

SUPER SUPERHIRN

Endpräsentation der theoretischen Ergebnisse - Gruppe 11

Carl Weiß, 572802
Can Wrobel, 585031
Alexander Yurovskyy, 572782
Linda Zubenin, 584625
Rico Flemming, 584322

INHALT

1

Wie sieht unsere Architektur aus?

3

Wie wurde die Qualität
gewährleistet?

2

Nicht-funktionale Anforderungen

4

Fazit zum Projekt

ARCHITEKTURSTIL

Schichtenarchitektur

Vorteile

Hohe Kohäsion und geringe Kopplung

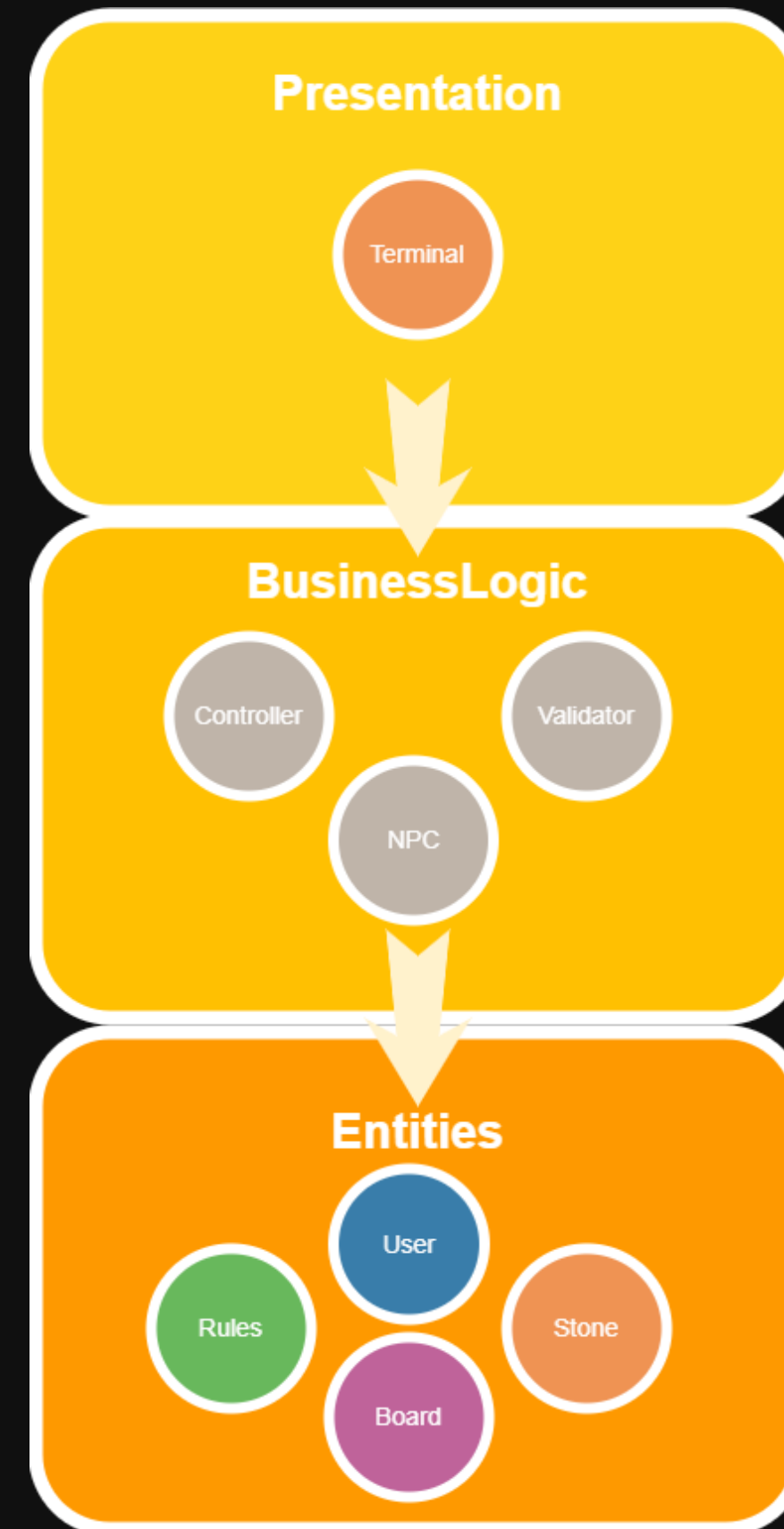
Gute Anpassbarkeit von bestehendem Code

Komplexität wird heruntergebrochen

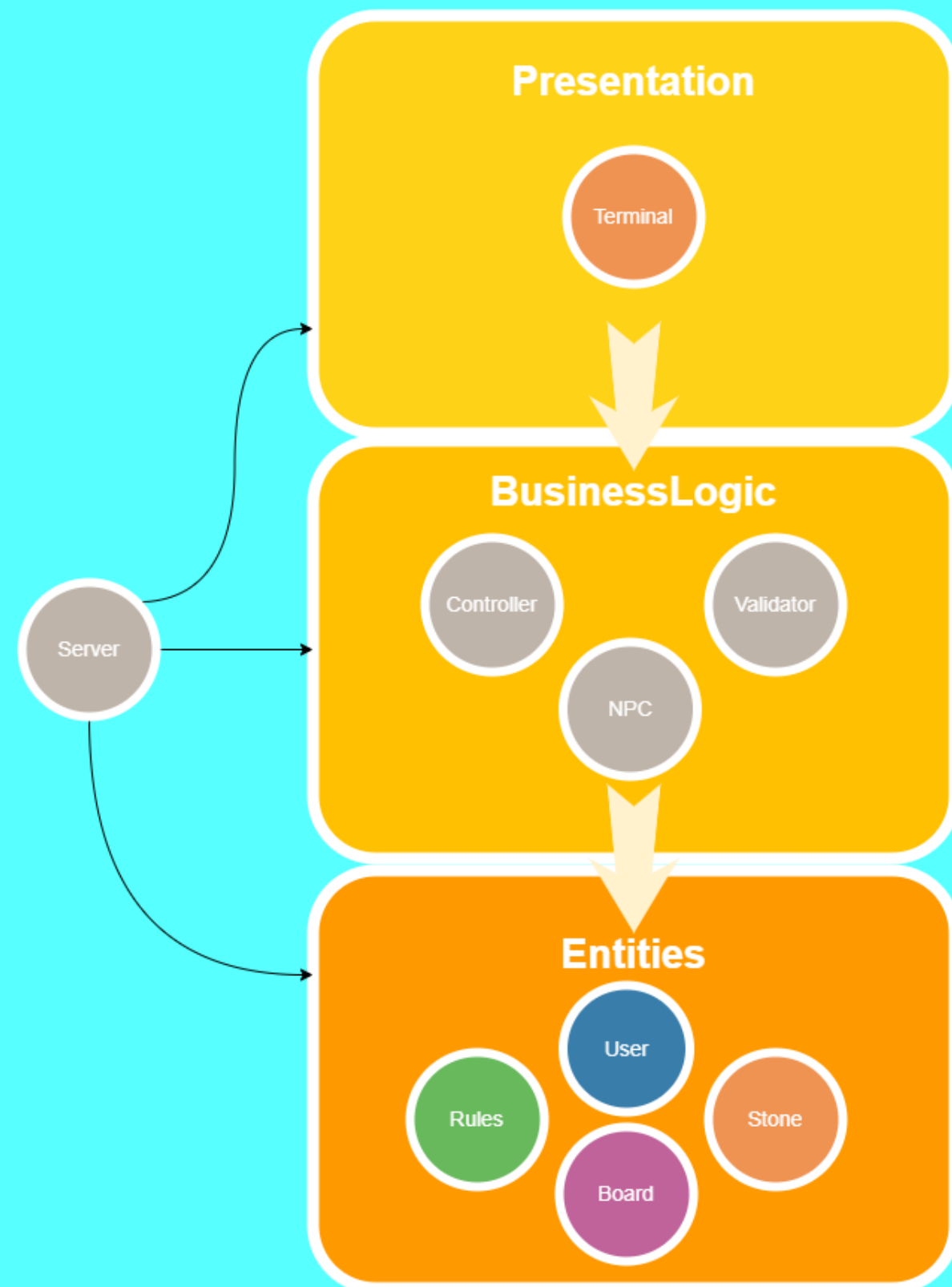
Nachteile

Strikte Einhaltung erfordert mehr Code

Neue Schichten sind aufwendig zu implementieren



WOHIN MIT DER SERVER-KOMPONENTE?



