

## POC 101 Grundsätzliche Implementations-Prüfung der App:

### Beschreibung:

In der Anwendung soll getestet werden, ob die App mit vollständigem Umfang und ohne visuelle Darstellungs- oder Implementierungsfehler vorhanden ist.

### Dazu gehört:

- Vollständige Darstellung aller Statistik-Methoden an verschiedenen Datensätzen
- Korrekte Navigation innerhalb aller UI-Screens und UI-Elemente

### Exit-Kriterien:

- Aufgaben aller Methoden werden an verschiedenen Datensätzen korrekt dargestellt
- Aufgaben aller Methoden werden an verschiedenen Datensätzen korrekt transformiert
- Aufgaben aller Methoden werden an verschiedenen Datensätzen generiert
- Bei Nutzung der App entstehen keine Bugs
- Bei Nutzung der App entstehen keine visuellen Darstellungsfehler oder Artefakte
- Die Darstellungen und UI-Elemente sind auf jedem Screen mit verschiedenen Endgeräten und Formaten korrekt
- Zu jeweiligen Datensätzen werden die richtigen Inhalte generiert

### Fail-Kriterien:

- Es sind visuelle Artefakte oder Bugs in einem oder mehreren Screens zu sehen
- Es werden falsche, keine oder unvollständige Darstellungen präsentiert
- Es werden falsche, keine oder unvollständige Datensätze generiert
- Aufgaben präsentieren Methoden die nicht für den Datensatz geeignet sind
- Der Nutzer wird nicht zu dem richtigen UI-Screen navigiert

### Fallbacks:

1. Die Entwickler erweitern Kompetenzen mit der Entwicklungsumgebung
2. Für schwierige Teile der Anwendungslogik wird outsourcing betrieben die auf Kotlin oder visuelle Darstellung spezialisiert sind
3. Es werden Funktionen gestrichen oder neu priorisiert
4. Die Wahl der Entwicklungsumgebung wird neu evaluiert und gegebenenfalls geändert

## POC 102: Prüfung der Aufgabenlogik:

### Beschreibung:

Es soll der Umfang aller Statistik Methoden geprüft werden, in dem jede Statistik-Methode in einem simplen Kernkonzept implementiert wird, um einschätzen zu können, wie lange die vollständige Implementierung der Methode benötigt und wie komplex diese ausfällt.

### Darstellungs-Methoden:

- Verkürzte Achsendarstellung
- Partielle Achsendarstellung
- Ungleichmäßige Intervalle der Achsendarstellung
- Ungünstige Skalierung der Datensätze oder skalierte Achsen relativ zum Datensatz
- Achsen die nicht bei 0 starten
- Fehlende Beschriftung
- Stilistische Mittel (Farbwahl, Kontrast, Transparenz)
- Überlappende oder okkludierende Datensätze

#### Verzerrungs-Methoden:

- Gewollte selektion von Daten eines kompletten Datensatzes (Cherry-Picking der Inhalte)
- Auswahl günstiger Zeitfenster

#### Exit-Kriterien:

- Es ist eine Basisversion aller Statistik-Methoden implementiert
- Die vollständige Implementierung alle Methoden ist ausreichend einschätzbar
- Der Implementierungsumfang keiner Methode überschreitet ein Projektgefährdetes Niveau

#### Fail-Kriterien:

- Eine Methodik kann nicht implementiert werden
- Eine Methodik weist sich als zu komplex aus
- Eine Methodik benötigt zu viel Zeit bei der Umsetzung

#### Fallbacks:

1. Es wird anderes Personal für die Aufgabenlogik eingeteilt
2. Es wird mehr Personal/Zeit für die Aufgabenlogik eingeteilt
3. Statistik Methoden werden reduziert

#### POC 103: Prüfung der Nutzervorraussetzungen:

##### Beschreibung:

Es soll die benötigte Grundkompetenz und der benötigte Wissensstand mit dem Thema Statistik eines Nutzers geprüft werden, um Inhalte zu verstehen, einordnen zu können und den Lernerfolg zu ermöglichen. Es werden Nutzertests durchgeführt und evaluiert, die aufzeigen sollen, wann und wie stark Nutzer Voraussetzungen in dem Rahmen nicht erfüllt werden.

