          | 7        |
| 8.4      | Aufgabe 05 - Musterprotokoll im Projektarchiv . . . . .         | 7        |
| 8.5      | Aufgabe 10 - Protokoll Erstellung . . . . .                     | 7        |

# 1 Einleitung

Dieses Versuchsprotokoll dient der Dokumentation der Bearbeitung von Aufgabenblatt01, welches die Anwendung von Git-Befehlen sowie eine Sicherheitseinweisung ins Labor umfasste. Im Folgenden werden die Aufgaben abgearbeitet und am Ende zu allen Aufgaben, wo es keine eigene Aufgabenstellung zugab (die hier explizit genannt werden muss) kurz zusammengefasst.

## 1.1 Material und Methode

Zur Verfügung stehen uns Laptops (oder Laborrechner), ein Server (Hopper), sowie das Internet. Das Ziel erreichen wir ebenfalls nur durch Nutzung unserer gegebenen Materialien.

### 1.1.1 Gegeben

Wir können unser eigenes Wissen und/oder das Internet benutzen, um die Aufgaben abzuschließen.

### 1.1.2 Gesucht

Unser Ziel besteht darin, den Aufgabenzettel 01 vollständig zu bearbeiten, indem wir alle Fragen umfassend beantworten.

### 1.1.3 Geplante Durchführung und Erwartung

Um die Aufgaben zu lösen, haben wir uns die Methoden auf dem Aufgabenblatt 01 zu nutzen gemacht. Wir gehen davon aus, dass die Aufgaben erfolgreich bearbeitet werden.

## 2 Aufgabe 01 - Sicherheitseinweisung Labor

Im Labor wurde eine Sicherheitseinweisung durchgeführt, die die grundlegenden Sicherheitsrichtlinien und Verhaltensweisen im Laborumfeld vermittelte. Dabei wurden folgende Stichpunkte von uns festgehalten:

- Im Falle eines Feuers: Verwendung des Feuerlöschers und/oder Anlegen von feuerfesten Handschuhen, um Objekte aus dem Labor zu tragen und draußen sicher fallenzulassen.
- Bei Verletzungen: Sofortige Meldung und Eintragung ins Verbandsbuch; Versicherung über die Hochschule.
- Trinken nur außerhalb der Labortische erlaubt, Essen nur außerhalb des Labors gestattet.
- Notausgänge: Außenfassade oder Tür ins Gebäude

- Treffpunkt bei Evakuierung -> Wiese an der Geeste.

### 3 Aufgabe 06 - Text Zusammenfassung

#### 3.1 Fassen Sie den Text mit Ihren eigenen Worten zusammen

Git-Commit-Nachrichten sind von zentraler Bedeutung für die Kommunikation innerhalb, von Entwicklerteams und fungieren als kontinuierliche Dokumentation der Codebasis. Es existieren bestimmte Richtlinien für das Verfassen dieser Nachrichten. Der Betreff sollte maximal 50 Zeichen umfassen und im Imperativ verfasst sein. Im Anschluss an den Betreff sollte eine präzise Beschreibung des Commits folgen, die die grundlegenden Fragen "Warum?", "Was genau?" und "Gibt es etwaige Probleme?" beantwortet. Die Beschreibung kann durchaus ausführlicher sein, sollte jedoch stets präzise sein und keine unwichtigen Informationen enthalten.

#### 3.2 Schreiben Sie drei Beispiel-Commit-Nachrichten

- Translation Fixes - Änderung in der Übersetzung auf Englisch durch umschreiben eines Satzes in der "Impressum-en.html"-datei.
- Bug Fixes - Fehler in der Datei "player.dll", Spieler Eigenschaften nun wieder richtig.
- Namensänderung Protokolldatei - Autor Namensänderung in der Protokolldatei.

### 4 Aufgabe 07 - git-Befehle

Es folgt nun eine Aufzählung der git-Befehle auf dem Arbeitsblatt 01 mit einer Erklärung der Funktionen.

- **git fetch:** Dieser Befehl holt alle Änderungen aus dem Remote-Repository, speichert sie jedoch nicht auf deinem lokalen Arbeitsbereich. Es ist nützlich, um zu sehen, was sich im Remote-Repository geändert hat, bevor du deine eigenen Änderungen integrierst.
- **git pull:** Der Befehl git pull holt ebenfalls Änderungen aus dem Remote-Repository, führt jedoch auch automatisch ein Merge durch, um die heruntergeladenen Änderungen in deinen lokalen Arbeitsbereich zu integrieren. Kurz gesagt, git pull ist eine Kombination aus git fetch und git merge.
- **git add:** Mit diesem Befehl fügst du Änderungen deiner Dateien dem sogenannten Staging-Bereich hinzu. Der Staging-Bereich ist eine Zwischenstufe, in der du festlegen kannst, welche Änderungen du für den nächsten Commit vorbereiten möchtest.

- **git commit:** Nachdem du Änderungen in den Staging-Bereich verschoben hast, kannst du mit `git commit` einen Snapshot dieser Änderungen erstellen. Ein Commit speichert deine Änderungen dauerhaft im lokalen Repository und ermöglicht es dir, eine Nachricht anzugeben, die beschreibt, was du geändert hast.
- **git push:** Dieser Befehl überträgt deine lokalen Commits auf das Remote-Repository, wodurch sie für andere Teammitglieder verfügbar werden. Es ist wichtig zu beachten, dass du vor dem Pushen normalerweise sicherstellen solltest, dass du die neuesten Änderungen aus dem Remote-Repository abgerufen und eventuell Konflikte gelöst hast.
- **git help:** Mit diesem Befehl kannst du Hilfe zu bestimmten Git-Befehlen erhalten. Zum Beispiel kannst du `git help fetch` verwenden, um die Dokumentation speziell für den `fetch`-Befehl aufzurufen.
- (git-scm.com Video) **git init:** Initialisieren eines neuen Git-Repositories.