Elegant

Eigene Schicht für die Kommunikation einführen

Effizient

Bestehende Schicht nutzen und erweitern

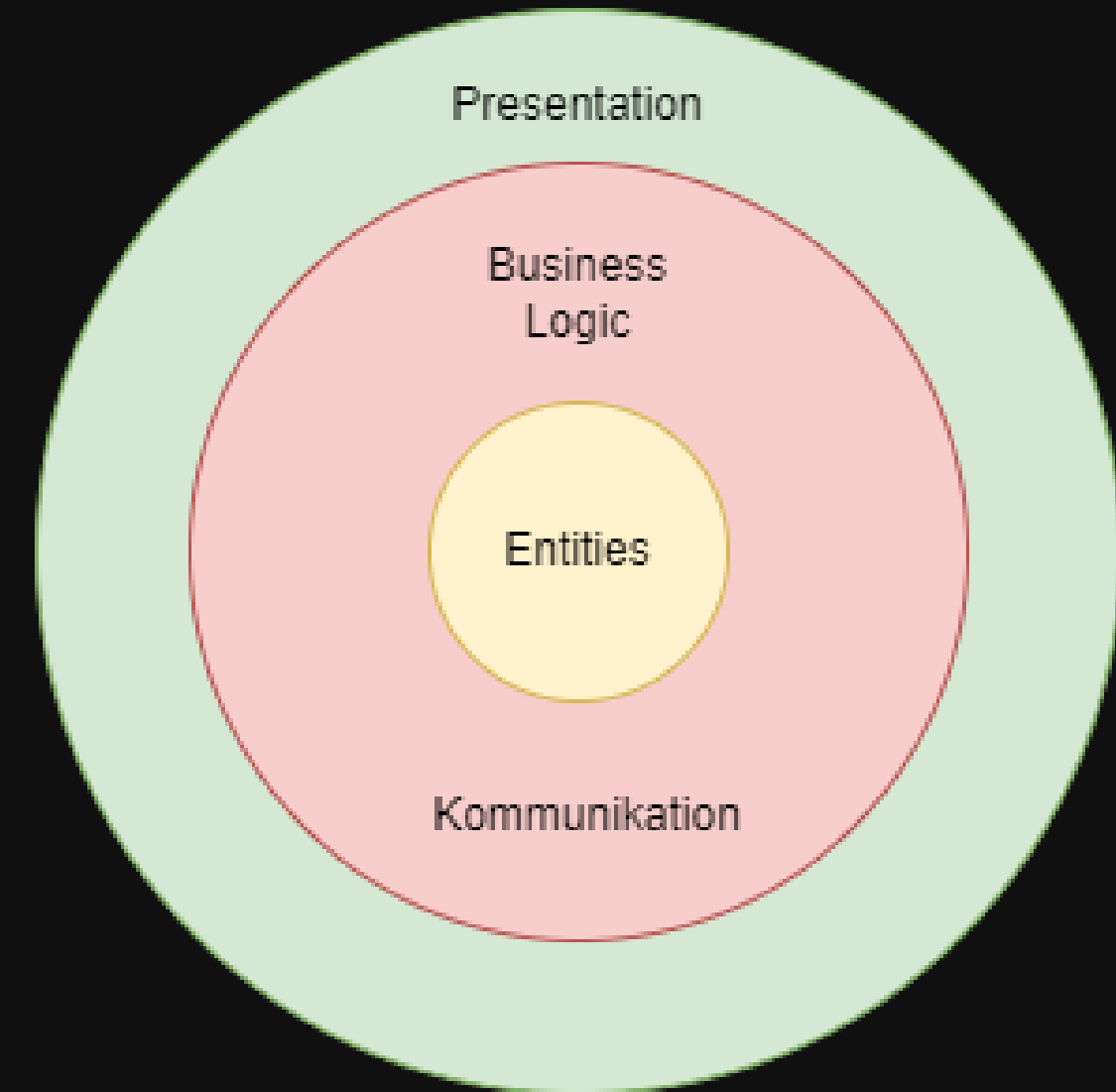
ARCHITEKTURSTIL

Zwiebelarchitektur

Vorteile

Kopplung geht nur in Richtung Zentrum

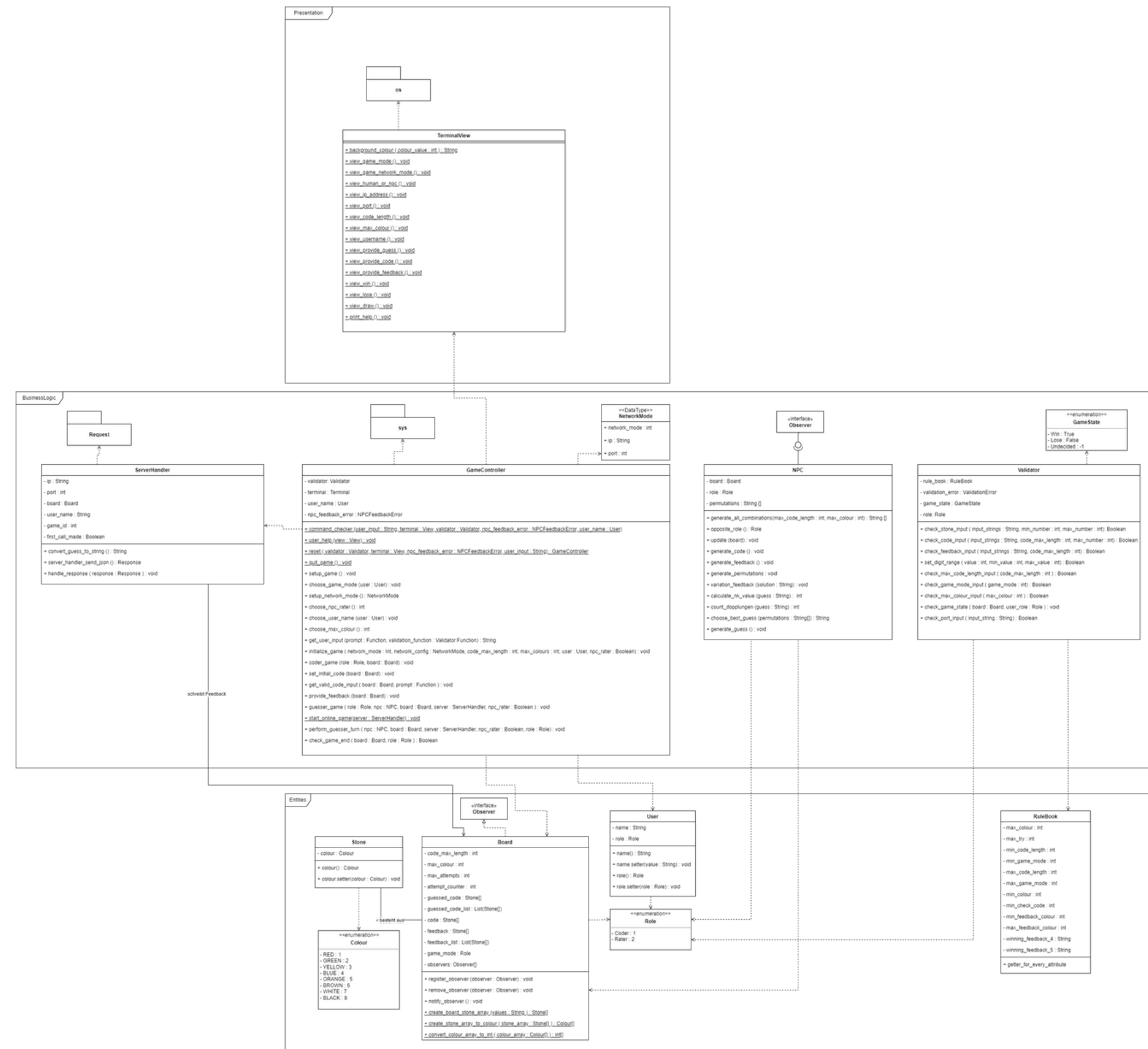
Erweiterbarkeit vereinfacht



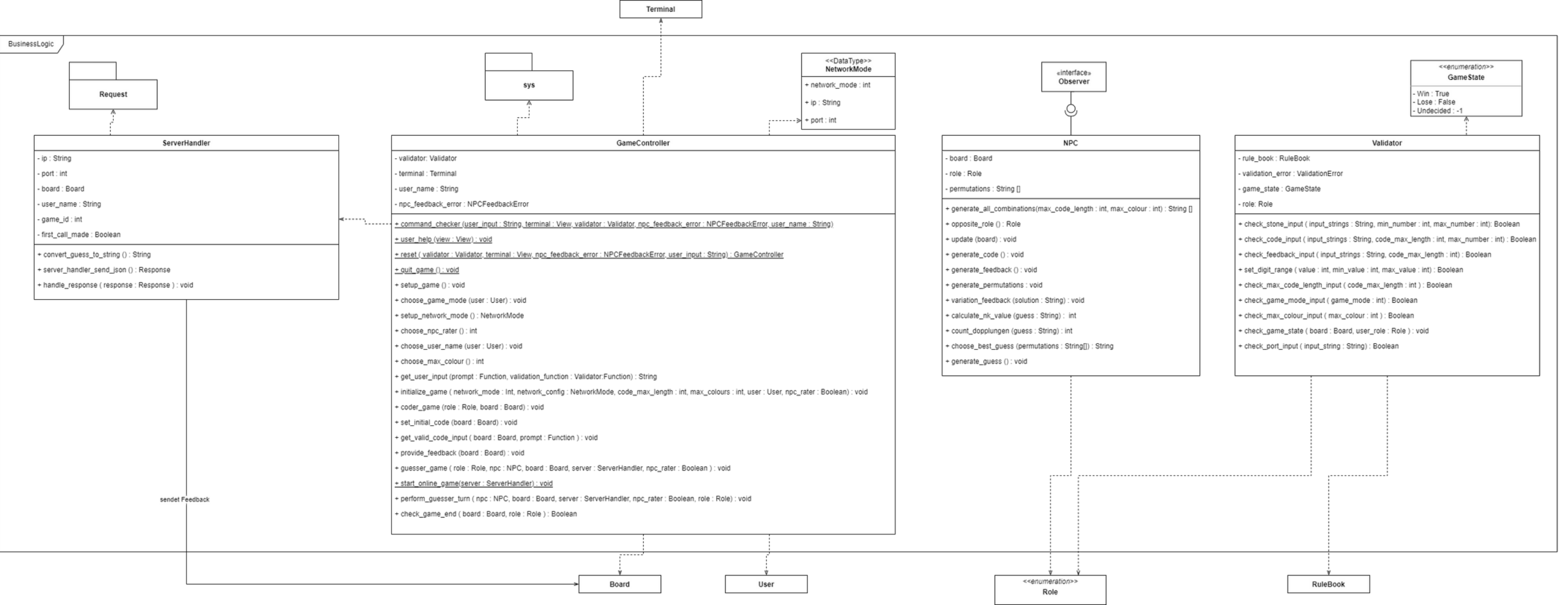
ERGEBNIS IMPLEMENTIERUNG

Keine Schichtübergreifenden
Zugriffe

