# Dokumentation Scrum 426

Silvan, Valentin, Elias, Noah, Jann

## Perspektive 3

Welche Patterns wurden angewendet:

* Model-View-Controller: Zusammenspiel beim Login von Front.- Backend
* Dependency Injection: Injection der einzelnen Komponenten
* Singleton-Pattern: Jeweils nur eine Instanz einer Komponente

Wiederverwendbare SW-Komponenten:

* Unsere ganze Website besteht mehr oder weniger aus wiederverwendbaren Komponenten, da dies unteranderem ein Grundprinzip von Angular ist. Der Vorteil dabei ist das die einzelnen Komponenten fast einwandfrei ausgetauscht werden können. Beispiele in unserem Code sind bspw. NavBar, Footer und FruitDivider

DoD:

* Wir nutzten DoD für alle Tasks, um zu definieren, wann dieser fertig ist. Zugleich dienten sie uns auch um Zielorientiert die Tasks abzubauen.

Code Nachhaltigkeit & Clean Code:

* Code Einrückungen
* Keine zu lange Zeilen (80-120 Zeichen)
* Ebenfalls wurde auf Nachhaltigkeit geschaut, genau gesagt das maximal viel wiederverwendbar gemacht wurde

Refactoring:

* Möglichst wenig Code
* Auslagern Farben

Pairprogramming:

* In der Develop-Phase setzte das Dev-Team viel auf Pair Programming, welches sich auch stetig verbesserte. Es wurde nicht allgemein eingesetzt, sondern lediglich bei Problemen.

Prinzipien :

* Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

  Automatisch generierte BeschreibungEin sehr geeignetes Beispiel, welches man sehr gut in unserem Code sieht ist DRY( Don’t Repeat Yourself). Aufgezeigt wird es super an unseren NavBar. Damit wir nicht in jedem unseren Komponenten den gleichen Code für den NavBar tuen müssen, haben wir aus dem NavBar eine eigene Komponente gemacht.

SW-Architektur:

* Da wir eine Frontendprojekt, mit einem kleinen Backend haben ist die Architektur bereits im Projekt ersichtlich.

## Perspektive 4

VCS:

* Als Version Controll System benutzen wir das in Azure DevOps integrierte GitLab.

SCRUM-Tool:

* Nach ein paar Überlegungen, entschieden wir uns als SCRUM-Tool Azure DevOps einzusetzen. Es erschien uns als eine gute Lösung, da es viele Sachen miteinander verbindet, so haben wir Repo und Artefakten an einem Ort.

Entwicklungsprozess Dev-Team:

* Ein stabiles Team von Anfang an:
  + Unser Entwicklerteam war von Anfang an nicht schlecht. Mit Scrum als unserer Arbeitsmethode haben wir klare Strukturen und gute Prozesse aufgebaut.
* Bessere Treffen, bessere Ergebnisse:
  + Die Meetings und Rituale am Anfang waren nicht perfekt, aber wir haben hart daran gearbeitet, sie zu verbessern. Unsere "Rückblick-Meetings" sind jetzt produktiver. Wir finden klare Verbesserungen und setzen Maßnahmen um.
* Zusammen lernen mit Pair Programming:
  + Wir arbeiten jetzt öfter im Team. Das macht nicht nur Spaß, sondern hilft uns auch, voneinander zu lernen. Die Qualität unseres Codes ist dadurch besser geworden.
* Besser schätzen, besser planen:
  + Unsere Schätzungen für Aufgaben sind genauer geworden. Wir haben smarte Methoden gefunden, um zu wissen, wie viel Arbeit drinsteckt. Das hilft uns, realistischere Pläne zu machen.
* Sozialkompetenz im SCRUM-Team:
  + Teamwork ist für uns selbstverständlich. Innerhalb des SCRUM-Teams pflegen wir eine Atmosphäre der Kooperation und unterstützen uns gegenseitig. Konstruktive Kritik ist ein integraler Bestandteil unserer Kommunikation. Diese Sozialkompetenz fördert nicht nur ein positives Arbeitsumfeld, sondern auch die Effizienz und Qualität unserer Arbeit.