## 5 Aufgabe 08 - weiterer git-Befehl

- **git request-pull:** Dieser Befehl wird normalerweise von Personen verwendet, die Änderungen an einem Repository vorgenommen haben und diese Änderungen dem Besitzer des Hauptrepositorys zur Überprüfung und möglichen Integration vorschlagen möchten. Es wird verwendet, um eine Anfrage um Zusammenführung (Pull-Request) zu erstellen. Dieser Befehl ist in Revision einordbar.
- **Everyday:** Nicht üblich, wird nicht täglich verwendet.
- **Revision:** Hilft bei der Anforderung und Zusammenfassung von Änderungen zwischen verschiedenen Revisionen oder Branches.
- **Workflows:** Wird in spezifischen Workflow-Szenarien wie Forks und Pull-Requests verwendet, um Änderungen zwischen verschiedenen Repositories zu verwalten und zu integrieren.

## 6 Aufgabe 09 - Video Notizen

- **Video 01:** Das erste Video betont die Vorteile von Git in einem Unternehmenskontext, insbesondere für Gruppenarbeit. Git erleichtert die Erstellung, Bearbeitung und Speicherung von Dateien und ermöglicht es Teams, effizient zusammenzuarbeiten. Es bietet einfache Kontrolle über verschiedene Versionen eines Projekts und speichert Zwischenschritte, was die Zusammenarbeit erleichtert.
- **Video 02:** Das zweite Video stellt grundlegende Git-Befehle vor, die für die Zusammenarbeit in Projekten wichtig sind, wie `"git init"`, `"git add"`, `"git commit"` und

”git push”. Es skizziert auch die Rollen in einem Git-Projekt, einschließlich Entwicklern, Projektmanagern, Reviewern und Mitwirkenden, und betont die Bedeutung einer klaren Rollenverteilung für eine effiziente Zusammenarbeit.

- **Video 03:** Im dritten Video ging es um die Installation von Git. Außerdem ging es um die erst Einrichtung von Git und wie man projekte erstellt und committed. Zu letzt wurde noch gesagt wo man Dokumentation und andere Hilfreiche dinge erfährt für den Umgang mit Git.
- **Video 04:** Im vierten Video liegt der Fokus auf der Inhaltsverwaltung mit Git. Hier wird gezeigt, wie Dateien eigenständig hochgeladen werden können, ohne auf automatische Uploads angewiesen zu sein. Diese Funktion ermöglicht es den Benutzern, gezielt Dateien auszuwählen und sie in das Repository hochzuladen, anstatt den gesamten Inhalt automatisch zu synchronisieren. Dies ist besonders nützlich, wenn nur bestimmte Änderungen oder Dateien aufgezeichnet und geteilt werden sollen, während andere vorläufig lokal bleiben sollen. Damit bietet Git eine flexible und präzise Methode zur Verwaltung von Inhalten in Repositories. Es wird auch darauf hingewiesen, dass Git soviel wie möglich lokal behält.

## 7 Ergebnis

Die Aufgaben wurden alle bearbeitet und somit ist das Ergebnis positiv. Des Weiteren sind wir nicht von der geplanten Durchführung abgewichen.

## 8 Aufgaben Zusammenfassung

Im Folgenden wird es eine Zusammenfassung zu den Aufgaben 02-05 und 10 geben.

### 8.1 Aufgabe 02 - Projektarchiv

In dieser Aufgabe ging es darum, eine gemeinsam nutzbare Oberfläche zu erstellen, um die Teamarbeit zu erleichtern. Jedes Teammitglied konnte Änderungen lokal auf seinem Rechner durchführen und anschließend zum Gesamtprojekt hochladen, um die Fortschritte zu synchronisieren.

### 8.2 Aufgabe 03 - Arbeiten mit git

In dieser Phase haben wir grundlegende Schritte mit Git unternommen und die essenziellen Kenntnisse für den Umgang damit erworben. Dazu gehörten das Wechseln von Verzeichnissen und das Erstellen von Dateien, um die Grundlagen der Versionsverwaltung zu verstehen.

### **8.3 Aufgabe 04 - Musterprotokoll**

Während dieser Aufgabe haben wir uns mit dem Musterprotokoll vertraut gemacht und es zum ersten Mal ausgefüllt. Darüber hinaus hat ein Teammitglied das Projekt heruntergeladen und sich mit dem Umgang mit Git vertraut gemacht, um eine effiziente Zusammenarbeit zu gewährleisten.

### **8.4 Aufgabe 05 - Musterprotokoll im Projektarchiv**

Wir haben ein Verzeichnis erstellt, um das aktuelle Versuchsprotokoll ordnungsgemäß abzugeben, und dabei den Umgang mit Git sowie die erforderliche Abgabeform trainiert und angewendet.

### **8.5 Aufgabe 10 - Protokoll Erstellung**

Aufgabe 10 ist die Erstellung dieses Protokolls.