Präsentation und Eingabe von
Entitäten getrennt



FEINARCHITEKTUR



```
Alex
Bitte wähle die Codelänge zwischen 4 und 5.
5
Bitte wähle die maximale Farbenanzahl zwischen 2 und 8.
8
Bitte gib deinen Code ein.
67458
Code: [6][7][4][5][8]
1. Rateversuch: [1][1][2][2][3]
Bitte gib dein Feedback

Code: [6][7][4][5][8]
1. Rateversuch: [1][1][2][2][3] 1. Feedback:
2. Rateversuch: [4][4][5][5][6]
Bitte gib dein Feedback
787
Code: [6][7][4][5][8]
1. Rateversuch: [1][1][2][2][3] 1. Feedback:
2. Rateversuch: [4][4][5][5][6] 2. Feedback: [7][8][7]
3. Rateversuch: [4][5][4][7][7]
Bitte gib dein Feedback
787
Code: [6][7][4][5][8]
1. Rateversuch: [1][1][2][2][3] 1. Feedback:
2. Rateversuch: [4][4][5][5][6] 2. Feedback: [7][8][7]
3. Rateversuch: [4][5][4][7][7] 3. Feedback: [7][8][7]
4. Rateversuch: [4][6][7][6][4]
Bitte gib dein Feedback
7|
```