##### Exit-Kriterien:

- Nutzer verstehen 80% der gezeigten/dargestellten Inhalte
- Nutzer können 80% der Inhalte bei Fragen richtig einordnen
- Nutzer lösen mindestens 20% der Aufgaben im leichtesten Schwierigkeitsgrad nach Durchführung des Tutorials
- Nutzer zeigen einen Lernerfolg von mindestens 20% nach dem Lösen von 100 Aufgaben

##### Fail-Kriterien:

- Nutzer verstehen weniger als 20% der gezeigten/dargestellten Inhalte
- Nutzer können nur weniger als 20% der gezeigten Inhalte bei Fragen richtig einordnen
- Nutzer schaffen das Tutorial nicht
- Nutzer lösen nur weniger als 20% der Aufgaben im einfachsten Schwierigkeitsgrad nach Durchführung des Tutorials
- Nutzer erzielen keinen ausreichenden Lernerfolg

##### Fallback:

1. Das Tutorial wird angepasst oder erweitert
2. Die Schwierigkeit der Aufgaben wird angepasst

3. Es werden App-Systeme hinzugefügt die innerhalb der ersten Aufgaben Hilfestellung bieten
4. Zu schwere Aufgaben werden gestrichen

## POC 104: Prüfung der Darstellung und Statistik Transformation:

### Beschreibung:

Es soll überprüft werden, ob alle Statistik-Darstellungen, Transformationen und Optionen zur vollständigen Entwicklung aller Methoden vorhanden und mit den nötigen Ressourcen umsetzbar sind. Diese Prüfung kann erst nach der erfolgreichen Durchführung von POC 101 angesetzt werden, da die Anwendungslogik für alle Darstellungen und Aufgaben-Typen vorhanden sein muss. Für diese Prüfung wird eine Basis-Implementierung der Darstellung und der Transformationen entwickelt.

### Darstellungs-Methoden:

- Verkürzte Achsendarstellung
- Partielle Achsendarstellung
- Ungleichmäßige Intervalle der Achsendarstellung
- Ungünstige Skalierung der Datensätze oder skalierte Achsen relativ zum Datensatz
- Achsen die nicht bei 0 starten
- Fehlende Beschriftung
- Stilistische Mittel (Farbwahl, Kontrast, Transparenz)
- Überlappende oder okkludierende Datensätze

### Verzerrungs-Methoden:

- Gewollte Selektion von Daten eines kompletten Datensatzes (Cherry-Picking der Inhalte)
- Auswahl günstiger Zeitfenster

### Exit-Kriterien:

- Die Entwicklungsumgebung der Kotlin Statistik API weiß keine Limitierungen auf die das Projekt gefährden könnten
- Alle Aufgaben-Typen und Statistik-Methoden können korrekt dargestellt werden
- Die Darstellungs-Transformationen zwischen allen Statistik-Formaten funktionieren

### Fail-Kriterien:

- Die Entwicklungsumgebung der Kotlin Statistik API limitiert eine Implementierung/Darstellung/Transformation oder gibt nicht die nötigen Mittel alle Methoden oder Aufgaben zu entwickeln

### Fallbacks:

1. Es wird evaluiert ob die Darstellung auch intern anders als über die Kotlin Statistik-API implementiert werden kann
2. Aufgaben-Typen / Darstellungen die nicht umgesetzt werden können, werden ausgetauscht oder gestrichen
3. Es wird eine weitere Schnittstelle zur App hinzugefügt die nötige Entwicklungsoptionen liefert
4. Es wird eine bessere Schnittstelle als grundlegende API zur Darstellung von Statistik gesucht

## POC 105 Prüfung der Kernfunktionen

### Beschreibung des Vorhabens:

Es soll geprüft werden, ob sich der geplante Gesamtfunktionsumfang der App in modulare Kernfunktionen herunterbrechen lässt, die trotz begrenzter Team- und Zeitkapazität vollständig, stabil und in nutzbarer Qualität implementiert werden können.

