

# Projekt «Soccer Table»

BBZW, Modul 426

---

Patrick Bucher

30.05.2024

- Es handelt sich um ein **agiles** Projekt, d.h. Sie erhalten keine schriftlichen Anforderungen, sondern müssen diese selber durch Nachfragen beim Auftraggeber (Lehrperson) erarbeiten.
- Sie arbeiten in Ihrem Scrum-Team, das sich zu Beginn des Semesters zusammengefunden hat.
- Für Aufträge, die zur Benotung zählen, erhalten Sie Bewertungskriterien.
- Die vorliegenden Folien werden laufend aktualisiert.

**Wichtig:** Lösen Sie das *gestellte* Problem; versuchen Sie nicht jemanden zu beeindrucken.

## Repository

[m426/soccer-results](#)

## Inhalt: Verzeichnisse (Ligen) mit Dateien (Spieltage)

- bundesliga
  - day01.txt
    - Bayer Leverkusen 2:2 1. FC Heidenheim
    - Bayern München 1:0 VfL Wolfsburg
    - ...
  - day02.txt
- la-liga
  - ...
- ...

Die Daten können erweitert und ergänzt werden, das Format sollte aber bleiben.

- Für jede Liga soll die Ligatabelle berechnet und ausgegeben werden.

# Auftrag 1: Analyse

## Form

- Gruppenarbeit (im Scrum-Team)
- Zeit: 20 min.

## Aufgabe

1. Betrachten Sie sich die Daten im Repository und versuchen Sie die Struktur zu verstehen.
2. Überlegen Sie sich, wie das Ergebnis aussehen könnte.
3. Notieren Sie sich Fragen und stellen Sie diese anschliessend in der Q&A-Session.

## Hausaufgabe

Beschäftigen Sie sich weiter mit der Problemstellung und notieren Sie sich Fragen für die nächste Q&A-Session.

## Auftrag 2: User Stories

### Form

- Gruppenarbeit (im Scrum-Team)
- Zeit: 20 min.

### Aufgabe

1. Besprechen Sie im Team Ihre Notizen zum Projekt und dessen Anforderungen.
2. Formulieren Sie **drei** User Stories gemäss INVEST-Kriterien. (Die Stories müssen *nicht* den ganzen Funktionsumfang abdecken.)

### Hausaufgabe

Reichen Sie Ihre User Stories per Teams ein (Aufgabe folgt).

## Auftrag 2: User Stories (Bewertungsraster)

Für folgende Kriterien gibt pro Story die folgenden Punkte:

- [1P] Die Stories folgen der vorgegebenen Form (siehe Folien).
- [2P] Die Stories sind klar und verständlich formuliert.
- [3P] Die Stories folgen den *INVEST*-Kriterien. [6x0.5P]

D.h. pro Story gibt es max. 6 Punkte; für die drei geforderten Stories max. 18 Punkte.

## Auftrag 3: Umsetzung

Setzen Sie die Anwendung in Ihrem Scrum-Team um.

**Der genaue Abgabetermin wird noch kommuniziert**



## Auftrag 3: Funktionale Anforderungen

- Der Benutzer wählt einen Ligaordner aus, für welchen die Ligatabelle berechnet und ausgegeben wird.
  - Optional wählt der Benutzer, bis zu welchem Spieltag von Anfang an die Ligatabelle berechnet und ausgegeben werden soll.
  - Fehlt diese Angabe, wird die komplette Ligatabelle bestehend aus allen Spieltagen berechnet/ausgegeben.
- Die Ligatabelle enthält die Spalten: Rang (1 bis n), Name, Punkte, Anzahl Siege/Niederlagen/Unentschieden, erzielte/kassierte Tore, Tordifferenz mit aussagekräftigen Spaltenüberschriften.
- Die Ligatabelle ist nach Kriterien sortiert: 1) Punkte (absteigend), 2) Tordifferenz (absteigend), 3) Anzahl Siege (absteigend) und 4) Name (aufsteigend) sortiert.

## Auftrag 3: Nicht-Funktionale Anforderungen

- Der Quellcode wird in einem Git-Repository verwaltet; die Abgabe wird mit dem Tag `v1.0.0` markiert.
- Das Repository enthält eine Datei namens `README.md`, welche die Instruktionen enthält um Anwendung und Testfälle auszuführen.
- Die Kernlogik (Auswahl der Spieltage, Berechnung, Sortierung) wird mithilfe von Unit Tests automatisch getestet (signifikante Testabdeckung).
- Der Code ist unter den *Clean Code*-Aspekten Formatierung, Benennung, Kommentare, Wiederverwendbarkeit und Klarheit qualitativ hochwertig.
- Alle Teammitglieder haben einen Beitrag zu Code und/oder Dokumentation geleistet und Commits am Repository vorgenommen.

**Das Bewertungsraster folgt**