ANSICHT DER GUI

NICHT FUNKTIONALE ANFORDERUNGEN

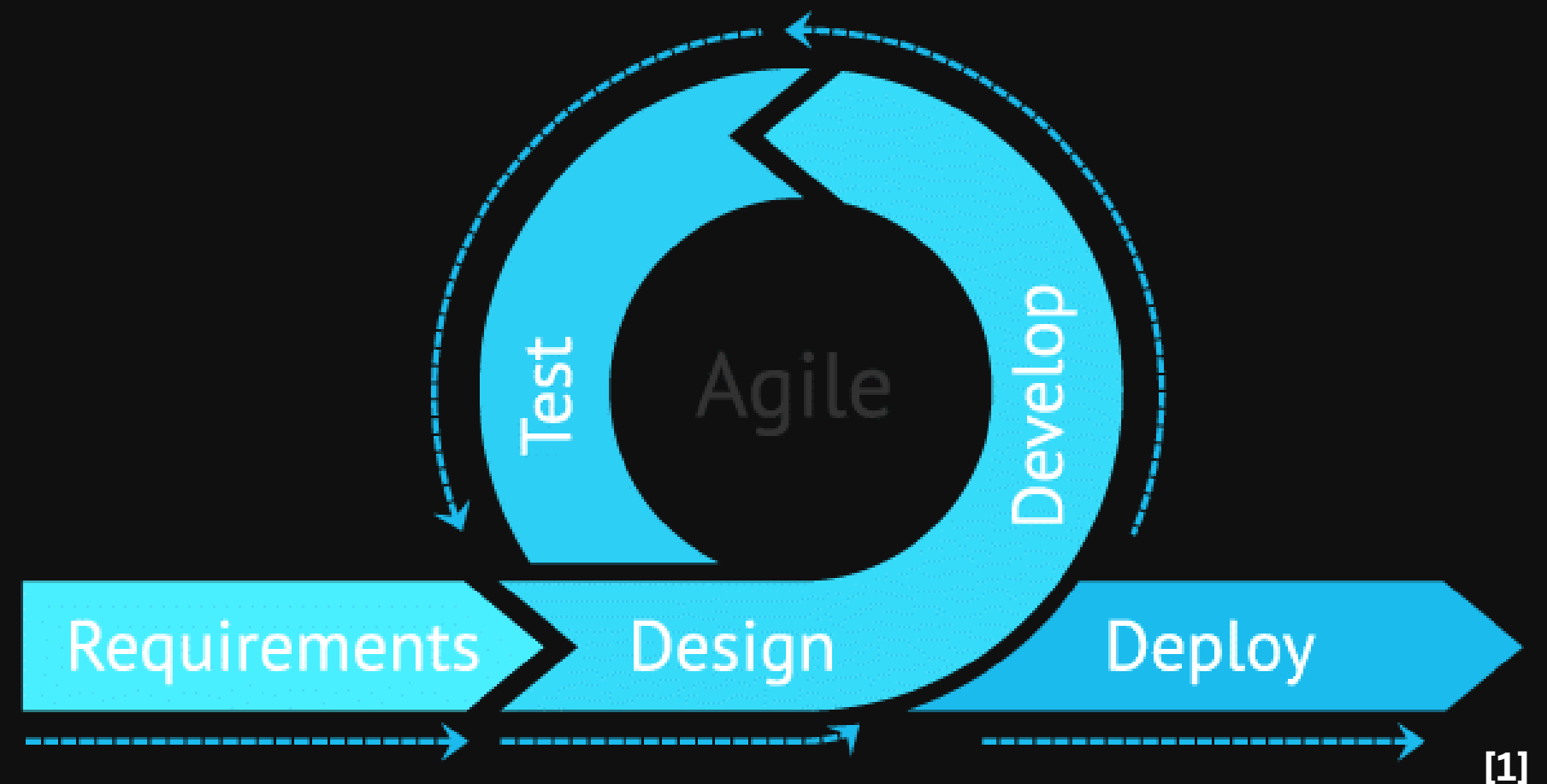
- deutsche Sprachausführung
- Modifizierbarkeit
- Wartbarkeit
- Erweiterbarkeit
- Fehlertoleranz
- Kompatibilität

QUALITÄTSSICHERUNG

Wie wurde eine hohe Qualität gewährleistet?