Der Poc überprüft, ob:

- Aufgaben anzeigen, Antwortauswahl, Antwortauswertung und Navigation im Kernprinzip ohne Erweiterungen(Animationen/Transformationseffekte, Gamification) stabil laufen.
- Technische Modularität, die eine schrittweise Erweiterung ermöglicht, vorhanden ist

### EXIT-Kriterien

- Die Kernfunktionen (Aufgaben anzeigen, Antworten auswählen, Lösungsabgabe, Lösungsausgabe und Navigation) funktionieren zuverlässig
- Der geschätzte Zeitaufwand wird eingehalten(+/- 20%)
- Kernfunktionen können parallel entwickelt werden
- Features sind sauber voneinander getrennt

### FAIL-Kriterien

- Deaktivierte Features führen zu App-Crashes oder inkonsistenter UI
- Die Kernfunktionskette funktioniert nicht
- Features sind nicht sauber voneinander getrennt
- Ein Großteil der Kernfunktionen benötigt deutlich mehr Zeit als geplant

### Fallbacks:

1. Funktionsumfang reduzieren
2. Aufgaben-Generierung vereinfachen

## POC 106 Gamification

Es soll geprüft werden, ob die App genügend Spielspaß und Erfolgserlebnisse liefert, zur Gewährleistung der Nutzermotivation.

### Gamification Methoden:

- Experience System
- XP Rewards (Titel, Customization)
- Achievements
- Daily Challenges/Streaks
- User Statistik
- Schwierigkeitsgrade

### EXIT-Kriterien:

- Mindestens 4/6 der Gamification Methoden wurden implementiert
- Nutzer zeigen wiederholte Sitzungen
- Nutzer bewerten die App als spaßig
- Nutzer berichten von positiven Erfolgserlebnissen

### FAIL-Kriterien:

- Ungenügender Einsatz von Gamification methoden (<4/6)
- Nutzer brechen Aufgaben ab oder kehren nicht wieder

- Belohnungssystem wird als langweilig oder unverständlich erlebt

Fallbacks:

1. Anpassung der Belohnungsfrequenz
2. leichter erreichbare Badges statt Achievements
3. Erweiterung mit weiteren Gamification Methoden
4. Anpassung der Schwierigkeitsgrade

## PoC 107 Accessibility

Beschreibung des Vorhabens:

Es soll geprüft werden, ob das User Interface für verschiedene Altersgruppen verständlich, leicht, zugänglich und intuitiv bedienbar ist.

Dazu gehört:

- Schriftgrößen, Lesbarkeit, Farbkontraste
- intuitive Navigation
- Bedienbarkeit
- Tutorial

EXIT-Kriterien::

- >80% der Testnutzer können die Grundfunktionen ohne Erklärung bedienen
- Navigation wird als klar und eindeutig bewertet
- Tutorial wird als klar und verständlich aufgenommen
- Nutzer bewerten die UI als ansprechend und übersichtlich

FAIL-Kriterien:

- <80% der Testnutzer können die Grundfunktionen ohne Erklärung bedienen
- Mehrfaches Auftreten von Bedienproblemen
- UI wird als überladen oder zu komplex wahrgenommen
- Tutorial erzielt keinen Nutzen

Fallbacks:

1: weitere ausführlichere Tutorials hinzufügen

2: Simplifizierung der UI

## PoC 108 Prüfung der Datenqualität

Beschreibung des Vorhabens:

Es soll geprüft werden, ob genügend hochwertige Datensätze/historische Bezugsinhalte bereitgestellt werden.

getestet wird:

- Varianz der Datenquellen
- Passende Beispiele im Bezug zur jeweiligen Aufgabe
- Realismus der Beispiele

EXIT-Kriterien

- Nach Lösen einer Aufgabe können über den "Where was this used" Button passende und qualitativ hochwertige Beispiele angezeigt werden.
- Zu jeder Manipulationsmethode gibt es mindestens 3 Beispiele
- Die Beispiele stammen ausschließlich aus seriösen Quellen
- Die Beispiele haben eine gesellschaftliche Relevanz

## FAIL-Kriterien

- “Where was this used button” liefert unpassende Beispiele
- Es wurden nicht ausreichend Beispiele aus seriösen Quellen gefunden
- Die Beispiele sind irrelevant/uninteressant

## Fallbacks:

1. Anstatt Realbeispielen werden fiktive Beispiele erstellt
2. Anstatt dem “Where was these Used” Button wird eine Auflistung mit Beispielen zu allen Manipulationsmethoden bereitgestellt
3. Beispiele werden über eine externe Schnittstelle bereitgestellt