Linear - Agile Projektentwicklung

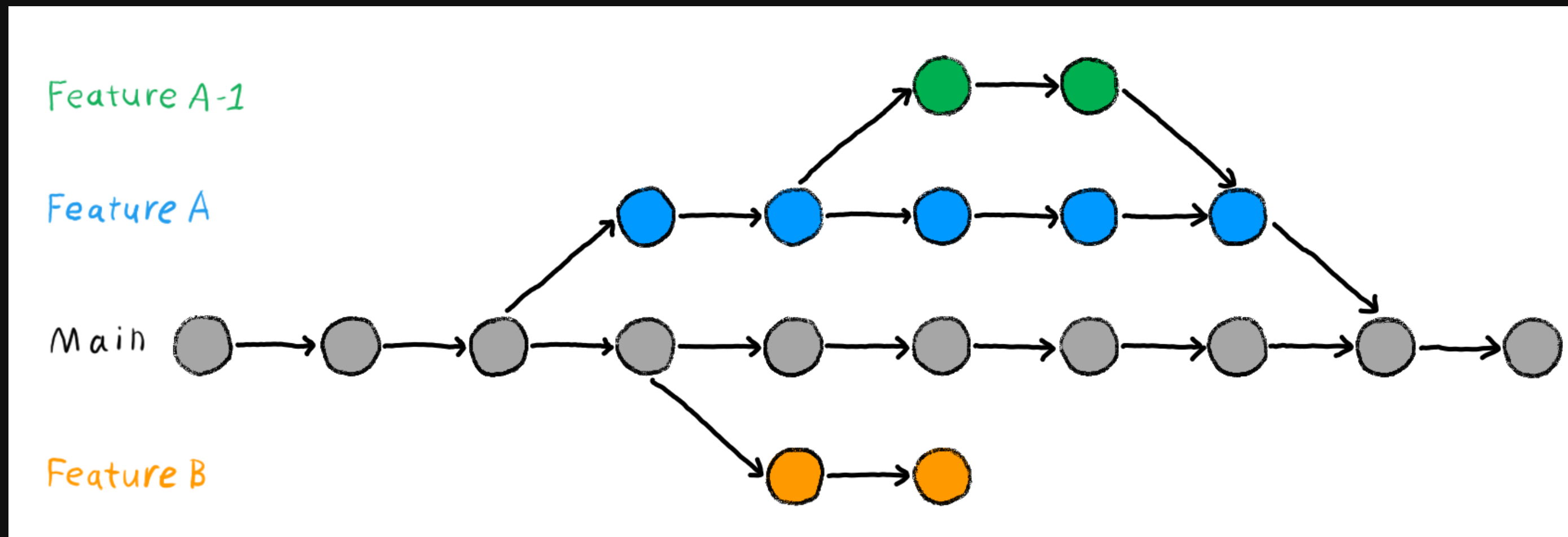
All issues	☆	Filter
🟢 In Review	▲ 2	
<input type="checkbox"/> --- SE2-89	🟢 AkzeptanzTest	› Test
--- SE2-85	🟢 Test	🔄 3/4
🟢 Done	▲ 196	
--- SE2-87	🟢 IntegrationsTests	› Test 2/2
--- SE2-86	🟢 UnitTest	› Test 6/6
--- SE2-41	🟢 presentation	🔄 1/1
--- SE2-45	🟢 terminal	› presentation 8/8
--- SE2-53	🟢 inputreader	› terminal
--- SE2-54	🟢 drawboard	› terminal



QUALITÄTSSICHERUNG

Wie wurde eine hohe Qualität gewährleistet?

Git - Versionskontrolle & Branches

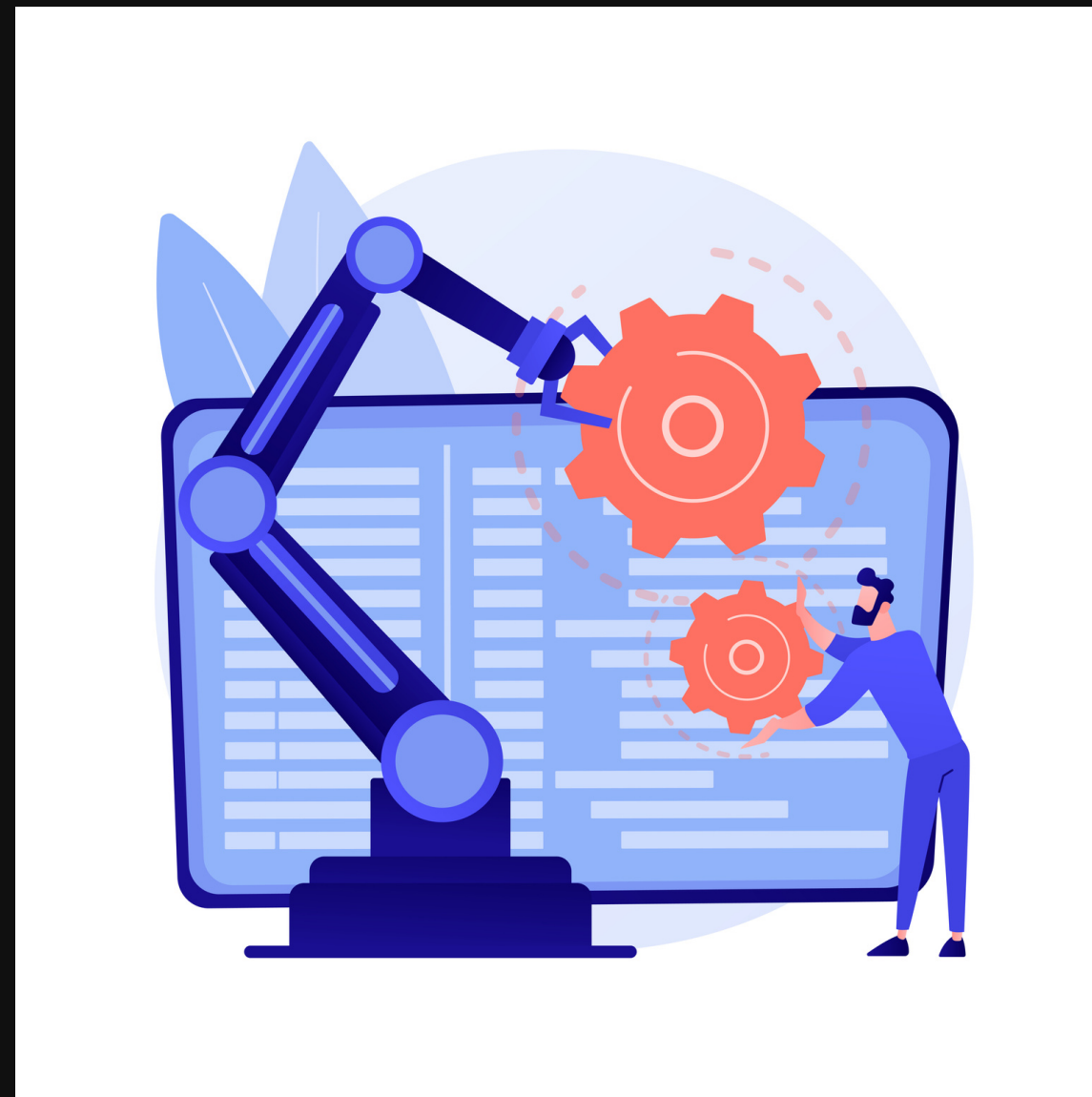


[2]

QUALITÄTSSICHERUNG

Wie wurde eine hohe Qualität gewährleistet?

Prototyping



[3]

QUALITÄTSSICHERUNG

Weitere Maßnahmen zur Qualitätssicherung

- Pair Programming
- Diagramme zur visuellen Stützung der Ideen
- PEP8 als Style Guide
- Test Driven Development

FAZIT ZUR PROJEKT ARBEIT

**VIELEN
DANK!**

Endpräsentation der theoretischen Ergebnisse

Carl Weiß, 572802
Can Wrobel, 585031
Alexander Yurovskyy, 572782
Linda Zubenin, 584625
Rico Flemming, 584322

BILDQUELLE

1. <https://the-turing-way.netlify.app/reproducible-research/vcs/vcs-git-branches.html/commons/2/2d/Mastermind.jpg>
2. <https://hive.com/blog/what-is-agile-project-management-methodology/>
3. https://www.freepik.com/free-vector/collaborative-robotics-abstract-concept-illustration_11668594.htm#page=7&query=robot%20assembly&position=0&from_view=keyword&track=ais&uuid=d75f9e53-2bd0-4bbd-a542-fb6dd1a5e377